Algorithmica
DOI 10.1007/s00453-016-0212-1

Towards a Runtime Comparison of Natural and Artificial Evolution

Tiago Paixão[1] · Jorge Pérez Heredia[2] · Dirk Sudholt[2] · Barbora Trubenová[1]

Received: 28 August 2015 / Accepted: 7 September 2016

Abstract Evolutionary algorithms (EAs) form a popular optimisation paradigm inspired by natural evolution. In recent years the field of evolutionary computation has developed a rigorous analytical theory to analyse the runtimes of EAs on many illustrative problems. Here we apply this theory to a simple model of natural evolution. In the Strong Selection Weak Mutation (SSWM) evolutionary regime the time between occurrences of new mutations is much longer than the time it takes for a mutated genotype to take over the population. In this situation, the population only contains copies of one genotype and evolution can be modelled as a stochastic process evolving one genotype by means of mutation and selection between the resident and the mutated genotype. The probability of accepting the mutated genotype then depends on the change in fitness. We study this process, SSWM, from an algorithmic perspective, quantifying its expected optimisation time for various parameters and investigating differences to a similar evolutionary algorithm, the well-known (1+1) EA. We show that SSWM can have a moderate advantage over the (1+1) EA at crossing fitness valleys and study an example where SSWM outperforms the (1+1) EA by taking advantage of information on the fitness gradient.

Keywords Runtime analysis · Evolutionary algorithms · Natural evolution · Population genetics · Theory · Strong selection weak mutation regime

An extended abstract of this article with preliminary results was presented at GECCO'15 [25].

✉ Dirk Sudholt
d.sudholt@sheffield.ac.uk

[1] IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria

[2] University of Sheffield, Sheffield S1 4DP, UK

Published online: 19 September 2016

1 Introduction

Evolutionary algorithms are popular general-purpose heuristics that have found countless applications in design and optimisation. For many large-scale problems they provide good results in reasonable time where many exact techniques fail [6]. They are a method of choice in black-box optimisation, where no information about the problem at hand is available, and the only way to obtain knowledge is to evaluate candidate solutions.

In the last 20 years evolutionary computation has developed a number of algorithmic techniques for the analysis of evolutionary and genetic algorithms. These methods typically focus on rigorously bounding the expected time required to reach a global optimum, or other well-specified high-fitness solutions. Problems studied include illustrative example functions from pseudo-Boolean optimisation that isolate characteristics found in more complex problems as well as problems from combinatorial optimisation. Various studies showed that the simple evolutionary algorithm called (1+1) EA evolving a single search point (see Sect. 2) finds global optima across a range of combinatorial optimisation problems in polynomial expected time, including shortest paths [29], sorting (as maximising sortedness) [29], minimum spanning trees [18], matroid optimisation problems [26], and Eulerian cycles [16]. The (1+1) EA (with restarts) further constitutes a polynomial-time randomised approximation scheme for the NP-hard Partition problem [35].

The runtime analysis of evolutionary algorithms has become one of the dominant concepts in evolutionary computation, leading to a plethora of results for evolutionary algorithms [1,10,20] as well as novel optimisation paradigms such as swarm intelligence [19,20,31] and artificial immune systems [11].

Interestingly, although evolutionary algorithms are heavily inspired by natural evolution, these methods have seldom been applied to natural evolution as studied in mathematical population genetics. This is a missed opportunity: the time it takes for a natural population to reach a fitness peak is an important question for the study of natural evolution. The kinds of results obtained from runtime analysis, namely how the runtime scales with genome size and mutation rate, are of general interest to population genetics. Moreover, recently there has been a renewed interest in applying computer science methods to problems in evolutionary biology with contributions from unlikely fields such as game theory [2], machine learning [33] and Markov chain theory [3]. Here, we present a first attempt at applying runtime analysis to the so-called Strong Selection Weak Mutation regime of natural populations.

The Strong Selection Weak Mutation model applies when mutations are rare enough and selection is strong enough that the time between occurrences of new mutations is long compared to the time a new genotype takes to replace the parent genotype [8]. Mutations occur rarely either because the mutation rate is low, or because the size of the underlying population is small. Upon occurrence, a new mutation represents relatively high fitness advantage or fitness loss, and strong selection ensures that it either promptly replaces the original genotype, or is entirely lost from the population (Fig. 1). Therefore, the population is most of the time composed of a single genotype and evolution occurs through "jumps" between different genotypes, corresponding to a new mutated genotype replacing the resident genotype in the population. The

