Performance analysis of randomised search heuristics operating with a fixed budget

Jansen, Thomas; Zarges, Christine

Published in:
Theoretical Computer Science

DOI:
10.1016/j.tcs.2013.06.007

Publication date:
2014

Citation for published version (APA):
Jansen, T., & Zarges, C. (2014). Performance analysis of randomised search heuristics operating with a fixed budget. *Theoretical Computer Science*, *545*, 39-58. https://doi.org/10.1016/j.tcs.2013.06.007

Performance Analysis of Randomised Search Heuristics Operating With a Fixed Budget☆

Thomas Jansen[a], Christine Zarges[b,1]

[a]*Department of Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, United Kingdom*
[b]*School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom*

Abstract

When for a difficult real-world optimisation problem no good problem-specific algorithm is available often randomised search heuristics are used. They are hoped to deliver good solutions in acceptable time. The theoretical analysis usually concentrates on the average time needed to find an optimal or approximately optimal solution. This matches neither the application in practice nor the empirical analysis since usually optimal solutions are not known and even if found cannot be recognised. More often the algorithms are stopped after some time. This motivates a theoretical analysis to concentrate on the quality of the best solution obtained after a pre-specified number of function evaluations called budget. Using this perspective two simple randomised search heuristics, random local search and the (1+1) evolutionary algorithm, are analysed on some well-known example problems. Upper and lower bounds on the expected quality of a solution for a fixed budget of function evaluations are proven. The analysis shows novel and challenging problems in the study of randomised search heuristics. It demonstrates the potential of this shift in perspective from expected run time to expected solution quality.

Keywords: runtime analysis, fixed budget computation, random local

☆Parts of the results appeared in the Proceedings of 14th Annual Conference on Genetic and Evolutionary Computation (GECCO 2012) [1].

Email addresses: `t.jansen@aber.ac.uk` (Thomas Jansen), `C.Zarges@cs.bham.ac.uk` (Christine Zarges)

[1]This author was partly supported by a postdoctoral fellowship from the German Academic Exchange Service (DAAD) while visiting the University of Warwick in Coventry, UK.

search, (1+1) EA, OneMax, LeadingOnes, Jump, Ridge

1. Introduction

Randomised search heuristics are a large class of heuristic search algorithms that comprises evolutionary algorithms, particle swarm optimisers, ant colony optimisation, artificial immune systems, local search, simulated annealing, and many others. In the theoretical analysis of randomised search heuristics it is most common to study the expected optimisation time, i. e., the average number of steps needed and sufficient to locate a global optimum [2]. Sometimes this is changed to study the expected approximation time, i. e., the average number of steps needed and sufficient to locate a solution that approximates the quality of a global optimum with a prespecified approximation ratio (see e. g. [3]). It turns out that in this case the type of results that is obtained does not change fundamentally. Even more importantly, the very same methods developed to analyse the expected optimisation time are applicable and useful when studying the expected approximation time. To simplify the discussion we refer to optimisation in the following. However, all remarks apply equally to the case of approximation.

Both approaches share a common perspective that is in some sense odd and not really fitting since randomised search heuristics are incomplete optimisers and the user never really knows when and if a global optimum (or a solution of a certain approximation quality) has been found. Thus, from an application point of view the interesting question is not how long one has to wait on average until an optimal solution is found. It is much more common that a solution needs to be found and the available time for the search is limited.

The same is true when the performance of randomised search heuristics is studied empirically. Unless known benchmark or example problems are used the value of an optimal solution is not known and therefore other performance measures need to be studied. One tool to do this are so called best-so-far curves [4]. They display the function value of the best search point that has been found in the first t steps of the algorithm (plotted as a function of t). Since an optimal value is not available to decide when a run should be terminated often algorithms are terminated after a fixed number of steps and their results are evaluated.

This corresponds to a situation where we have a fixed budget of steps that we may spend and we ask ourselves how good a solution we can expect to

find with this budget. In particular, practitioners would like to know about the impact a change in the budget can be expected to have. 'If I double my budget how will this affect the quality of the solution?' Note that this question is similar but not identical to the question how best to spend a fixed budget, in one long or multiple short runs [5].

It makes a difference if one considers small or large budgets. If the budget is large in comparison to the expected time needed to locate an optimum it makes more sense to ask how close to 1 is the probability to have found an optimum already. If the budget is small in comparison to that time it makes more sense to investigate the expected quality of the solution. For the case of a large budget results on the expected optimisation time deliver some information. Even simple application of Markov's inequality yields some bounds. The case of a smaller budget presents us with new challenging analytical problems and it is this case of small computational budgets that we consider here.

We remark that the expected optimisation time is sometimes but not always a good indicator for the time needed and sufficient to find an optimal solution. The function

$$f(x) = \begin{cases} n - .5 & \text{if } x = 0^n \\ \sum_{i=1}^{n} x[i] & \text{otherwise} \end{cases}$$

is an example where this is the case. The unique global optimum of the function is the all ones bit string with function value n. For almost all other bit strings the function value also equals the number of 1-bits. Thus, it is easy to follow a path of increasing numbers of 1-bits leading to the global optimum quickly. The only exception is the all zeros bit string. It has function value $n - .5$, the second best function value. If this bit string is found it may be difficult to get away from it. For simple search algorithms like the (1+1) EA this does indeed lead to the result that the expected optimisation time is very large (in fact, it is $\Omega((n/2)^n)$ for the (1+1) EA) while in practice the global optimum is almost always found very efficiently (in time $O(n \log n)$ for the same algorithm). However, for the example functions considered here such pathological cases are avoided and the expected optimisation time is an adequate measure for the time that is typically needed to find a global optimum.

We concentrate on the more interesting case of smaller budgets and restrict our analysis to two very simple randomised search heuristics. It is sub-

ject of future research to extend the analysis to more complete (and therefore challenging) randomised search heuristics. We consider random local search (RLS) where in each step exactly one bit is chosen uniformly at random and is flipped. In addition, we consider the (1+1) EA where in each step each bit is flipped independently with probability $1/n$. We will see that the analysis for the (1+1) EA is already much more difficult.

In the next section we give a formal description of our model, the considered algorithms and introduce the basic example functions. Moreover, we motivate the choices we make, explain the perspective and outline challenges for current and future research. Section 3 is concerned with the analysis for random local search. The simple structure of this search heuristic facilitates analysis. We present results in the form of upper and lower bounds on the expected function value after a fixed budget of function evaluations for five different example problems. All example problems are well-known and have been studied before, they are not introduced to facilitate our analysis. We accompany our bounds by the results of experiments to give an impression of the quality of the obtained bounds. In Section 4 we consider the (1+1) EA that uses mutations instead of neighbourhood search. We show that this seemingly small change complicates things considerably. Finally, Section 5 summarises and points out open questions and topics for future research.

2. Models and Notation

The framework we introduce is an alternative to the usual way of analysing the performance of randomised search heuristics. We deliberately design the framework similar to the usual perspective in order to allow for comparisons and transfer of results as well as analytical methods.

In our framework as well as in the usual model the most crucial notion is that of time. Usually, in computer science time denotes the number of computation steps in a model of computation. However, almost always the analysis of randomised search heuristics is simplified by not actually counting the number of computation steps but instead counting the number of times the objective function is evaluated. Randomised search heuristics are often algorithmically simple and evaluating the objective function can be the by far most costly operation. In such cases this simplification is justified and useful. It holds often (but not always [6]) and we adopt this decision here, too. Thus, a fixed budget b means that a randomised search heuristic may

make in total b function evaluations to find a point in the search space that is as good as possible.

Arbitrary randomised search heuristics can be analysed using this perspective of fixed budget searches. We consider two very simple examples that have both been studied in great depth, namely random local search and the (1+1) EA. In particular, comparisons between the two have been made [7]. We define both algorithms formally as Algorithm 1 and Algorithm 2, respectively. In both algorithms we leave the initialisation open. We consider two different variants: initialising uniformly at random and starting in a fixed starting point.

Algorithm 1. *Random Local Search (RLS)*

1. $t := 0$. *Select* $x_t \in \{0,1\}^n$. *Evaluate* $f(x_t)$.
2. *While* $t + 1 < b$ *do*
3. $\quad t := t + 1$. $y := x_{t-1}$.
4. $\quad$ *Select* $i \in \{1, 2, \ldots, n\}$ *uniformly at random. Flip* i*-th bit in* y.
5. $\quad$ *Evaluate* $f(y)$.
6. $\quad$ *If* $f(y) \geq f(x_{t-1})$ *then* $x_t := y$ *else* $x_t := x_{t-1}$.

Algorithm 2. *(1+1) Evolutionary Algorithm ((1+1) EA)*

1. $t := 0$. *Select* $x_t \in \{0,1\}^n$. *Evaluate* $f(x_t)$.
2. *While* $t + 1 < b$ *do*
3. $\quad t := t + 1$. $y := x_{t-1}$.
4. $\quad$ *For each* $i \in \{1, 2, \ldots, n\}$*: With probability* $1/n$ *flip* i*-th bit in* y.
5. $\quad$ *Evaluate* $f(y)$.
6. $\quad$ *If* $f(y) \geq f(x_{t-1})$ *then* $x_t := y$ *else* $x_t := x_{t-1}$.

Both algorithms use a 'population' of search points of size only 1 and are elitist: $f(x_t)$ is a non-decreasing function of t. Therefore it is obvious what the expected function value after b function evaluations is. When analysing an (evolutionary) algorithm with a larger population it will make most sense to analyse the expected function value after a generation is completed. When interested in best-so-far curves one would always consider the maximal function value encountered so far even if, for non-elitist algorithms, the corresponding search point is no longer part of the population.

When analysing new algorithms, introducing new analytical techniques or discussing new aspects it is customary in the theory of randomised search heuristics to start with particularly simple example functions. Two of the

best known of these simple example functions are OM (often called OneMax) and LO (often called LeadingOnes) [6, 8, 9, 10]. Their simple structure facilitates analysis and exemplifies important points in a paradigmatic way. For these reasons we also consider these functions and in addition some others, namely J_k (usually called Jump_k[9]), R (often called Ridge [11]), and P (short for Prefix). We give a precise formal definition for all functions here.

Definition 3. *Let $n \in \mathbb{N}$ and $k \in \{1, 2, \ldots, n\}$. The following five functions are all defined so that they map from $\{0,1\}^n$ to $\mathbb{N}_0$.*

- $\mathrm{OM}(x) = \sum_{i=1}^{n} x[i]$
- $\mathrm{LO}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x[j]$
- $\mathrm{P}(x) = n \cdot \mathrm{LO}(x) - \mathrm{OM}(x)$
- $\mathrm{R}(x) = \begin{cases} n + i & \textit{if } x = 1^i 0^{n-i} \\ n - \mathrm{OM}(x) & \textit{otherwise} \end{cases}$
- $\mathrm{J}_k(x) = \begin{cases} n - \mathrm{OM}(x) & \textit{if } n - k < \mathrm{OM}(x) < n \\ k + \mathrm{OM}(x) & \textit{otherwise} \end{cases}$

All functions have the all ones bit string as their unique global optimum. The function OM yields as function value the number of 1-bits, the function LO the number of consecutive 1-bits counting from left to right. The function P is similar to LO but 'insists' on bit strings of the form $1^i 0^{n-i}$. Additional spare 1-bits in the suffix of the function are punished by reducing the function value by 1 for each such 1-bit. The function R is very similar but even stricter. Any bit string that is not of the form $1^i 0^{n-i}$ has a function value that is at most n and is decreased by 1 for each 1-bit. The function J_k, finally, is similar to OM but contains a gap between the all ones bit string (the global optimum) and all bit strings with $n - k$ 1-bits, making these local optima.

In the following section we analyse the performance of RLS on these functions for a fixed budget b that is smaller than the expected optimisation time, i. e., $b < \mathrm{E}\left(T_{RLS,f}\right)$ where $\mathrm{E}\left(T_{RLS,f}\right)$ denotes the expected optimisation time of random local search on the objective function f. We prove results for

$\mathrm{E}(f(x_b))$ using the notation from Algorithm 1. In Section 4 we go beyond RLS and consider the (1+1) EA.