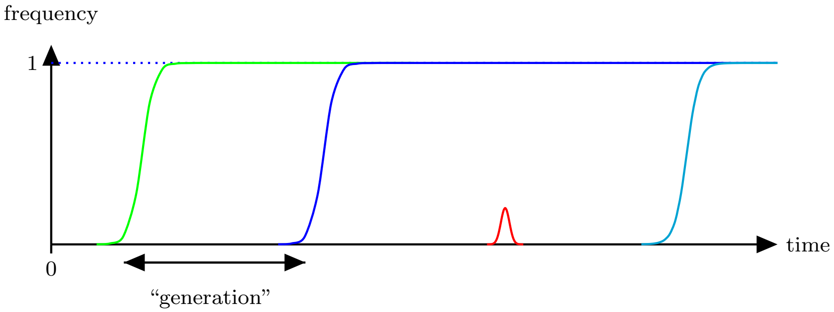

Fig. 1 Illustration of the strong selection weak mutation regime. The y-axis represents the frequency of genotypes that carry a specific mutation. Most of the time the population is composed of a single genotype, as new mutations (represented by *different colors*) are quickly either fixed (*green*, *blue*, *cyan*) or lost (*red*) (Color figure online)

relevant dynamics can then be characterized by a Markov stochastic process evolving one genotype. This model is obtained as a limit of many other models, such as the Wright–Fisher model [7]. One important aspect of this model is that new solutions are accepted with a probability $p_{\text{fix}} = \frac{1-e^{-2\beta\Delta f}}{1-e^{-2N\beta\Delta f}}$ that depends on the fitness difference Δf between the new mutated genotype and the resident genotype. Here N reflects the size of the underlying population (not to be confused with the population size in population-based EAs) and β represents the selection strength. One can think of f as defining a phenotype that is under selection to be maximized; β quantifies how strongly a unit change in f is favoured. This probability was first derived by Kimura [14] for a population of N individuals that are sampled binomially in proportion to their fitness.

This choice of acceptance function introduces two main differences to the (1+1) EA. First, while the (1+1) EA never decreases its current fitness (a property called *elitism*), SSWM accepts solutions of lower fitness (worsenings) with some positive probability. This is reminiscent of the Metropolis algorithm (Simulated Annealing with constant temperature) which can also accept worsenings (see e. g. [12]). Second, and in contrast to the Metropolis algorithm, solutions of higher fitness can be rejected, since they are accepted with a probability that is roughly proportional to the relative advantage they have over the current solution.

We cast this model of natural evolution as a stochastic process, referred to as SSWM, using common mutation operators from evolutionary algorithms. We then present first runtime analyses of this process using techniques from the analysis of randomised and evolutionary algorithms.

Our aims are manifold:

- to explore the performance of natural evolution in the context of runtime, comparing it against simple evolutionary algorithms like the (1+1) EA,
- to investigate the non-elitistic selection mechanism implicit to SSWM and its usefulness in the context of evolutionary algorithms, and
- to show that techniques for the analysis of evolutionary algorithms can be applied to simple models of natural evolution, aiming to open up a new research field at the intersection of evolutionary computation and population genetics (see [24] for recent work on unifying these two fields).

Our results are summarised as follows. For the simple function OneMax $(x) := \sum_{i=1}^{n} x_i$, an easy hill-climbing task, we show in Sect. 3 that with suitably large population sizes or selection strength, when $2(N-1)\beta \geq \ln(cn)$ with $c > 1.2$ and n being the problem size, SSWM is an effective hill climber as it optimises OneMax in expected time $O\left((n \log n)\left(1 + \frac{1}{2\beta}\right)\right)$. However, when $2N\beta$ is smaller than $\ln n$ by any constant factor, we encounter a phase transition and SSWM requires exponential time even on OneMax.

We then illustrate the particular features of the selection rule in more depth. In Sect. 4, we consider a function Cliff_d where a fitness valley of Hamming distance d needs to be crossed in order to reach the global optimum. For $d = \omega(\log n)$ the (1+1) EA needs time $\Theta(n^d)$, but SSWM is faster by a factor of $e^{\Omega(d)}$ because of its ability to accept worse solutions. Finally, in Sect. 5 we illustrate on the function Balance [27] that SSWM can drastically outperform the (1+1) EA because the fitness-dependent selection drives it to follow the steepest gradient. While the (1+1) EA needs exponential time in expectation with $\Omega(1)$ probability, SSWM with overwhelming probability finds an optimum in polynomial time.

The main technical difficulties are that in contrast to the simple (1+1) EA, SSWM is a non-elitist algorithm, hence fitness-level arguments based on elitism are not applicable. Popular techniques such as level-based theorems for non-elitist populations [4] are not applicable either because they require population sizes larger than 1. Moreover, while for the (1+1) EA transition probabilities to better solutions are solely determined by probabilities for flipping bits during mutation, for SSWM these additionally depend on the fitness difference. The analysis of SSWM is more challenging than the analysis of the (1+1) EA, and requires tailored proof techniques. We hope that these techniques will be helpful for analysing other evolutionary algorithms with fitness-based selection schemes.

2 Preliminaries

We define the optimisation time of SSWM as the number of the first iteration at which the optimum is accepted as a new individual.

As can be seen from the description above, the model resembles the (1+1) EA in that it only maintains one genotype that may be replaced by mutated versions of it. In fact, as will be obvious next, if fitness differences are large, SSWM behaves as the (1+1) EA. The candidate solutions are accepted with probability

$$p_{\text{fix}}(\Delta f) = \frac{1 - e^{-2\beta\Delta f}}{1 - e^{-2N\beta\Delta f}} \tag{1}$$

where $\Delta f \neq 0$ is the fitness difference to the current solution and $N \geq 1$ is the size of the underlying population. For $\Delta f = 0$ we define $p_{\text{fix}}(0) := \lim_{\Delta f \to 0} p_{\text{fix}}(\Delta f) = \frac{1}{N}$, so that p_{fix} is continuous and well defined for all Δf (Fig. 2). If $N = 1$, this probability is $p_{\text{fix}}(\Delta f) = 1$, meaning that any offspring will be accepted, and if $N \to \infty$, it will only accept solutions for which $\Delta f > 0$. The formula (1) was first derived by Kimura [14] and represents the *probability of fixation*, that is, the probability that a

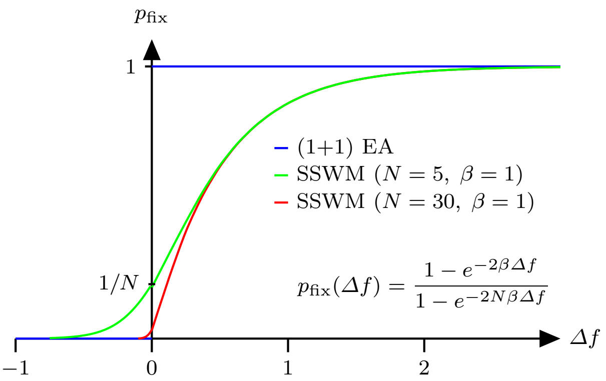

Fig. 2 Probability of fixation of a new mutation that is initially present in one copy in the population

gene initially present in one copy in a population of N individuals is eventually present in all individuals.

Since the acceptance function in this algorithm depends on the difference in fitness between genotypes, we include a parameter $\beta > 0$ that effectively scales the fitness function and that in population genetics models the strength of selection on a phenotype. By incorporating β as a parameter of this function (and hence of the algorithm) we avoid having to explicitly rescale the fitness functions we analyse, while allowing us to explore the performance of this algorithm on a family of functions.

For all $N > 1$ the function p_{fix} is strictly increasing (see Lemma 16 in the appendix) with a sigmoidal shape and limits $\lim_{\Delta f \to -\infty} p_{\text{fix}}(\Delta f) = 0$ as well as $\lim_{\Delta f \to \infty} p_{\text{fix}}(\Delta f) = 1$ (Fig. 2). Similar limits are obtained when β tends to ∞,

$$\lim_{\beta \to \infty} p_{\text{fix}}(\Delta f) = \begin{cases} 0 & \text{if } \Delta f < 0 \\ 1/N & \text{if } \Delta f = 0 \\ 1 & \text{if } \Delta f > 0 \end{cases}$$

As such, for large $|\beta \Delta f|$ this probability of acceptance is close to the one in the (1+1) EA, as long as $N > 1$, defeating the purpose of the comparison, with the only difference being the tie-breaking rule. While the (1+1) EA always accepts the new solution in case of a tie in fitness ($\Delta f = 0$), SSWM only accepts the new solution with probability $1/N$.

We can then cast the SSWM regime as Algorithm 1, where the function mutate(x) can be either standard bit mutation (all bits are mutated independently with probability $p_m = 1/n$, which we call *global mutations*) or flipping a single bit chosen uniformly at random (which we call *local mutations*). The SSWM model is a good approximation of natural evolution when the expected number of new mutants in the population is much less than one, which implies that local mutations are a better approximation for this regime. However, we also consider global mutations in order to facilitate a comparison with evolutionary algorithms such as the (1+1) EA (Algorithm 2), which uses global mutations. Most of our analyses apply to both mutation operators; the analysis in Sect. 4 applies to global mutations only.