The model of fixed budget computation aims at delivering a theory that is useful for practitioners. Thus, we must be able to answer questions about the effects of having budget $2b$ instead of b. This implies that an asymptotic analysis like $\mathrm{E}(f(x_b)) = \Theta(s(n))$ is not sufficiently precise. We point out that obtaining such precise results is also a current trend in the analysis of expected optimisation times [12, 13].

3. Fixed Budget Results for Random Local Search

We consider the expected performance of random local search on the five different example functions in an order that facilitates the analysis. It turns out that the analysis for OM is particularly simple. This is not surprising since the process is identical to the well known coupon collector scenario (see, e. g., [14]). Note, however, that there the expected number of coupons that needs to be bought to have a complete collection is analysed. This corresponds to the expected optimisation time and is, of course, very different from our perspective here.

3.1. Random Local Search on OM

We begin with RLS on OM. We start with the special case of deterministic initialisation in the all zeros bit string 0^n. This allows to get a clearer picture of the performance of the algorithm that is not obscured by the random effects of initialisation.

Theorem 4. *With $x_0 = 0^n$, for all budgets $b \in \mathbb{N}$*

$$E(\mathrm{OM}(x_b)) = n \cdot \left(1 - \left(1 - \frac{1}{n}\right)^b\right)$$

holds for RLS on OM.

Proof. We observe that all bits $x[1], x[2], \ldots, x[n]$ are initially set to 0 and have the value 1 if and only if there is a point of time when this specific bit is flipped. Let random variables $X_{i,t} \in \{0, 1\}$ obtain the value 1 if and only if $x_t[i] = 1$. Using this definition $\mathrm{OM}(x_b) = \sum_{i=1}^{n} X_{i,b}$ holds. We conclude that

we have

$$\mathrm{E}\left(\mathrm{OM}(x_b)\right) = \mathrm{E}\left(\sum_{i=1}^{n} X_{i,b}\right) = \sum_{i=1}^{n} \mathrm{E}\left(X_{i,b}\right)$$

by linearity of expectation and $\mathrm{E}\left(\mathrm{OM}(x_b)\right) = n \cdot \mathrm{E}\left(X_{1,b}\right)$ by symmetry. Since $X_{1,b}$ is an indicator variable $\mathrm{E}\left(X_{1,b}\right) = \mathrm{Prob}\left(X_{1,b} = 1\right)$ holds. It is easy to see that we have $X_{1,b} = 0$ if and only the first bit was never flipped in all b steps. Thus, $\mathrm{E}\left(X_{1,b}\right) = 1 - \mathrm{Prob}\left(X_{1,b} = 0\right) = 1 - (1 - 1/n)^b$ holds and we obtain the claimed result. □

For random initialisation things are only slightly more involved. It is known that one expects to have half of the bits set to 1, initially. For the remaining bits we have the same process as for initialisation 0^n. This leads to $\mathrm{E}\left(\mathrm{OM}(x_b)\right) = (n/2) + (n/2) \cdot \left(1 - (1 - 1/n)^b\right)$ and it is not hard to prove this to be correct rigorously.

Theorem 5. *With $x_0 \in \{0,1\}^n$ uniformly at random, for all budgets $b \in \mathbb{N}$*

$$E\left(\mathrm{OM}(x_b)\right) = \frac{n}{2} + \frac{n}{2} \cdot \left(1 - \left(1 - \frac{1}{n}\right)^b\right)$$

holds for RLS on OM.

Proof. The main tool for the proof is the law of total probability. This yields

$$\mathrm{E}\left(\mathrm{OM}(x_b)\right) = \sum_{i=0}^{n} \mathrm{Prob}\left(\mathrm{OM}(x_0) = i\right) \cdot \mathrm{E}\left(\mathrm{OM}(x_b) \mid \mathrm{OM}(x_0) = i\right).$$

Given that initially we have i 1-bits the expected function value after b function evaluations equals $i + (n - i)\left(1 - (1 - 1/n)^b\right)$ so that

$$\mathrm{E}\left(\mathrm{OM}(x_b)\right) = \sum_{i=0}^{n} \mathrm{Prob}\left(\mathrm{OM}(x_0) = i\right) \cdot \left(i + (n - i)\left(1 - \left(1 - \frac{1}{n}\right)^b\right)\right)$$

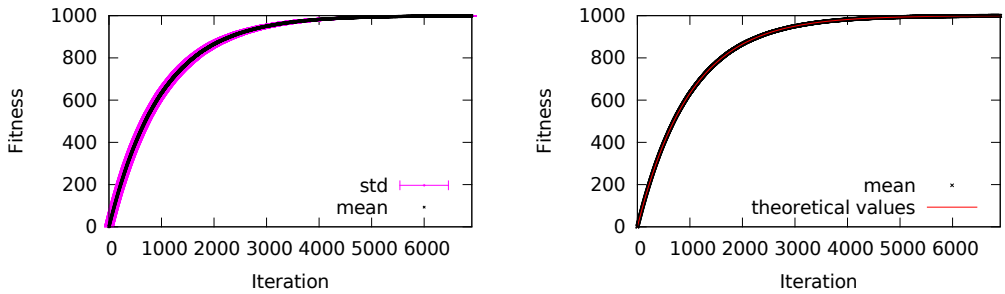

Figure 1: Empirical results for OM, $n = 1000$, and deterministic initialisation in 0^{1000}: average values together with their standard deviations (left) and with the theoretical bounds (right).

follows. Plugging in $\text{Prob}\,(\text{OM}(x_0) = i) = \binom{n}{i} \cdot 2^{-n}$ we obtain

$$\begin{aligned}
\text{E}\,(\text{OM}(x_b)) &= \sum_{i=0}^{n} \binom{n}{i} 2^{-n} \left(i + (n-i) \cdot \left(1 - \left(1 - \frac{1}{n} \right)^b \right) \right) \\
&= \left(\sum_{i=0}^{n} i \binom{n}{i} 2^{-n} \right) + \sum_{i=0}^{n} (n-i) \binom{n}{i} 2^{-n} \cdot \left(1 - \left(1 - \frac{1}{n} \right)^b \right) \\
&= \frac{n}{2} + \frac{n}{2} \cdot \left(1 - \left(1 - \frac{1}{n} \right)^b \right).
\end{aligned}$$

□

Since we have an exact result here there seems to be little value in presenting empirical results. We still do this and by doing so get an impression on the influence of random fluctuations when averaging over a finite number of runs. Here and in the following we perform experiments for $n = 1000$ (a rather arbitrary choice for n that is neither extremely large nor particularly small) and present results from 100 runs. We display the results in two diagrams, one that contains only the empirical results displaying the average values together with their standard deviations. The second diagram contains only the average values and for the purpose of comparison the theoretical bounds. The results for deterministic initialisation in 0^n can be found in Figure 1; the results for random initialisation are depicted in Figure 2.

We see in both figures in the left image that the standard deviation is really small. Therefore, it is acceptable to do the comparison of the theoretical

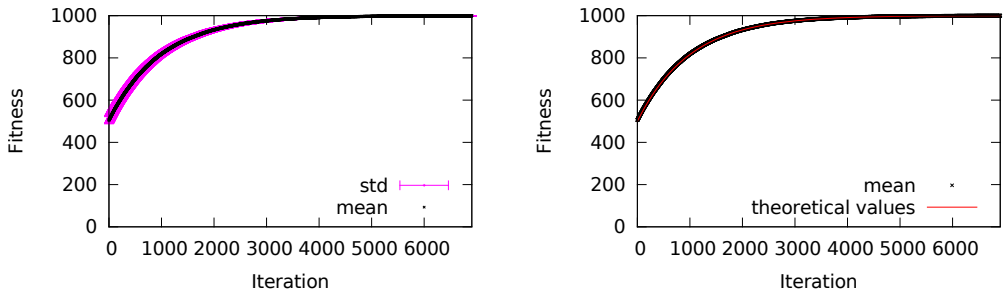

Figure 2: Empirical results for OM, $n = 1000$, and random initialisation: average values together with their standard deviations (left) and with the theoretical bounds (right).

result with the average, only. In both figures the empirical data matches the exact curve of the expected result perfectly.

3.2. Random Local Search on J_k

Since J_k is similar to OM we consider J_k next. We restrict our interest to the case of initialisation in the all zeros bit string 0^n, here. This avoids complications for very large values of k (like $k \geq n/2$) where the initial bit string may be in the region of the search space where the function value decreases with an increasing number of 1-bits. When initialising in the all zeros bit string we have $\mathrm{OM}(x_t) \leq n-k$ for the RLS on J_k at all times since local search cannot get beyond the local optima of J_k. For all local optima the function value equals $k + (n-k) = n$ and we have

$$\begin{aligned}
&\mathrm{E}\left(\mathrm{J}_k(x_b)\right) \\
&= \mathrm{E}\left(\mathrm{J}_k(x_b) \mid \mathrm{OM}(x_b) = n-k\right) \cdot \mathrm{Prob}\left(\mathrm{OM}(x_b) = n-k\right) \\
&\quad + \mathrm{E}\left(\mathrm{J}_k(x_b) \mid \mathrm{OM}(x_b) < n-k\right) \cdot \mathrm{Prob}\left(\mathrm{OM}(x_b) < n-k\right) \\
&= n \cdot \mathrm{Prob}\left(\mathrm{OM}(x_b) = n-k\right) \\
&\quad + \big(k + \mathrm{E}\left(\mathrm{OM}(x_b) \mid \mathrm{OM}(x_b) < n-k\right)\big) \cdot \mathrm{Prob}\left(\mathrm{OM}(x_b) < n-k\right)
\end{aligned}$$

since $\mathrm{J}_k(x) = k + \mathrm{OM}(x)$ holds for the bit strings RLS can encounter. Before we tackle the problem of exact bounds we present a very simple upper and a very simple lower bound.

Theorem 6. *With $x_0 = 0^n$,*

$$k + (n-k) \cdot \left(1 - \left(1 - \frac{1}{n}\right)^b\right) \leq E(\mathrm{J}_k(x_b))$$
$$\leq \min\left\{n, k + n \cdot \left(1 - \left(1 - \frac{1}{n}\right)^b\right)\right\}$$

holds for RLS on J_k.

Proof. The upper bound $\min\left\{n, k + n \cdot \left(1 - (1-1/n)^b\right)\right\}$ combines two simple observations. On the one hand, local search started in 0^n cannot get to any search point beyond the local optima that all have function value n. On the other hand, the performance on J_k is not better than the performance on OM. The addition of k takes care of the difference in function value k between J_k and OM for all bit strings RLS may encounter.

For the lower bound we again use the performance of RLS on OM for comparison. Different from the situation there the number of 1-bits cannot increase beyond $n-k$. We model this by simply fixing k bits, say the first k bits, and ignore any increase of the number of 1-bits there. This leads to $(n-k) \cdot \left(1 - (1-1/n)^b\right)$ as bound and by adding the difference in function value k as before we obtain the lower bound. □

Since $\left(1 - (1-1/n)^b\right)$ is a probability we have $\left(1 - (1-1/n)^b\right) \leq 1$ and see that the difference between the upper and lower bound is bounded above by k. The function J_k is usually considered for small, constant values of k since typical evolutionary algorithms have optimisation time $\Omega\left(n^k\right)$ without crossover and $\Omega\left(2^{2k}\right)$ with uniform crossover on J_k [15]. Therefore the bounds from Theorem 6 are actually already quite good.

We demonstrate this empirically and numerically in the diagrams below where we compare the two bounds for $n = 1000$ and $k = 5$ with the results of experiments and the actual true values for $\mathrm{E}\left(\mathrm{J}_k(x_b)\right)$ that we obtain numerically using the following ansatz.

We have $\mathrm{E}\left(\mathrm{J}_k(x_b)\right) = \sum_{i=k}^{n} i \cdot \mathrm{Prob}\left(\mathrm{J}_k(x_b) = i\right)$ since we initialise in the all zeros bit string (with $\mathrm{J}_k(0^n) = k$) and local search can at best reach a local optimum with $n-k$ one bits and function value n.