Algorithm 1 SSWM

```
Choose $x \in \{0, 1\}^n$ uniformly at random
repeat
  $y \leftarrow \text{mutate}(x)$
  $\Delta f = f(y) - f(x)$
  Choose $r \in [0, 1]$ uniformly at random
  if $r < p_{\text{fix}}(\Delta f)$ then
    $x \leftarrow y$
  end if
until stop
```

Algorithm 2 (1+1) EA

```
Choose $x \in \{0, 1\}^n$ uniformly at random
repeat
  $y \leftarrow \text{mutate}(x)$


  if $f(y) \geq f(x)$ then
    $x \leftarrow y$
  end if
until stop
```

Next, we derive upper and lower bounds for $p_{\text{fix}}(\Delta f)$ that will be useful throughout the manuscript. The bounds for $\Delta f > 0$ show that p_{fix} is roughly proportional to the fitness difference between solutions $\beta \Delta f$.

Lemma 1 *For every $\beta \in \mathbb{R}^+$ and $N \in \mathbb{N}^+$ the following inequalities hold. If $\Delta f > 0$ then*

$$\frac{2\beta\Delta f}{1 + 2\beta\Delta f} \leq p_{\text{fix}}(\Delta f) \leq \frac{2\beta\Delta f}{1 - e^{-2N\beta\Delta f}}.$$

If $\Delta f < 0$ then

$$\frac{-2\beta\Delta f}{e^{-2N\beta\Delta f}} \leq p_{\text{fix}}(\Delta f) \leq \frac{e^{-2\beta\Delta f}}{e^{-2N\beta\Delta f} - 1}.$$

Proof In the following we frequently use $1 + x \leq e^x$ and $1 - e^{-x} \leq 1$ for all $x \in \mathbb{R}$ as well as $e^x \leq \frac{1}{1-x}$ for $x < 1$.

If $\Delta f > 0$,

$$p_{\text{fix}}(\Delta f) = \frac{1 - e^{-2\beta\Delta f}}{1 - e^{-2N\beta\Delta f}} \geq 1 - e^{-2\beta\Delta f} \geq 1 - \frac{1}{1 + 2\beta\Delta f} = \frac{2\beta\Delta f}{1 + 2\beta\Delta f}$$

as well as

$$p_{\text{fix}}(\Delta f) = \frac{1 - e^{-2\beta\Delta f}}{1 - e^{-2N\beta\Delta f}} \leq \frac{2\beta\Delta f}{1 - e^{-2N\beta\Delta f}}.$$

If $\Delta f < 0$, using the fact that $e^{-x} - 1 \leq e^{-x}$:

$$p_{\text{fix}}(\Delta f) = \frac{e^{-2\beta\Delta f} - 1}{e^{-2N\beta\Delta f} - 1} \leq \frac{e^{-2\beta\Delta f}}{e^{-2N\beta\Delta f} - 1}.$$

Similarly:

$$p_{\text{fix}}(\Delta f) = \frac{e^{-2\beta\Delta f} - 1}{e^{-2N\beta\Delta f} - 1} \geq \frac{e^{-2\beta\Delta f} - 1}{e^{-2N\beta\Delta f}} \geq \frac{-2\beta\Delta f}{e^{-2N\beta\Delta f}}.$$

□

The next lemma shows that the probability of accepting an improvement of Δf is exponentially larger (in $N\beta\Delta f$) than accepting its symmetric fitness variation $-\Delta f$.

Lemma 2 *For every $\beta \in \mathbb{R}^+$, $\Delta f \in \mathbb{R}$ and $N \in \mathbb{N}^+$*

$$\frac{p_{\text{fix}}(-\Delta f)}{p_{\text{fix}}(+\Delta f)} = e^{-2(N-1)\beta\Delta f}.$$

Proof

$$\begin{aligned}\frac{p_{\text{fix}}(-\Delta f)}{p_{\text{fix}}(+\Delta f)} &= \frac{e^{2\beta\Delta f}-1}{e^{2N\beta\Delta f}-1}\cdot\frac{1-e^{-2N\beta\Delta f}}{1-e^{-2\beta\Delta f}}\\ &= \frac{e^{2\beta\Delta f}}{e^{2N\beta\Delta f}} = e^{-2(N-1)\beta\Delta f}\end{aligned}$$