Due to our initialisation we have $\text{Prob}\,(\mathrm{J}_k(x_0) = k) = 1$ and consequently $\text{Prob}\,(\mathrm{J}_k(x_0) = i) = 0$ for all $i > k$. In the first step RLS reaches some bit string with precisely one 1-bit so that $\text{Prob}\,(\mathrm{J}_k(x_1) = k+1) = 1$ and $\text{Prob}\,(\mathrm{J}_k(x_1) = i) = 0$ for all $i \neq k+1$ follows. For all $t > 1$ and all $i < n$ we have

$$\text{Prob}\,(\mathrm{J}_k(x_t) = i) = \text{Prob}\,(\mathrm{J}_k(x_{t-1}) = i-1) \cdot \frac{n-(i-1-k)}{n} + \text{Prob}\,(\mathrm{J}_k(x_{t-1}) = i) \cdot \frac{i-k}{n}$$

since $\mathrm{J}_k(x_t) = i$ can only hold if one of the following two was the case at time step $t-1$. Either we have $\mathrm{J}_k(x_{t-1}) = i-1$ and one of $n-(i-1-k)$ 0-bits flipped (so that the number of 1-bits in x_t is increased by 1 in comparison to x_{t-1}), or we have $\mathrm{J}_k(x_{t-1}) = i$ and one of the $i-k$ 1-bits is flipped (so that $x_t = x_{t-1}$ holds after selection (line 6 in Algorithm 1)). Finally, for all $t > 1$ and $i = n$ we have

$$\text{Prob}\,(\mathrm{J}_k(x_t) = n) = \text{Prob}\,(\mathrm{J}_k(x_{t-1}) = n-1) \cdot \frac{k+1}{n} + \text{Prob}\,(\mathrm{J}_k(x_{t-1}) = n)$$

since the situation for $\mathrm{J}_k(x_t) = n$ is very similar to the case $i < n$. The only difference is that once we have $\mathrm{J}_k(x_{t-1}) = n$ this is guaranteed never to change again.

This recursive system of equations can easily be solved numerically for any value of n, k, and b (in time $O(nb)$ and space $O(n)$). The results for $n = 1000$ and $k = 5$ exemplify that the upper bound is close to the truth while $\mathrm{J}_5(x_t)$ is still clearly smaller than $n-5$ and again once it is very close to $n-5$ (Figure 3). The detailed view in Figure 4 exhibits that only for a short phase in between the upper bound deviates from the actual values. Note that the minor overshooting of the empirical results in Figure 4 is very limited and appears exaggerated due to the different scaling in this figure.

3.3. Random Local Search on P

We continue our analysis with RLS on the LO-like function P. This prepares us for the analysis on LO. Again we consider initialisation in the all zeros bit string 0^n.

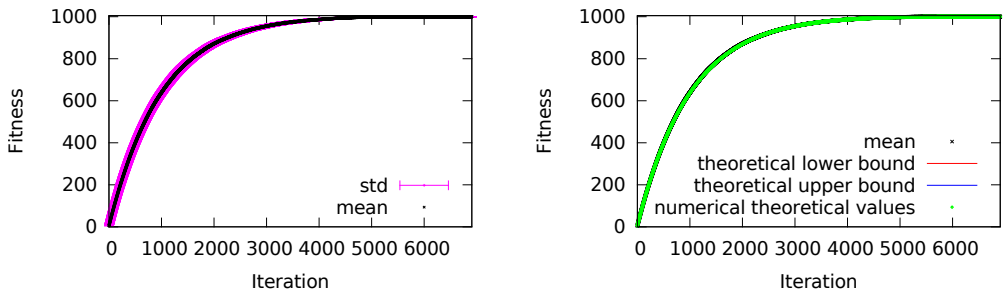

Figure 3: Empirical results for J_5, $n = 1000$, and deterministic initialisation in 0^{1000}: average values together with their standard deviations (left) and with the theoretical bounds (right).

Theorem 7. *Let the budget $b \leq (1-\varepsilon)n^2$ for some positive constant $\varepsilon < 1$. With $x_0 = 0^n$,*

$$b\left(1-\frac{1}{n}\right) - e^{-\Omega(n)} \leq E\left(\mathrm{P}(x_b)\right) \leq b\left(1-\frac{1}{n}\right)$$

holds for RLS on P.

Proof. Since we start with $x_0 = 0^n$ and RLS is an elitist algorithm we have that all x_t have the form $1^i 0^{n-i}$ for different values of $i \in \{0, 1, \ldots, n\}$. This implies that we have a bijection between P and the number of leading 1-bits. We consider the number of leading 1-bits for notational simplicity and investigate $\mathrm{E}\left(\mathrm{LO}(x_b) \mid x_0 = 0^n\right)$. Note that this is quite different from analysing RLS on LO (something we do in Section 3.4). We call a step a *success* if the number of leading 1-bits increases. In step t we start with x_{t-1} and generate x_t. We have $\mathrm{Prob}\left(\text{success in step } t+1 \mid \mathrm{LO}(x_t) < n\right) = 1/n$ and $\mathrm{Prob}\left(\text{success in step } t+1 \mid \mathrm{LO}(x_t) = n\right) = 0$.

For each $t \in \mathbb{N}$ we have that $\mathrm{E}\left(\mathrm{LO}(x_t) \mid x_0 = 0^n\right)$ equals the number of successes in the first t steps. Let S_t be a random variable with

$$S_t = \begin{cases} 1 & \text{if } \mathrm{LO}(x_t) > \mathrm{LO}(x_{t-1}), \\ 0 & \text{otherwise.} \end{cases}$$

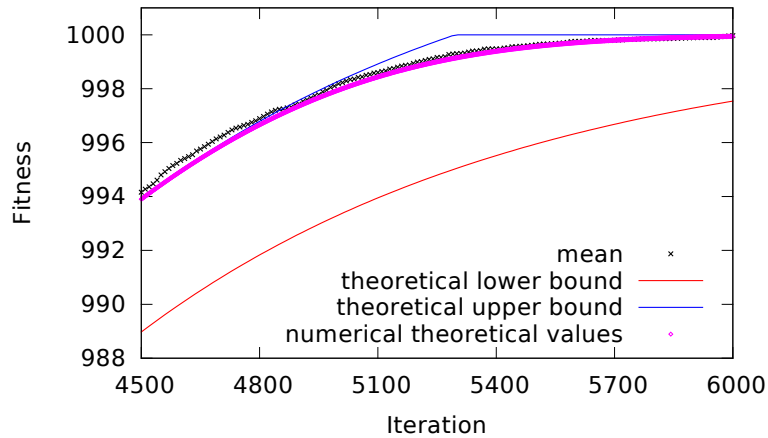

Figure 4: Empirical results for J_5, $n = 1000$, and deterministic initialisation in 0^{1000}: average values together with the theoretical bounds.

This implies

$$\begin{aligned}
&\mathrm{E}\left(\mathrm{LO}(x_b) \mid x_0 = 0^n\right) \\
&= \sum_{t=0}^{b-1} \mathrm{Prob}\left(S_{t+1} = 1\right) \\
&= \sum_{t=0}^{b-1} \mathrm{Prob}\left(S_{t+1} = 1 \mid \mathrm{LO}(x_t) < n\right) \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \\
&\quad + \mathrm{Prob}\left(S_{t+1} = 1 \mid \mathrm{LO}(x_t) = n\right) \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) = n\right) \\
&= \frac{\sum_{t=0}^{b-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right)}{n}.
\end{aligned}$$

With $\mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \leq 1$ we have $\mathrm{E}\left(\mathrm{LO}(x_b) \mid x_0 = 0^n\right) \leq \sum_{t=0}^{b-1} 1/n = b/n$ as an immediate consequence.

We consider a sequence of b completely independent random variables $S_1^*, S_2^*, \ldots, S_b^* \in \{0, 1\}$ with $\mathrm{Prob}\left(S_i^* = 1\right) = 1/n$ for all $i \in \{1, 2, \ldots, b\}$. For a given t we consider the first t of these random variables, $S_1^*, S_2^*, \ldots, S_t^*$. The probability to have $S_i^* = 1$ less than n times in this sequence equals

$\text{Prob}\,(\text{LO}(x_t) < n)$. Thus,

$$\text{Prob}\,(\text{LO}(x_t) < n) = \sum_{i=0}^{n-1} \binom{t}{i} \left(\frac{1}{n}\right)^i \left(1 - \frac{1}{n}\right)^{t-i}.$$

Together this yields

$$\text{E}\,(\text{LO}(x_b) \mid x_0 = 0^n) = \frac{1}{n} \sum_{t=0}^{b-1} \sum_{i=0}^{n-1} \binom{t}{i} \left(\frac{1}{n}\right)^i \left(1 - \frac{1}{n}\right)^{t-i}.$$

We consider the case $b \leq (1-\varepsilon)n^2$ for some positive constant $\varepsilon < 1$. For a lower bound on $\text{E}\,(\text{LO}(x_b) \mid x_0 = 0^n)$ we need a lower bound on $\text{Prob}\,(\text{LO}(x_t) < n)$ for all $t \leq b$. Instead we consider an upper bound on $\text{Prob}\,(\text{LO}(x_t) \geq n)$ for all $t \leq b$. Remember that $\text{LO}(x_t) = \sum_{i=1}^{t} S_i$.

We see that the S_i^* and S_i fulfil the conditions of Lem. 1.20 in [16]. Thus $\text{Prob}\,(\text{LO}(x_t) \geq n) \leq \text{Prob}\left(\sum_{i=1}^{t} S_i^* \geq n\right)$ holds. Since the S_i^* are completely independent we can apply Chernoff bounds [17]. Note that $\text{Prob}\left(\sum_{i=1}^{t} S_i^* \geq n\right)$ is monotonically increasing in t. Thus, we replace t by $t^* := \max\{t, n^2/2\}$. This yields

$$\begin{aligned}
\text{Prob}\,(\text{LO}(x_t) \geq n) &\leq \text{Prob}\left(\sum_{i=1}^{t} S_i^* \geq n\right) \\
&= \text{Prob}\left(\sum_{i=1}^{t} S_i^* \geq \left(1 + \left(\frac{n^2}{t} - 1\right)\right)\frac{t}{n}\right) \\
&\leq e^{-(t^*/n)\cdot\left(\left(n^2/t^*\right)-1\right)^2/3} \\
&\leq e^{-(n/2)\cdot((1/(1-\varepsilon))-1)^2/3} \\
&= e^{-\Theta(n)}
\end{aligned}$$

and $\text{E}\,(\text{LO}(x_b) \mid x_0 = 0^n) \geq (b/n)\cdot\left(1 - e^{-\Omega(n)}\right)$ follows. Together this implies the result when converting from $\text{LO}(x_b)$ back to $\text{P}(x_b)$. □

As before we present empirical results for $n = 1000$ over 100 runs. The results for deterministic initialisation in 0^n are depicted in Figure 5. Since

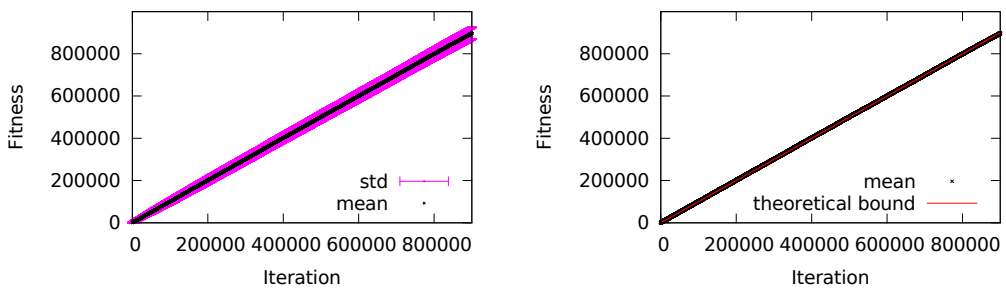

Figure 5: Empirical results for P, $n = 1000$, and deterministic initialisation in 0^{1000}: average values together with their standard deviations (left) and with the theoretical bounds (right).

we have $n = 1000$ terms of order $e^{-\Omega(n)}$ have no influence on the numerical values and we simply omit them. Thus, the upper and lower bound we have are actually equal. We see that for R the variance increases with increasing budget. With respect to the average values we again have a perfect match with the theoretical bounds.

3.4. Random Local Search on LO

After this preparation we consider RLS on LO where the initial bit string x_0 is generated uniformly at random. At any time step $i \in \mathbb{N}_0$ we have that the bits $x_i[v+2]\ x_i[v+3] \cdots x_i[n]$ are distributed uniformly at random in $\{0,1\}^{n-v-1}$ where $\mathrm{LO}(x_i) = v$. For standard bit mutations this was already stated in [9]. Before we prove our main result, we show that this also holds for RLS.

Lemma 8. *Consider RLS on* LO*. Let* $v := \mathrm{LO}(x_i)$ *for some* $i \in \mathbb{N}_0$*,* $x_i \in \{0,1\}^n$*. Then, the bits* $x_i[v+2]\ x_i[v+3] \cdots x_i[n]$ *are distributed uniformly at random in* $\{0,1\}^{n-v-1}$*.*

Proof. We prove the lemma by induction over i. For $i = 0$ it holds since the initial bit string is generated uniformly at random. Assume that it holds for i. Let $\mathrm{LO}(x_{i+1}) = v$, let $x'_{i+1} = x_{i+1}[v+2]x_{i+1}[v+3] \cdots x_{i+1}[n]$, let $x'_i = x_i[v+2]$ $x_i[v+3] \cdots x_i[n]$. If $\mathrm{LO}(x_i) \neq \mathrm{LO}(x_{i+1})$ the bits in x'_i and x'_{i+1} are identical and thus the distribution does not change. If $\mathrm{LO}(x_i) = \mathrm{LO}(x_{i+1})$ the bits in x'_i may be subject to mutation. Let $\mathrm{mut}(z')$ denote the result of applying

mutation to z'. For any $y' \in \{0,1\}^{n-v-1}$ we have

$$\text{Prob}\left(x'_{i+1} = y'\right) = \sum_{z' \in \{0,1\}^{n-v-1}} \text{Prob}\left((x'_i = z') \wedge (\text{mut}(z') = y')\right).$$

Furthermore,

$$\sum_{z' \in \{0,1\}^{n-v-1}} \text{Prob}\left((x'_i = z') \wedge (\text{mut}(z') = y')\right)$$
$$= \sum_{z' \in \{0,1\}^{n-v-1}} \text{Prob}\left(x'_i = z'\right) \cdot \text{Prob}\left(\text{mut}(z') = y'\right)$$

since mutation is carried out independently of the bit string. By assumption we have $\text{Prob}\left(x'_i = z'\right) = 2^{-(n-v-1)}$ for all z'. Moreover, $\text{Prob}\left(\text{mut}(z') = y'\right) = \text{Prob}\left(\text{mut}(y') = z'\right)$ for all y', z' is a property of the mutation operator. Together this yields for all $y' \in \{0,1\}^{n-v-1}$ that $\text{Prob}\left(x'_{i+1} = y'\right) = 2^{-(n-v-1)} \cdot \sum_{z' \in \{0,1\}^{n-v-1}} \text{Prob}\left(\text{mut}(y') = z'\right) = 2^{-(n-v-1)}$ holds since

$$\sum_{z' \in \{0,1\}^{n-v-1}} \text{Prob}\left(\text{mut}(y') = z'\right) = 1$$

as it describes a probability distribution. □

We are now ready to prove our main result for LO. We consider a budget $b = (1-\beta)n^2$ for any β with $(1/2)+\beta' < \beta < 1$ where β' is a positive constant. Note that any budget $b \leq (1-\varepsilon)n^2$ (ε a positive constant) can be expressed this way since β is not assumed to be a constant. We prove a lower bound that is $\Theta(1)$ for any budget $b = O(n)$. For any budget $b = \omega(n)$ we have an expected function value of $1 + (2b/n) - o(1)$.

Theorem 9. *Let the budget $b = (1-\beta)n^2$ for any β with $(1/2)+\beta' < \beta < 1$ where β' is a positive constant. With $x_0 \in \{0,1\}^n$ uniformly at random,*

$$1 + \frac{2b}{n} - 2^{-\Omega((1-\beta)n)} \leq E(\text{LO}(x_b)) \leq 1 + \frac{2b}{n} - 2^{-n}$$

holds for RLS on LO.

Proof. Let $S_t := \mathrm{LO}(x_{t+1}) - \mathrm{LO}(x_t)$ for $t < b$. We have

$$\mathrm{E}\left(\mathrm{LO}(x_b)\right) = \mathrm{E}\left(\mathrm{LO}(x_0) + \sum_{t=0}^{b-1} S_t\right) = \mathrm{E}\left(\mathrm{LO}(x_0)\right) + \sum_{t=0}^{b-1} \mathrm{E}\left(S_t\right).$$

We observe that

$$\begin{aligned}
\mathrm{E}\left(S_t\right) &= \mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) < n\right) \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \\
&\quad + \mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) = n\right) \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) = n\right) \\
&= \mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) < n\right) \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \\
&= \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \cdot \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \cdot \mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) = i\right)
\end{aligned}$$

holds where the last equation makes use of the law of total expectation. To compute $\mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) = i\right)$ we note that in order to have an increase in the number of leading 1-bits the left-most 0-bit must flip (probability $1/n$). The increase equals j if the following $j-1$ bits all are 1-bits and the jth bit is a 0-bit, if there are at least j subsequent bits. In case there are only $j-1$ subsequent bits it suffices if these are all 1-bits. Since the bits are uniformly distributed (Lemma 8) we have probability $(1/2)^j$ for the first case and $(1/2)^{j-1} = (1/2)^{n-i-1}$ in the second case. This implies

$$\begin{aligned}
\mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) = i\right) &= \left(\sum_{j=1}^{n-i-1} j \cdot \frac{1}{n} \cdot \left(\frac{1}{2}\right)^j\right) + (n-i) \cdot \frac{1}{n} \cdot \left(\frac{1}{2}\right)^{n-i-1} \\
&= \frac{2 - (1/2)^{n-i-1}}{n}
\end{aligned}$$

where the first summand in the first equality covers the cases of at least j subsequent bits and the second summand covers the remaining case of $j-1$ subsequent bits.

Plugging this in yields

$$\begin{aligned}
\mathrm{E}\left(S_t\right) &= \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \cdot \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \cdot \frac{2 - (1/2)^{n-i-1}}{n} \\
&= \left(\frac{2}{n} \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \cdot \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right)\right) \\
&\quad - \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \cdot \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \cdot \frac{(1/2)^{n-i-1}}{n} \\
&= \left(\frac{2}{n} \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right)^2\right) \\
&\quad - \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \cdot \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \cdot \frac{(1/2)^{n-i-1}}{n} \\
&= \frac{2}{n} \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \\
&\quad \cdot \left(\mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) - \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \cdot \left(\frac{1}{2}\right)^{n-i}\right).
\end{aligned}$$

We observe that $\mathrm{E}\left(S_t\right) \leq 2/n$ and

$$\mathrm{E}\left(\mathrm{LO}(x_b)\right) = \mathrm{E}\left(\mathrm{LO}(x_0)\right) + \sum_{t=0}^{b-1} \mathrm{E}\left(S_t\right) \leq \mathrm{E}\left(\mathrm{LO}(x_0)\right) + \frac{2b}{n}$$

follows. We have

$$\mathrm{E}\left(\mathrm{LO}(x_0)\right) = n \cdot 2^{-n} + \sum_{i=1}^{n-1} i \cdot 2^{-(i+1)} = 1 - 2^{-n}$$

and have $\mathrm{E}\left(\mathrm{LO}(x_b)\right) \leq 1 + 2b/n - 2^{-n}$ as a direct consequence.

For a lower bound we need a lower bound on

$$\begin{aligned}
\sum_{t=0}^{b-1} \mathrm{E}\left(S_t\right) &= \sum_{t=0}^{b-1} \frac{2}{n} \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \\
&\quad \cdot \left(\mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) - \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \cdot \left(\frac{1}{2}\right)^{n-i}\right).
\end{aligned}$$

To this end we want to prove for some positive constant $\delta < 1$ a lower bound on $\text{Prob}\,(\text{LO}(x_t) < (1-\delta)n)$. We claim that $\text{Prob}\,(\text{LO}(x_t) < (1-\delta)n) = 1 - 2^{-\Omega((1-\beta)n)}$ holds for any constant $\delta < 1$ and any β with $(1/2) + \beta' < \beta < 1$ where β' is a positive constant. We formulate this claim as Lemma 10 and prove this later. Note that this bound implies $\text{Prob}\,(\text{LO}(x_t) < n) = 1 - 2^{-\Omega((1-\beta)n)}$. Moreover, we obtain

$$
\begin{aligned}
&\sum_{i=0}^{n-1} \text{Prob}\,(\text{LO}(x_t) = i) \left(\frac{1}{2}\right)^{n-i} \\
&= \sum_{i=0}^{(1-\delta)n} \text{Prob}\,(\text{LO}(x_t) = i)\, 2^{i-n} + \sum_{i=(1-\delta)n+1}^{n-1} \text{Prob}\,(\text{LO}(x_t) = i)\, 2^{i-n} \\
&\leq \left(\left(\frac{1}{2}\right)^{n-(1-\delta)n} \sum_{i=0}^{(1-\delta)n} \text{Prob}\,(\text{LO}(x_t) = i) \right) \\
&\quad + \left(\frac{1}{2} \cdot \sum_{i=(1-\delta)n+1}^{n-1} \text{Prob}\,(\text{LO}(x_t) = i) \right) \\
&\leq 2^{-\delta n} + \frac{1}{2} \cdot \text{Prob}\,(\text{LO}(x_t) > (1-\delta)n) \\
&= 2^{-\Omega((1-\beta)n)}
\end{aligned}
$$

and have

$$
\begin{aligned}
\sum_{t=0}^{b-1} \text{E}\,(S_t) &= \sum_{t=0}^{b-1} \frac{2}{n} \cdot \text{Prob}\,(\text{LO}(x_t) < n) \\
&\quad \cdot \left(\text{Prob}\,(\text{LO}(x_t) < n) - \sum_{i=0}^{n-1} \text{Prob}\,(\text{LO}(x_t) = i) \cdot \left(\frac{1}{2}\right)^{n-i} \right) \\
&\geq \sum_{t=0}^{b-1} \frac{2}{n} \left(1 - 2^{-\Omega((1-\beta)n)}\right) \left(\left(1 - 2^{-\Omega((1-\beta)n)}\right) - 2^{-\Omega((1-\beta)n)}\right) \\
&= \frac{2b}{n} \left(1 - 2^{-\Omega((1-\beta)n)}\right) = \frac{2b}{n} - 2^{-\Omega((1-\beta)n)}
\end{aligned}
$$

as a direct consequence. Together with our result about initialisation,

$\mathrm{E}(\mathrm{LO}(x_0)) = 1 - 2^{-n}$, we obtain $\mathrm{E}(\mathrm{LO}(x_b)) \geq 1 + (2b/n) - 2^{-\Omega((1-\beta)n)}$ and the result follows. $\square$

We are left with the proof of the lower bound on $\mathrm{Prob}(\mathrm{LO}(x_t) < (1-\delta)n)$ for some positive constant $\delta < 1$.

Lemma 10. *Consider RLS on* LO*. Let* $t \leq (1-\beta)n^2$ *for any* β *with* $(1/2) + \beta' < \beta < 1$ *where* β' *is a positive constant. Moreover, let* $\delta < 1$ *be a constant. Then,*

$$\mathit{Prob}(\mathrm{LO}(x_t) < (1-\delta)n) = 1 - 2^{-\Omega((1-\beta)n)}$$

holds.