where we have applied the relation $\frac{e^x-1}{1-e^{-x}} = e^x$. □

3 SSWM on OneMax

The function OneMax$(x) := \sum_{i=1}^{n} x_i$ has been studied extensively in natural computation because of its simplicity. It represents an easy hill climbing task, and it is the easiest function with a unique optimum for all evolutionary algorithms that only use standard bit mutation for variation [30]. Showing that SSWM can optimise OneMax efficiently serves as proof of concept that SSWM is a reasonable optimiser. It further sheds light on how to set algorithmic parameters such as the selection strength β and the population size N. To this end, we first show a polynomial upper bound for the runtime of SSWM on OneMax for a selection strength of $2(N-1)\beta \geq \ln(cn)$. We then show that SSWM exhibits a phase transition on its runtime as a function of $2N\beta$; decreasing this parameter by a constant factor below $\ln n$ leads to exponential runtimes on OneMax.

Another reason why studying OneMax for SSWM makes sense is because not all evolutionary algorithms that use a fitness-dependent selection perform well on OneMax. Neumann et al. [17] as well as Oliveto and Witt [23] showed that evolutionary algorithms using fitness-proportional selection, including the Simple Genetic Algorithm, fail badly on OneMax even within exponential time, with very high probability.

3.1 Upper Bound for SSWM on OneMax

We first show the following simple lemma, which gives an upper bound on the probability of increasing or decreasing the number of ones in a search point by k in one mutation.

Lemma 3 *For any positive integer $k > 0$, let $\mathrm{mut}(i, i \pm k)$ for $0 \leq i \leq n$ be the probability that a global mutation of a search point with i ones creates an offspring* *with* $i \pm k$ *ones. Then*

$$\begin{aligned}\mathrm{mut}(i, i+k) &\le \left(\frac{n-i}{n}\right)^k \left(1-\frac{1}{n}\right)^{n-k} \cdot \frac{1.14}{k!}\\ \mathrm{mut}(i, i-k) &\le \left(\frac{i}{n}\right)^k \left(1-\frac{1}{n}\right)^{n-k} \cdot \frac{1.14}{k!}\\ \mathrm{mut}(i, i-k) &\le \frac{\mathrm{mut}(i, i-1)}{k!}.\end{aligned}$$

The proof can be found in the appendix; it uses arguments from the proof of Lemma 2 in [30]. The second inequality follows immediately from the first one due to the symmetry $\mathrm{mut}(i, i-k) = \mathrm{mut}(n-i, n-i+k)$.

Now we introduce the concept of drift and find some bounds for its forward and backward expression.

Definition 1 Let X_t be the number of ones in the current search point after mutation and selection have been applied. Then for all $1 \le i \le n$ the forward and backward drifts are

$$\begin{aligned}\Delta^+(i) =& E[X_{t+1} - i \mid X_t = i, X_{t+1} > i] \cdot P(X_{t+1} > i \mid X_t = i)\\ \Delta^-(i) =& E[X_{t+1} - i \mid X_t = i, X_{t+1} < i] \cdot P(X_{t+1} < i \mid X_t = i)\end{aligned}$$

and the net drift is the expected increase in the number of ones

$$\Delta(i) = \Delta^+(i) + \Delta^-(i).$$

Lemma 4 *Consider SSWM on* OneMax *and mutation probability* $p_m = \frac{1}{n}$. *Then for global mutations, the forward and backward drifts can be bounded by*

$$\begin{aligned}\Delta^+(i) \ge& \frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1} p_{\mathrm{fix}}(1)\\ |\Delta^-(i)| \le& 1.14 \left(1-\frac{1}{n}\right)^{n-1} \cdot \left(p_{\mathrm{fix}}(-1) + e \cdot p_{\mathrm{fix}}(-2)\right).\end{aligned}$$

For local mutations the relations are as follows

$$\begin{aligned}\Delta^+(i) =& \frac{n-i}{n} \cdot p_{\mathrm{fix}}(1)\\ |\Delta^-(i)| \le& p_{\mathrm{fix}}(-1).\end{aligned}$$

Proof For global mutations firstly we compute the lower bound for the forward drift. Expanding from Definition 1,

$$\begin{aligned}\Delta^+(i) =& \sum_{j=1}^{n-i} \text{mut}(i, i+j) \cdot j \cdot p_{\text{fix}}(j)\\ \geq& \,\text{mut}(i, i+1) \cdot p_{\text{fix}}(1)\\ \geq& \frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1} p_{\text{fix}}(1).\end{aligned}$$

Secondly we calculate the upper bound for the backward drift,

$$|\Delta^-(i)| = \sum_{j=1}^{i} \text{mut}(i, i-j) \cdot j \cdot p_{\text{fix}}(-j),$$

where j is now the number of new zeros. Bounding $\text{mut}(i, i-j)$ from above by Lemma 3 and bounding $i/n \leq 1 - 1/n$ yields

$$\begin{aligned}|\Delta^-(i)| \leq& \sum_{j=1}^{i} \frac{1.14}{j!} \cdot \left(1-\frac{1}{n}\right)^{j} \cdot \left(1-\frac{1}{n}\right)^{n-j} \cdot j \cdot p_{\text{fix}}(-j)\\ \leq& \sum_{j=1}^{i} \frac{1.14}{j!} \cdot \left(1-\frac{1}{n}\right)^{n-1} \cdot j \cdot p_{\text{fix}}(-j).\end{aligned}$$

Separating the case $j = 1$ and bounding the remaining fixation probabilities by $p_{\text{fix}}(-2)$

$$\begin{aligned}|\Delta^-(i)| \leq& 1.14\left(1-\frac{1}{n}\right)^{n-1} p_{\text{fix}}(-1) + 1.14\left(1-\frac{1}{n}\right)^{n-1} \cdot p_{\text{fix}}(-2) \cdot \sum_{j=2}^{i} \frac{1}{(j-1)!}\\ \leq& 1.14\left(1-\frac{1}{n}\right)^{n-1} \left(p_{\text{fix}}(-1) + p_{\text{fix}}(-2) \cdot e\right)\end{aligned}$$

where in the last step we have used $\sum_{j=2}^{i} \frac{1}{(j-1)!} \leq \sum_{j=1}^{\infty} \frac{1}{j!} = e - 1 \leq e$.

Finally, the case for local mutations is straightforward since the probability of a local mutation increasing the number of ones is $\frac{n-i}{n}$ and that of decreasing it is at most 1. □

The following theorem shows that SSWM is efficient on OneMax whenever $2(N-1)\beta \geq \ln(cn)$ for some constant $c > 1.2$, since then $p_{\text{fix}}(1)$ starts being greater than $n \cdot p_{\text{fix}}(-1)$, allowing for a positive drift even on the hardest fitness level ($n-1$ ones).

Theorem 5 *If* $2(N-1)\beta \geq \ln(cn)$, *for a constant* $c > 1.2$ *and* $\beta \in \mathbb{R}^+$ *then the expected optimisation time of SSWM on* OneMax *with local or global mutations is at most*

$$\frac{en\ln(n)+O(n)}{p_{\text{fix}}(1)} \leq \left(1+\frac{1}{2\beta}\right)\cdot(en\ln(n)+O(n))$$

for every initial search point.

Our preliminary work [25, Theorem 4] required $2N\beta \geq \ln(11n)$ and $\beta \leq 1$. The latter condition is counterintuitive as increasing β leads to more elitistic behaviour. The former condition is related to the one in Theorem 5 as

$$2(N-1)\beta \geq \ln(cn) \Leftrightarrow 2N\beta \geq \ln(e^{2\beta}cn). \tag{2}$$

The advantage of the condition $2(N-1)\beta \geq \ln(cn)$ is that the result now holds for all scaling factors $\beta \in \mathbb{R}^+$, thus removing the previous restriction on β.

The factor $\frac{1}{p_{\text{fix}}(1)} \leq 1+\frac{1}{2\beta}$ (by Lemma 2) on the runtime bound represents the extra time paid due to the probability of rejecting a better search point. For small selection strength $\beta \ll 1$ the upper bound essentially increases with $1/(2\beta)$. This makes sense as for small β (and $N\beta \gg 0$) we have $p_{\text{fix}}(1) \approx 2\beta$ (cf. Lemma 2). In this regime absolute fitness differences are small and improvements are only accepted with a small probability.

Proof of Theorem 5 We only give a proof for global mutations here. The analysis for local mutations follows the same way, with simpler calculations, and without the factor "*e*" in the running time bound.

We estimate an upper bound for $p_{\text{fix}}(-2)$

$$\begin{aligned}p_{\text{fix}}(-2) &= \frac{e^{4\beta}-1}{e^{4N\beta}-1} = \frac{e^{2\beta}-1}{e^{2N\beta}-1}\cdot\frac{e^{2\beta}+1}{e^{2N\beta}+1}\\ &= p_{\text{fix}}(-1)\cdot\frac{e^{2\beta}+1}{e^{2N\beta}+1} \leq p_{\text{fix}}(-1)\cdot\frac{2e^{2\beta}}{e^{2N\beta}}.\end{aligned}$$

Using $2(N-1)\beta \geq \ln(cn)$ leads to $p_{\text{fix}}(-2) \leq p_{\text{fix}}(-1)\cdot\frac{2}{cn}$.