Proof. We call a step a *leap* if the number of leading 1-bits leaps forward, i. e., $\mathrm{LO}(x_i) < \mathrm{LO}(x_{i+1})$. As above we have that for any number of steps t the expected number of leaps is bounded above by t/n. Moreover, using Chernoff bounds, we can bound the probability to have at least $(1+\alpha)t/n$ leaps by $e^{-\alpha^2 t/(3n)}$ for any positive constant $\alpha \leq 1$.

For any time step t there are two different sources for 1-bits contributing to $\mathrm{LO}(x_t)$. One is the number of leading 1-bits that are present initially, i. e., $\mathrm{LO}(x_0)$. The second is the number of bits added to the leading 1-bits in a leap. We first consider the latter source.

Consider the random process $x_0, x_1, x_2, \ldots$ generated by RLS on LO. Recall that we call a step a leap if the number of leading 1-bits leaps forward, i. e., $\mathrm{LO}(x_i) < \mathrm{LO}(x_{i+1})$. In such a leap the number of leading 1-bits is increased by a random number N with $N \in \{1, 2, \ldots, n - \mathrm{LO}(x_i)\}$. Since the bits $x_i[j]$ with $j > \mathrm{LO}(x_i) + 1$ are distributed uniformly at random (see Lemma 8) we know the precise distribution of N. We are interested in the distribution of the sum of these random variables N for the first l leaps, $N_1 + N_2 + \cdots + N_l$. To this end consider an infinite bit string $y = y[1]y[2]\cdots$ where each bit is chosen independently with equal probability from $\{0, 1\}$. Let v_i be the function value before the i-th leap, so that in the i-th leap the function value is increased from v_i to $v_i + N_i = v_{i+1}$. In the i-th leap we consider N_i bits at position $v_i + 2$, $v_i + 3$, $\ldots$, $v_i + N_i + 1$ and map these to the bits $\left(\sum_{j=1}^{i-1} N_j\right) + 1$, $\left(\sum_{j=1}^{i-1} N_j\right) + 2$, $\ldots$, $\left(\sum_{j=1}^{i-1} N_j\right) + N_i$ in the infinite bit string. This is illustrated in Figure 6.

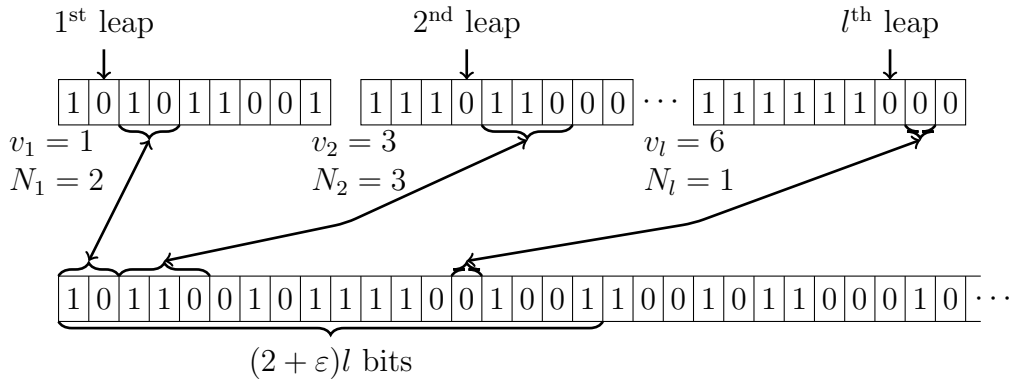

Figure 6: Illustration of mapping of random bits to bound increase in function value.

Consider the first $\sum_{i=1}^{l} N_i$ bits in y for any $l \in \mathbb{N}$. The number of 0-bits in these bits equals l since among the N_i bits there is always exactly one 0-bit, the final bit in the sequence. We observe that for any $l, k \in \mathbb{N}$ we have

$$\text{Prob}\left(\sum_{i=1}^{l} N_i \geq k\right) = \text{Prob}\left(\sum_{i=1}^{k} y[i] \geq k - l\right)$$

since $\sum_{i=1}^{k} y[i]$ denotes the number of 1-bits in the first k bits of y and having at least $k - l$ 1-bits corresponds to having at most l 0-bits. Using that the bits in y are independently and uniformly distributed, the following holds for any $\varepsilon > 0$ and any l by application of Chernoff bounds:

$$\begin{aligned}
\text{Prob}\left(\sum_{i=1}^{l} N_i \geq (2+\varepsilon)l\right) &= \text{Prob}\left(\sum_{i=1}^{(2+\varepsilon)l} y[i] \geq (1+\varepsilon)l\right) \\
&= \text{Prob}\left(\sum_{i=1}^{(2+\varepsilon)l} y[i] \geq \left(1 + \frac{\varepsilon}{2+\varepsilon}\right) \cdot \left(1 + \frac{\varepsilon}{2}\right) l\right) \\
&\leq e^{-(1+\varepsilon/2)(\varepsilon/(2+\varepsilon))^2 l/3} \\
&= e^{-\left(\varepsilon^2/(12+6\varepsilon)\right) l}
\end{aligned}$$

We need a lower bound on $\text{Prob}\left(\text{LO}(x_t) < (1-\delta)n\right)$ and instead prove an upper bound on $\text{Prob}\left(\text{LO}(x_t) \geq (1-\delta)n\right)$. Let A denote the event $\text{LO}(x_t) \geq$

$(1-\delta)n$, let B denote the event $\mathrm{LO}(x_0) \geq (1-\gamma)(1-\delta)n$, let C denote the event that the increase in the number of leading 1-bits in t generations is at least $\gamma(1-\delta)n$ for some constant $\gamma < 1$. Clearly, $\mathrm{Prob}\,(A) \leq \mathrm{Prob}\left(B \vee \left(\overline{B} \wedge C\right)\right) \leq \mathrm{Prob}\,(B \vee C) \leq \mathrm{Prob}\,(B) + \mathrm{Prob}\,(C)$ holds. We have

$$\mathrm{Prob}\,(B) = \sum_{i=(1-\gamma)(1-\delta)n}^{n} \mathrm{Prob}\,(\mathrm{LO}(x_0) = i)$$
$$\leq \sum_{i=(1-\gamma)(1-\delta)n}^{n} \left(\frac{1}{2}\right)^i < 2^{1-(1-\gamma)(1-\delta)n}$$

and need an upper bound $\mathrm{Prob}\,(C)$. We consider $t \leq (1-\beta)n^2$ steps (for some positive $\beta < 1$) and have that with probability $1-e^{-\alpha^2 t/(3n)}$ the number of leaps is bounded above by $(1+\alpha)t/n$ for any positive constant $\alpha < 1$. In at most $(1+\alpha)t/n$ leaps the number of leading 1-bits is increased by at most $(2+\varepsilon)(1+\alpha)t/n$ with probability at least $1 - e^{-\left(\varepsilon^2/(12+6\varepsilon)\right)(1+\alpha)t/n}$ for any positive constant ε.

Remember that $S_t = \mathrm{LO}(x_{t+1}) - \mathrm{LO}(x_t)$. Thus, the increase in the number of leading 1-bits in t steps equals $\sum_{i=0}^{t-1} S_i$. Recall that C is the event that $\sum_{i=0}^{t-1} S_i \geq \gamma(1-\delta)n$. For any constants $0 < \alpha < 1$, $\varepsilon > 0$ and any $t \leq (1-\beta)n^2$ with $0 < \beta < 1$ we have

$$\mathrm{Prob}\left(\sum_{i=0}^{t-1} S_i \geq (2+\varepsilon)(1+\alpha)t/n\right) \leq e^{-\left(\left(\varepsilon^2(1+\alpha)\right)/(12+6\varepsilon)\right)\cdot(t/n)}.$$

We observe that the probability $\mathrm{Prob}\left(\sum_{i=0}^{t-1} S_i \geq k\right)$ is monotonically increasing in t for any fixed k. Thus,

$$\mathrm{Prob}\left(\sum_{i=0}^{t-1} S_i \geq (2+\varepsilon)(1+\alpha)(1-\beta)n\right) \leq e^{-\left(\left(\varepsilon^2(1+\alpha)\right)/(12+6\varepsilon)\right)\cdot(1-\beta)n}$$

for any $t \leq (1-\beta)n^2$ with $0 < \beta < 1$. For estimating $\mathrm{Prob}\,(C)$ we need

$$(2+\varepsilon)(1+\alpha)(1-\beta)n = \gamma(1-\delta)n. \tag{1}$$

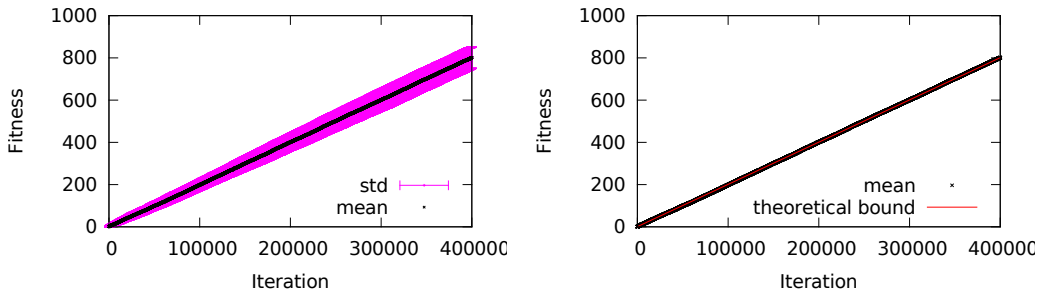

Figure 7: Empirical results for LO, $n = 1000$, and random initialisation: average values together with their standard deviations (left) and with the theoretical bounds (right).

The above holds for arbitrary constants $0 < \alpha, \gamma < 1$ and $\varepsilon > 0$. We set $\alpha := \varepsilon$ and $\gamma := 1 - \varepsilon$. For a given $1/2 < \beta < 1$ we solve equation (1) for ε and obtain

$$\varepsilon = \frac{4 - 3\beta - \delta - \sqrt{12 - 12\beta + \beta^2 - 12\delta + 10\beta\delta + \delta^2}}{2(\beta - 1)}.$$

For $\beta > (1 + \delta)/2$ we observe $0 < \varepsilon < 1$ for $\delta < 1$.

We summarise what we have and obtain

$$\begin{aligned}\text{Prob}\,(\text{LO}(x_t) < (1 - \delta)n) &\geq 1 - 2^{1-\varepsilon(1-\delta)n} - e^{-\left(\left(\varepsilon^2(1+\varepsilon)\right)/(12+6\varepsilon)\right)\cdot(1-\beta)n} \\ &= 1 - 2^{-\Omega((1-\beta)n)}\end{aligned}$$

with ε as above for any $t \leq (1 - \beta)n^2$ with $\beta > (1 + \delta)/2$. □

As before we present empirical results for $n = 1000$ over 100 runs. The results for random initialisation are depicted in Figure 7. We see that the variance is even more increased than it is for R (compare Figure 8). Since with $n = 1000$ we can safely omit terms of order $e^{-\Omega(n)}$ we have matching upper and lower bounds. Again we have a perfect match of the empirical values with the theoretical bounds.

3.5. Random Local Search on R

For random local search we can draw on our results on OM and P. After random initialisation RLS will behave like it does on OM with probability very close to 1. After finding a search point of the form $1^i 0^{n-i}$ for the first time it behaves like it does on P. Taking this into account it is not difficult to come up with an upper bound for the expected performance.

Theorem 11. *Let the budget $b \leq (1-\varepsilon)n^2$ for some positive constant $\varepsilon < 1$. With $x_0 \in \{0,1\}^n$ uniformly at random,*

$$E(\mathrm{R}(x_b)) \leq \frac{n}{2} + \frac{n}{2}\left(1 - \left(1 - \frac{1}{n}\right)^b\right) + \frac{b}{n} + e^{-\Omega(n)}$$

holds.

Proof. Once RLS has found a search point of the form $1^i 0^{n-i}$ all subsequent search points will have the same form. For such search points RLS behaves on R in the same way as on P. Thus, we can re-use results from the proof of Theorem 7. For such search points the number of 1-bits equals the number of leading 1-bits. We have $\mathrm{R}\left(1^i 0^{n-i}\right) = n + i$ and $\mathrm{P}\left(1^i 0^{n-i}\right) = n \cdot i - i$.