Continuing from Lemma 4,

$$\begin{aligned}\Delta(i) &\geq \left(1-\frac{1}{n}\right)^{n-1}\cdot\left(\frac{n-i}{n}\cdot p_{\text{fix}}(1) - 1.14 p_{\text{fix}}(-1) - 1.14e\cdot p_{\text{fix}}(-2)\right)\\ &= \left(1-\frac{1}{n}\right)^{n-1} p_{\text{fix}}(1)\cdot\left(\frac{n-i}{n} - 1.14\frac{p_{\text{fix}}(-1)}{p_{\text{fix}}(1)} - 1.14e\frac{p_{\text{fix}}(-2)}{p_{\text{fix}}(1)}\right).\end{aligned}$$

Using $p_{\text{fix}}(-2) \leq p_{\text{fix}}(-1)\cdot\frac{2}{cn}$ and then Lemma 2 leads to

$$\begin{aligned}\Delta(i) &\geq \left(1-\frac{1}{n}\right)^{n-1} p_{\text{fix}}(1)\cdot\left(\frac{n-i}{n} - 1.14\frac{p_{\text{fix}}(-1)}{p_{\text{fix}}(1)} - \frac{2.28e}{cn}\frac{p_{\text{fix}}(-1)}{p_{\text{fix}}(1)}\right)\\ &= \left(1-\frac{1}{n}\right)^{n-1} p_{\text{fix}}(1)\cdot\left(\frac{n-i}{n} - \left(1.14+\frac{2.28e}{cn}\right)\cdot e^{-2(N-1)\beta}\right).\end{aligned}$$

Since $2(N-1)\beta \geq \ln(cn)$, $c > 1.2$ and assuming $n \geq 100$ (otherwise an $O(n)$ bound is trivial) we obtain

$$\Delta(i) \geq \left(1 - \frac{1}{n}\right)^{n-1} p_{\text{fix}}(1) \cdot \left(\frac{n-i}{n} - \frac{1.2}{cn}\right).$$

Using $c > 1.2$ we verify that $(n-i)/n - 1.2/(cn)$ is always positive for $i \geq 1$, hence we may bound $(1-1/n)^{n-1} \geq 1/e$, leading to

$$\begin{aligned}\Delta(i) &\geq p_{\text{fix}}(1) \cdot \left(\frac{n-i}{en} - \frac{1.2}{cen}\right) \\ &\geq p_{\text{fix}}(1) \cdot \frac{c(n-i) - 1.2}{cen}.\end{aligned}$$

Now we apply Johannsen's variable drift theorem [13] to the number of zeros, as this represents the distance to the optimum. In a nutshell, the theorem states that if the expected decrease of the distance to the optimum $\Delta(i)$ is bounded from below by a function $h(z)$ then the expected optimisation time, starting in state X_0, is bounded as

$$E(T \mid X_0) \leq \frac{z_{\min}}{h(z_{\min})} + \int_{z_{\min}}^{X_0} \frac{1}{h(z)} \, \mathrm{d}z$$

where z is the number of zeros $(n-i)$, T is the optimisation time and $z_{\min}$ is the smallest positive state. Here we have $z_{\min} = 1$, $X_0 \leq n$ and by using

$$\Delta(i) \geq \; h(z) = \; p_{\text{fix}}(1) \cdot \frac{cz - 1.2}{cen} \tag{3}$$

we obtain an upper bound for the runtime

$$\begin{aligned}E(T \mid X_0) &\leq \frac{1}{h(1)} + \int_1^n \frac{1}{h(z)} \, \mathrm{d}z \\ &\leq \frac{1}{p_{\text{fix}}(1)} \cdot \left(\frac{cen}{(c-1.2)} + \int_1^n \frac{cen}{(cz-1.2)} \mathrm{d}z\right) \\ &= \frac{1}{p_{\text{fix}}(1)} \cdot \left(O(n) + en \ln\left(\frac{cn-1.2}{c-1.2}\right)\right) \\ &\leq \frac{1}{p_{\text{fix}}(1)} \cdot \left(O(n) + en \ln\left(\frac{cn}{c-1.2}\right)\right) \\ &= \frac{1}{p_{\text{fix}}(1)} \cdot \left(O(n) + en \ln(n) + en \ln\left(\frac{c}{c-1.2}\right)\right) \\ &= \frac{en \ln(n) + O(n)}{p_{\text{fix}}(1)} \leq \left(1 + \frac{1}{2\beta}\right) \cdot (en \ln(n) + O(n))\end{aligned}$$

where in the last step we have used Lemma 2. □

3.2 A Critical Threshold for SSWM on OneMax

The upper bound from Theorem 5 required $2(N-1)\beta \geq \ln(cn)$, or equivalently, $2N\beta \geq \ln(n) + \ln(c) + 2\beta$. This condition is vital since if $N\beta$ is chosen too small, the runtime of SSWM on OneMax is exponential with very high probability, as we show next.