If RLS is started in 0^n, we have $\mathrm{E}\left(\mathrm{P}\left(x_b\right) \mid x_0 = 0^n\right) \leq b - (b/n)$ from Theorem 7. Under the same condition this translates to $\mathrm{E}\left(\mathrm{R}\left(x_b\right) \mid x_0 = 0^n\right) \leq n + b/n$ here.

Note that here we are concerned with an initial search point that is selected uniformly at random from $\{0,1\}^n$. We have $\mathrm{Prob}\left(\mathrm{R}(x_0) > n\right) = n \cdot 2^{-n}$. Therefore, this contributes $n \cdot 2^{-n} \cdot 2n = e^{-\Omega(n)}$ to the expected function value.

In order to reach the ridge (i. e., the sequence of points 0^n, 10^{n-1}, 110^{n-2}, $\ldots$, 1^n) the $(n - \mathrm{OM})$-part of R needs to be optimised. Using the result on OM (Theorem 5) we can replace this by $(n/2) + (n/2)\left(1 - (1 - 1/n)^b\right)$. This yields the claimed upper bound. □

A lower bound is more difficult to obtain. Let T be the random point of time when the ridge is first encountered. Up to this point the expected function value is given as for OM. After this time it is at least $n + (b/n) - e^{-\Omega(n)}$ like on P. We use precisely this idea to derive the following lower bound.

Theorem 12. *Let the budget $b \leq (1-\varepsilon)n^2$ for some positive constant $\varepsilon < 1$. With $x_0 \in \{0,1\}^n$ uniformly at random,*

$$E(\mathrm{R}(x_b)) \geq \max\left\{\frac{n}{2} + \frac{n}{2}\left(1 - \left(1 - \frac{1}{n}\right)^b\right), \max\left\{0, 1 - \frac{1}{n^{(t/(n \ln n))-1}}\right\} \cdot \left(n + \frac{b-t}{n} - e^{-\Omega(n)}\right)\right\}$$

holds for any $t \leq b$.

Proof. The term $((n/2) + (n/2)\left(1 - (1 - 1/n)^b\right)$ in the maximum is correct since RLS performs on R no worse than on OM. Therefore, we concentrate on the other part of the maximum. Let $p(t)$ denote a lower bound on the probability that the ridge is encountered after at most t steps. Then $p(t) \cdot \left(n + ((b-t)/n) - e^{-\Omega(n)}\right)$ is a lower bound on the expected function value after a total of b steps due to our result on P (Theorem 7). With random initialisation the expected number of steps needed to locate the ridge is smaller than the expected number of coupons needed to obtain a complete set of coupons in the coupon's collector problem (see [14] for the problem and the bound we use in the following). Therefore, we can use the well-known bound $n^{-\beta+1}$ on the probability to need more than $\beta n \ln n$ coupons to have a complete collection that holds for all $\beta > 0$. Together this yields the result. □

Again we present empirical results for $n = 1000$ and random initialisation over 100 runs. The results are depicted in Figure 8. Additionally, we show more detailed results in Figure 9 for the area where it becomes more likely to already have reached the ridge. The size of the standard deviations is small in the beginning when RLS is confronted with the OM-part of R. It increases later to similar values like on P on the part that is P-like. In this part our lower and upper bounds are quite tight and we see that the empirical results are within these bounds. Only in the part where there is the random change from the OM-part to the P-part of R our bounds are a bit more loose. Figure 9 zooms into this area. We see that the difference between the lower and upper bound for $n = 1000$ is less than 15 for function values of more than 1000. The average of the empirical results is well within these bounds.

4. Fixed Budget Results for (1+1) EA

We consider the (1+1) EA on LO with random initialisation. We concentrate on this case because this random process is very well analysed and understood [18]. Moreover, the optimisation time of the (1+1) EA on LO is very much concentrated around its expected value [9]. We will see that in spite of these advantages and our preparations for this in Section 3.4 the analysis is still difficult.

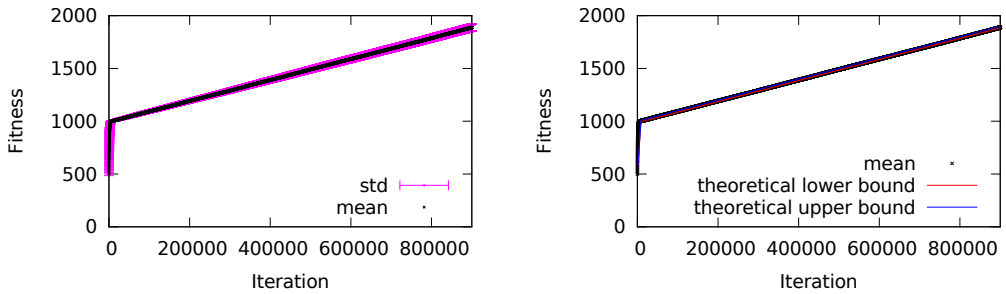

Figure 8: Empirical results for R, $n = 1000$, and random initialisation: average values together with their standard deviations (left) and with the theoretical bounds (right).

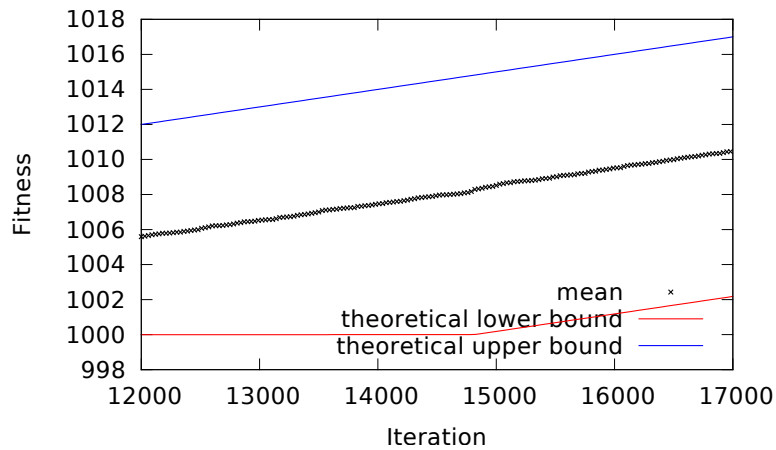

Figure 9: Empirical results for R, $n = 1000$, and random initialisation: average values together with the theoretical bounds.

We observe that for any x_t the bits $x_t[v+2]x_t[v+3]\cdots x_t[n]$ with $v = \mathrm{LO}(x_t)$ are distributed independently uniformly. The property of local mutations employed in RLS that is crucial for the proof, $\forall x, y\colon \mathrm{Prob}\,(\mathrm{mut}(x) = y) = \mathrm{Prob}\,(\mathrm{mut}(y) = x)$, also holds for the standard bit mutations employed in the (1+1) EA [9]. Thus, $\mathrm{E}\,(\mathrm{LO}(x_{t+1}) \mid x_t = x)$ is always at least as large for RLS as it is for the (1+1) EA. This is true since the probability for increasing the function value equals $1/n$ for RLS while it equals $(1/n) \cdot (1 - 1/n)^v$ (with $v = \mathrm{LO}(x)$) for the (1+1) EA and the distribution of the other bits is identical. Thus, the expected function value for RLS is an upper bound for the expected function value for the (1+1) EA. We conclude that

$\mathrm{E}\left(\mathrm{LO}(x_b)\right) \leq 1 + (2b/n) - 2^{-n}$ holds for any budget $b = (1-\beta)n^2$ for any β with $(1/2) + \beta' < \beta < 1$ where β' is a positive constant. One may speculate that due to the considerably smaller success probability this upper bound is not tight for the (1+1) EA. We prove that it actually is tight for budgets b that are asymptotically smaller than the expected optimisation time.

Theorem 13. *Let the budget $b = (1-\beta)n^2/\alpha(n)$ for any β with $(1/2)+\beta' < \beta < 1$ where β' is a positive constant and $\alpha(n) = \omega(1)$, $\alpha(n) \geq 1$. With $x_0 \in \{0,1\}^n$ uniformly at random,*

$$E\left(\mathrm{LO}(x_b)\right) = 1 + \frac{2b}{n} - O\left(\frac{b}{n\alpha(n)}\right) = 1 + \frac{2b}{n} - o\left(\frac{b}{n}\right)$$

holds for (1+1) EA on LO.

Proof. We re-use the notation from the previous section and see that

$$\mathrm{E}\left(\mathrm{LO}(x_b)\right) = 1 - 2^{-n} + \sum_{t=0}^{b-1} \mathrm{E}\left(S_t\right)$$

and

$$\mathrm{E}\left(S_t\right) = \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) = i\right)$$

both hold for the (1+1) EA, too. Due to the different mutation we have in particular a change in $\mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) = i\right)$. We have, similar to the proof of Theorem 9 and again using Lemma 8,

$$\begin{aligned}
&\mathrm{E}\left(S_t \mid \mathrm{LO}(x_t) = i\right) \\
&\quad = \sum_{j=1}^{n-i-1} j \cdot \left(1 - \frac{1}{n}\right)^i \frac{1}{n} \cdot \left(\frac{1}{2}\right)^j + (n-i) \cdot \left(1 - \frac{1}{n}\right)^i \frac{1}{n} \cdot \left(\frac{1}{2}\right)^{n-i-1}
\end{aligned}$$

and come to

$$\begin{aligned}
\mathrm{E}\left(S_t\right) = &\left(\frac{2}{n} \cdot \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \cdot \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \left(1 - \frac{1}{n}\right)^i\right) \\
&- \mathrm{Prob}\left(\mathrm{LO}(x_t) < n\right) \cdot \sum_{i=0}^{n-1} \mathrm{Prob}\left(\mathrm{LO}(x_t) = i\right) \cdot \frac{(1/2)^{n-i-1}}{n} \cdot \left(1 - \frac{1}{n}\right)^i
\end{aligned}$$

as before. Remember that the performance of RLS is an upper bound on the performance of the (1+1) EA, thus we have

$$\operatorname{Prob}(\mathrm{LO}(x_t) < n) \cdot \sum_{i=0}^{n-1} \operatorname{Prob}(\mathrm{LO}(x_t) = i) \cdot \frac{(1/2)^{n-i-1}}{n} \left(1 - \frac{1}{n}\right)^i = 2^{-\Omega((1-\beta)n)}$$

as before for any $t = (1-\beta)n^2$ for any (not necessarily constant) β with $(1/2) + \beta' < \beta < 1$ where $\beta' > 0$ is a constant.

$$\mathrm{E}(S_t) = \left(\frac{2}{n} \cdot \sum_{i=0}^{n-1} \operatorname{Prob}(\mathrm{LO}(x_t) = i) \left(1 - \frac{1}{n}\right)^i\right) - 2^{-\Omega((1-\beta)n)}$$

follows. We observe that the $(1-1/n)^i$ behave asymptotically differently for $i = o(n)$ and $i = \Omega(n)$. Thus, it makes sense to make a case distinction. We consider the (1+1) EA with a small budget b that is $b(n) = o(n^2)$ here. The other case is dealt with in Theorem 14.

We have

$$\begin{aligned}\sum_{i=0}^{n-1} \operatorname{Prob}(\mathrm{LO}(x_t) = i) \left(1 - \frac{1}{n}\right)^i &\geq \sum_{i=0}^{n/\alpha(n)} \operatorname{Prob}(\mathrm{LO}(x_t) = i) \left(1 - \frac{1}{n}\right)^i \\ &\geq \left(1 - \frac{1}{n}\right)^{n/\alpha(n)} \cdot \operatorname{Prob}(\mathrm{LO}(x_t) \leq n/\alpha(n))\end{aligned}$$

for any function $\alpha\colon \mathbb{N} \to \mathbb{R}^+$ with $\alpha(n) = \omega(1)$.

For RLS and $t = (1-\beta)n^2$ with β as before we proved

$$\operatorname{Prob}(\mathrm{LO}(x_t) < (1-\delta)n) = 1 - 2^{-\Omega((1-\beta)n)}.$$

We observe that we get basically the same statement for any function $\alpha\colon \mathbb{N} \to \mathbb{R}^+$ with $\alpha(n) = \omega(1)$, i. e., $\operatorname{Prob}(\mathrm{LO}(x_t) < n/\alpha(n)) = 1 - 2^{-\Omega((1-\beta)n/\alpha(n))}$.

Since this holds for RLS, it holds for the (1+1) EA, too. Together we have

$$\begin{aligned}
\mathrm{E}\left(S_t\right) &= \left(\frac{2}{n}\cdot\left(1-\frac{1}{n}\right)^{n/\alpha(n)}\cdot\left(1-2^{-\Omega((1-\beta)n/\alpha(n))}\right)\right)-2^{-\Omega((1-\beta)n)}\\
&= \left(\frac{2}{n}\cdot\left(1-\frac{1}{n}\right)^{(n-1)\cdot n/((n-1)\alpha(n))}\right)-2^{-\Omega((1-\beta)n/\alpha(n))}\\
&\geq \left(\frac{2}{n}\cdot e^{-n/((n-1)\alpha(n))}\right)-2^{-\Omega((1-\beta)n/\alpha(n))}\\
&\geq \left(\frac{2}{n}\cdot\left(1-\frac{n}{(n-1)\alpha(n)}\right)\right)-2^{-\Omega((1-\beta)n/\alpha(n))}=\frac{2}{n}-O\left(\frac{1}{n\alpha(n)}\right)
\end{aligned}$$

so that the result follows. □

For budgets that are not so small we start with rather trivial upper and lower bounds. We have already observed that the performance of RLS is an upper bound on the performance of the (1+1) EA. This implies $\mathrm{E}\left(\mathrm{LO}(x_b)\right)\leq 1+(2b/n)-2^{-n}$ here, too. For a lower bound we observe that the term $(1-1/n)^i$ decreases monotonically in i and becomes minimal for $i=n-1$. Thus, the expected increase in the number of leading 1-bits is in each step bounded below by $(2/n)\cdot(1-1/n)^{n-1}\geq 2/(en)$. This establishes

$$\mathrm{E}\left(\mathrm{LO}(x_b)\right)\geq 1-2^{-n}+\frac{2}{en}\sum_{t=0}^{b-1}\mathrm{Prob}\left(\mathrm{LO}(x_t)<n\right)$$

as lower bound. Using the upper bound we see that for any $b\leq(1-\beta)n^2$ for any β with $(1/2)+\beta'<\beta<1$ where $\beta'>0$ is a constant we obtain $\mathrm{E}\left(\mathrm{LO}(x_b)\right)\geq 1+(2b/en)-2^{-\Omega(n)}$ as lower bound. We restrict ourselves to budgets $b=cn^2$ where $0<c<1/2$ is a constant. We observe that there is a multiplicative gap of size $e\approx 2.72$ between the upper and lower bound. We aim to close this gap.

Our approach follows a simple divide and conquer scheme and works as follows. We consider the b steps to be taken and divide them into k intervals of equal size that we call phases. In the beginning of each phase we have upper and lower bounds on the number of leading 1-bits that hold with high probability. In a first step we use these bounds to determine an upper bound on the number of leading 1-bits at the end of this phase that coincides with the beginning of the next phase. With this upper bound and the lower bound

for the beginning of the phase we derive a lower bound for the end of the phase and thus the beginning of the next phase. Formally,

$$U_0 = v,\ U_i = U_{i-1} + (1+\varepsilon)(2s/n)\,(1-1/n)^{\min\{n, L_{i-1}\}}$$
$$L_0 = 0,\ L_i = L_{i-1} + (1-\varepsilon)(2s/n)\,(1-1/n)^{\min\{n, U_i\}},$$

where U_i is the upper bound on the number of leading 1-bits at the end of phase i, L_i is the corresponding lower bound, $v \in \mathbb{N}$ is an upper bound on $\mathrm{LO}(x_0)$ (later chosen sufficiently large to have high probability), $s = b/k$ is the length of each of the k phases, and $\varepsilon > 0$ is a small constant needed to have the bounds hold with high probability. For the initialisation and in each phase we have failure probabilities. We choose v and k in a way that the sum of all failure probabilities is $o(1)$. It is easy to observe that by application of Chernoff bounds in each phase it suffices to have $v = \Omega(\log n)$, $k \leq \delta n/\log n$ for a sufficiently small constant $\delta > 0$ and $\varepsilon > 0$ constant and arbitrarily small. We set $v = \Theta(\log n)$ appropriately. With this we have $L_k \leq \mathrm{LO}(x_b) \leq U_k$ with high probability. We are interested in getting U_k/L_k small, i. e., close to 1.

It is interesting to observe that this divide and conquer approach is better than the trivial bounds even if one does not divide, i. e., if one considers the base case $k = 1$. We start with this case. Due to our assumptions we have $b = cn^2$ where $0 < c < 1/2$ is a constant. We have $U_1 = 2c(1+\varepsilon)n + v$, and consider U_1/n in the limit for $n \to \infty$ and $\varepsilon \to 0$. This yields $\lim\limits_{n\to\infty,\varepsilon\to 0} U_1/n = 2c$ as upper bound. Similarly we obtain $L_1 = 2c(1-\varepsilon)\,(1-1/n)^{2c(1+\varepsilon)n+v}\,n$ and get $\lim\limits_{n\to\infty,\varepsilon\to 0} L_1/n = \lim\limits_{n\to\infty} 2c\,(1-1/n)^{2cn+v} = 2ce^{-2c}$.

We summarise what we have using $k = 1$ so far. We have

$$\mathrm{Prob}\left(2ce^{-2c} \leq \frac{\mathrm{LO}(x_b)}{n} \leq 2c\right) = 1 - o(1)$$

for $b = cn^2$ with $0 < c < 1/2$ constant. These bounds are visualized in Figure 10.

We can repeat the same for $k > 1$. For $k = 2$ we obtain $\lim\limits_{n\to\infty,\varepsilon\to 0} U_2/n = c\left(1 + e^{-ce^{-c}}\right)$ as upper bound and $\lim\limits_{n\to\infty,\varepsilon\to 0} L_2/n = ce^{-c}\left(1 + e^{-ce^{-ce^{-c}}}\right)$ in the very same way. Together this yields

$$\mathrm{Prob}\left(ce^{-c}\left(1 + e^{-ce^{-ce^{-c}}}\right) \leq \frac{\mathrm{LO}(x_b)}{n} \leq c\left(1 + e^{-ce^{-c}}\right)\right) = 1 - o(1)$$

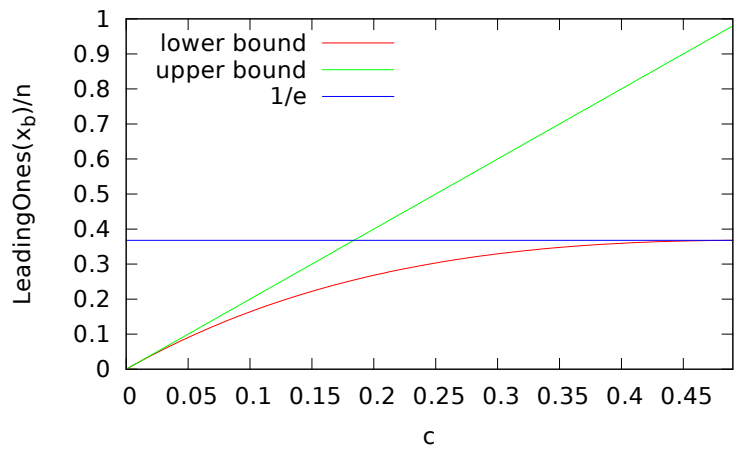

Figure 10: Illustration of upper and lower bounds for $\mathrm{LO}(x_b)/n$ holding with high probability derived with the divide and conquer approach for $k = 1$.

for $k = 2$. While these bounds are tighter they are also more complicated. We illustrate the improvement by plotting the improved bounds for $k = 2$ together with the bounds for $k = 1$ in Figure 11. This complication of the analytical bounds continues due to the recursive structure of our estimation. We therefore abstain from continuing this analytically. For $k > 2$ we perform the same calculations for U_k and L_k but now numerically for some sufficiently large value of n. We present the results of numerical analysis that establish bounds on U_k, L_k and the quotient U_k/L_k for some finite values of k and fixed budget of $b = .45n^2$ and $b = .25n^2$. The numerical results for U_k and L_k together with the trivial bounds are given in Figures 12 and 13, respectively.

We observe that both bounds start to improve quickly with increasing k. For illustration we give upper bounds for the values of U_k/L_k in the following table, i. e., for $c \to 0.5$.

k	1	2	3	4	5	6	7	8	9	10
U_k/L_k	2.47	1.61	1.38	1.28	1.22	1.18	1.16	1.14	1.12	1.11
k	11	12	13	14	15	16	17	18	19	20
U_k/L_k	1.10	1.09	1.08	1.08	1.07	1.07	1.07	1.06	1.06	1.06

We show how the quotients approach their true values with increasing n by plotting the quotient obtained numerically for $n \in \{1\,000, 10\,000, 100\,000\}$ in Figure 14. It becomes obvious that for $n \geq 10\,000$ there is hardly any difference.

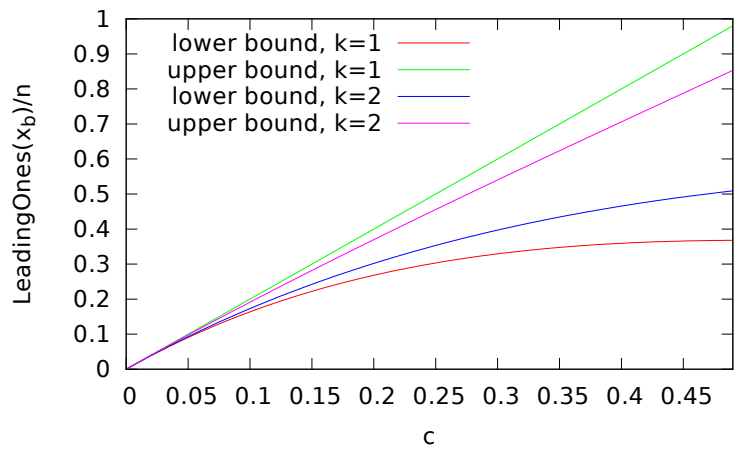

Figure 11: Illustration of upper and lower bounds for $\mathrm{LO}(x_b)/n$ holding with high probability derived with the divide and conquer approach for $k = 1$ and $k = 2$.

Clearly, the most important factor is the budget b. Remember that we assume $b = cn^2$ for some constant $0 < c < 1/2$ in this section. Recall that our divide and conquer approach can be extended until $k = O(n/\log n)$ and becomes better with increasing k. We perform numerical analysis for $k = n/\ln n$, $n = 10\,000$ and values of $c \in \{.01, .02, \ldots, .49\}$. The results are shown in Figure 15 together with the analytical bounds we obtained above for $k = 1$. Moreover, we summarise the analytical bound that we obtained for $k = 2$ in a theorem.

Theorem 14. *Let the budget $b = cn^2$ for any constant c with $0 < c < 1/2$. With $x_0 \in \{0,1\}^n$ uniformly at random,*

$$\mathit{Prob}\left(ce^{-c}\left(1 + e^{-ce^{-ce^{-c}}}\right) \leq \frac{\mathrm{LO}(x_b)}{n} \leq c\left(1 + e^{-ce^{-c}}\right)\right) = 1 - o(1)$$

holds for (1+1) EA on LO*.*

Finally, we present empirical results for $n = 1000$ and random initialisation in Figure 16. We see that for the (1+1) EA the variance in the function value is clearly larger than for random local search. For the comparison we use the bounds from Theorem 13. These bounds are less tight than the bounds for RLS. We decide not to bother about the terms of lesser order in the bounds (that may actually be non-negligible) so that both bounds

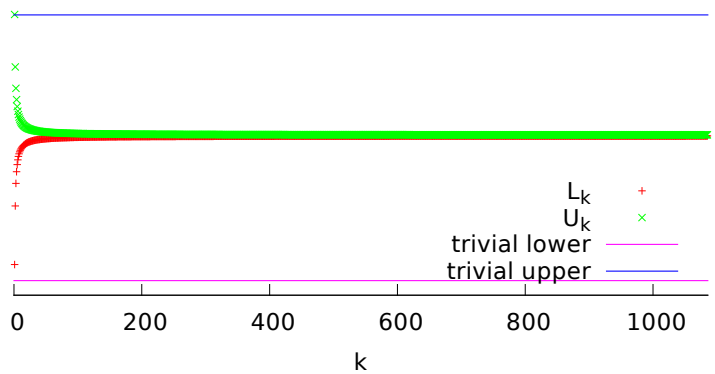

Figure 12: Values of U_k and L_k together with the trivial bounds for $c = 0.45$.

are simply $1 + (2b/n)$. Note that the bounds are only valid for rather small computational budgets $b \leq (1-\beta)n^2/\alpha(n)$ where $\beta > 1/2$ and $\alpha(n) = \omega(1)$. We observe a very good match of the empirical results with the bound for budgets up to 30000. For larger budgets it appears to be the case that the bound is too optimistic. We remember that the budget is systematically optimistic for budgets that are too large and this way get an idea of what $b \leq (1-\beta)n^2/\alpha(n)$ means in practice for $n = 1000$. Note that for say $\beta = 0.6$ and $\alpha(n) = \ln(n)$ we have $(1-\beta)n^2/\alpha(n) \approx 57906$ so that not considering values larger than 60000 is reasonable.

5. Conclusions

The analysis of randomised search heuristic aims at delivering valuable insight into potential and limitations of randomised search heuristics. It is motivated by the desire to be able to apply randomised search heuristics in a more informed and effective way.

Currently, the analysis of randomised search heuristics is dominated by the analysis of the expected optimisation time or the expected approximation time. While this perspective has produced a number of useful results it is in strange contradiction to the way randomised search heuristics are usually applied. This contributes to a substantial gap between theory and practice in the area of randomised search heuristics. We aim at helping to close this gap

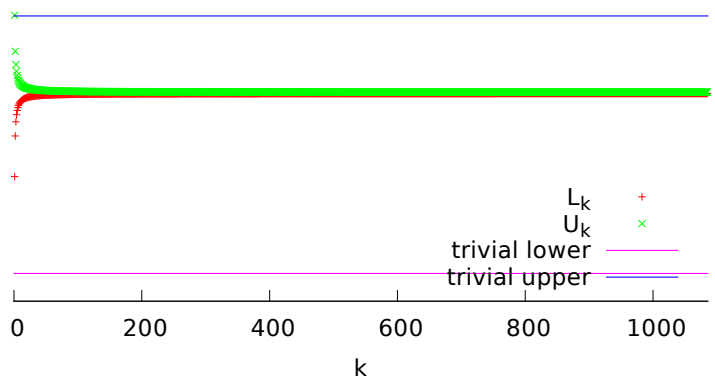

Figure 13: Values of U_k and L_k together with the trivial bounds for $c = 0.25$.

by introducing a different analytical perspective. We analyse the expected function value after a given number b of function evaluations, called budget.

In addition to the introduction of the analytical perspective we present a number of results. For this we consider as the main vehicle random local search. This particularly simple randomised search heuristic facilitates analysis. We exemplify the difficulty of the analysis of less simple randomised search heuristics by considering the (1+1) EA, a very simple evolutionary algorithm.

We obtain results for random local search for five well-known example functions. For OM we prove exact results. For P and LO we prove upper and lower bounds that are asymptotically tight up to an exponentially small summand. For J_k we have on the one hand upper and lower bounds that are tight up to a summand of k and on the other hand an exact recursive formulation that allows us to compute exact results for concrete values of n and k numerically. For R we have lower and upper bounds that are matching for quite small and quite large budgets. They are not tight for values where we expect RLS to reach the ridge of R for the first time. Around this time the progress in function value changes from being OM-like to being P-like. While we have exact bounds for both functions combining them to an exact bound for R that combines both functions is not trivial.

Considering the (1+1) EA we see that the greater variability in mutations presents us with substantial analytical difficulties. We consider the situation

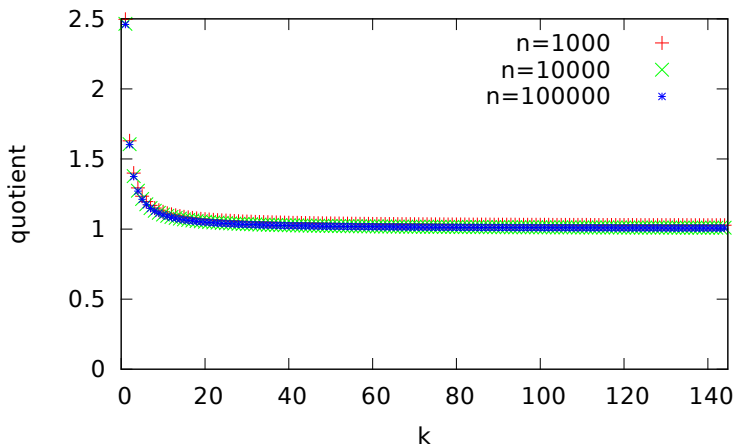

Figure 14: Quotients U_k/L_k for $n \in \{1\,000, 10\,000, 100\,000\}$.

for LO that is particularly well understood. For budgets that are asymptotically smaller than the expected optimisation time for LO we obtain results that are tight up to a summand of smaller order. For budgets that are asymptotically equal to the expected optimisation time we have upper and lower bounds. For this case we present an analytical method that allows to derive upper and lower bounds analytically and numerically. Our analytical lower and upper bounds have a factor of $e^c(1+e^{-ce^{-c}})/(1+e^{-ce^{-ce^{-c}}}) < 1.695$ between them if the budget is $b = cn^2$, $c < 1/2$ constant. We recognise room for improvement since it is desirable to have upper and lower bounds that are tight up to not too large summands.

The introduction of the new perspective to the theory of randomised search heuristics and the presentation of first results is hoped to motivate further research. Clearly, there is a large gap between the results on simple example functions here and insights that are actually useful in practice. The most important and pressing issue for further studies is the development of adequate analytical methods and tools. A first step in this direction that concentrates on re-using known results on the expected optimisation time has recently been presented by Doerr et al. [19]. We hope that theoreticians find this an interesting, challenging, and fruitful direction for future research. More importantly, we hope that the type of results obtained this way is more appealing and relevant for practitioners. Ultimately, we hope to bring theory and practice in the field closer together.

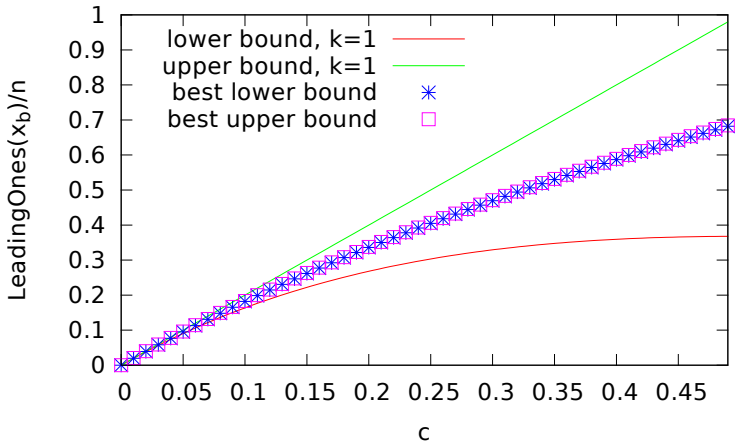

Figure 15: Bounds for $\mathrm{LO}(x_b)/n$ for $b = cn^2$ with $c \in \{.01, .02, \ldots, .49\}$.

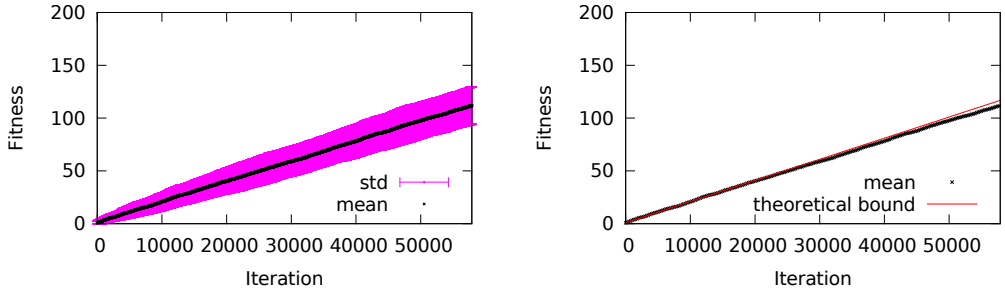

Figure 16: Empirical results for LO, $n = 1000$, and random initialisation: average values together with their standard deviations (left) and with the theoretical bounds (right).

References

[1] T. Jansen, C. Zarges, Fixed budget computations: A different perspective on run time analysis, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2012), ACM Press, 2012, pp. 1325–1332.

[2] A. Auger, B. Doerr (Eds.), Theory of Randomized Search Heuristics, World Scientific, 2011.

[3] Y. Yu, X. Yao, Z.-H. Zhou, On the approximation ability of evolutionary optimization with application to minimum set cover, Artificial Intelligence 180–181 (2012) 20–33.

[4] N. P. Troutman, B. E. Eskridge, D. F. Hougen, Is "best-so-far" a good algorithmic performance metric?, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008), ACM Press, 2008, pp. 1147–1148.

[5] E. Cantu-Paz, D. E. Goldberg, Are multiple runs of genetic algorithms better than one?, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003), Part 1, LNCS 2723, Springer, 2003, pp. 801–812.

[6] T. Jansen, C. Zarges, Analysis of evolutionary algorithms: From computational complexity analysis to algorithm engineering, in: 11th ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA 2011), ACM Press, 2011, pp. 1–14.

[7] B. Doerr, T. Jansen, C. Klein, Comparing global and local mutations on bit strings, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008), ACM Press, 2008, pp. 929–936.

[8] B. Doerr, F. Neumann, D. Sudholt, C. Witt, Runtime analysis of the 1-ant ant colony optimizer, Theoretical Computer Science 412 (17) (2011) 1629–1644.

[9] S. Droste, T. Jansen, I. Wegener, On the analysis of the (1+1) evolutionary algorithm, Theoretical Computer Science 276 (1–2) (2002) 51–81.

[10] D. Sudholt, C. Witt, Runtime analysis of a binary particle swarm optimizer, Theoretical Computer Science 411 (21) (2010) 2084–2100.

[11] R. J. Quick, V. J. Rayward-Smith, G. D. Smith, Fitness distance correlation and ridge functions, in: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN 1998), LNCS 1498, Springer, 1998, pp. 77–86.

[12] B. Doerr, M. Fouz, C. Witt, Sharp bounds by probability-generating functions and variable drift, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011), ACM Press, 2011, pp. 2083–2090.

[13] C. Witt, Optimizing linear functions with randomized search heuristics — The robustness of mutation, in: Proceedings of the 29th International

Symposium on Theoretical Aspects of Computer Science (STACS 2012), 2012, pp. 420–431.

[14] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1997.

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

[16] B. Doerr, Analyzing randomized search heuristics: Tools from probability theory, in: A. Auger, B. Doerr (Eds.), Theory of Randomized Search Heuristics, World Scientific, 2011, Ch. 1, pp. 1–20.

[17] M. Mitzenmacher, E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, 2005.

[18] S. Böttcher, B. Doerr, F. Neumann, Optimal fixed and adaptive mutation rates for the LeadingOnes problem, in: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN 2010), Part 1, LNCS 6238, Springer, 2010, pp. 1–10.

[19] B. Doerr, T. Jansen, C. Witt, C. Zarges, A method to derive fixed budget results from expected optimisation times, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2013), ACM Press, 2013, to appear.