If $2N\beta$ is smaller than $\ln(n)$ by a factor of $1-\varepsilon$, for some constant $\varepsilon > 0$, the optimisation time is exponential in n, with overwhelming probability. SSWM therefore exhibits a phase transition behaviour: changing $N\beta$ by a constant factor makes a difference between polynomial and exponential expected optimisation times on OneMax.

Theorem 6 *If $2 \leq 2N\beta \leq (1-\varepsilon)\ln n$ for some $0 < \varepsilon < 1$, then the optimisation time of SSWM with local or global mutations on* OneMax *is at least $2^{cn^{\varepsilon/2}}$ with probability $1 - 2^{-\Omega(n^{\varepsilon/2})}$, for some constant $c > 0$.*

The condition $N\beta \geq 1$ is used to ease the presentation; it is not essential and we believe it can be dropped when using more detailed calculations. The idea behind the proof of Theorem 6 is to show that for all search points with at least $n - n^{\varepsilon/2}$ ones, there is a negative drift for the number of ones. This is because for small $N\beta$ the selection pressure is too weak, and worsenings in fitness are more likely than steps where mutation leads the algorithm closer to the optimum.

We then use the negative drift theorem with self-loops presented in Rowe and Sudholt [28] (an extension of the negative drift theorem [22] to stochastic processes with large self-loop probabilities). It is stated in the following for the sake of completeness. The theorem uses "$p_{k,k\pm d} \leq x$" as a shorthand for "$p_{k,k+d} \leq x$ and $p_{k,k-d} \leq x$".

Theorem 7 (Negative drift with self-loops [28]) *Consider a Markov process X_0, $X_1, \ldots$ on $\{0, \ldots, m\}$ with transition probabilities $p_{i,j}$ and suppose there exist integers a, b with $0 < a < b \leq m$ and $\varepsilon > 0$ such that for all $a \leq k \leq b$ the drift towards 0 is*

$$E(k - X_{t+1} \mid X_t = k) < -\varepsilon \cdot (1 - p_{k,k}) \tag{4}$$

where $p_{k,k}$ is the self-loop probability at state k. Further assume there exist constants $r, \delta > 0$ (i. e. they are independent of m) such that for all $k \geq 1$ and all $d \geq 1$

$$p_{k,k\pm d} \leq \frac{r(1-p_{k,k})}{(1+\delta)^d}. \tag{5}$$

Let T be the first hitting time of a state at most a, starting from $X_0 \geq b$. Let $\ell = b - a$. Then there is a constant $c > 0$ such that

$$\Pr\left(T \leq 2^{c\ell/r}\right) = 2^{-\Omega(\ell/r)}.$$

Proof of Theorem 6 We only give a proof for global mutations; the same analysis goes through for local mutations with similar, but simpler calculations.

The drift theorem, Theorem 7, will be applied to the number of zeros at the current point in time as distance function to the optimum, when the number of zeros is in the interval $[0, n^{\varepsilon/2}]$. By Chernoff bounds, SSWM starts with a fitness of at most $n - n^{\varepsilon/2}$ with probability $1 - 2^{-\Omega(n)}$. We assume in the following that this happens; the claimed probability bound then follows from a union bound of the failure probability $2^{-\Omega(n)}$ and a failure probability $2^{-\Omega(n^{\varepsilon/2})}$ which will result from an application of the drift theorem.

Let $p_{k,j}$ be the probability that SSWM will make a transition from a search point with k ones to one with j ones. Note that, although the drift theorem applies to the number of zeros, our notation of transition probabilities $p_{k,j}$ refers to numbers of ones for simplicity and consistency with other parts of the paper. Throughout the remainder of the proof we assume $k \geq n - n^{\varepsilon/2}$.

From Lemma 3 for every $1 \leq j \leq n - k$ we have

$$\begin{aligned} p_{k,k+j} \leq & \frac{1.14}{j!} \cdot \left(\frac{n-k}{n}\right)^j \cdot \left(1 - \frac{1}{n}\right)^{n-j} \cdot p_{\text{fix}}(j) \\ \leq & \frac{1.14}{j!} \cdot \left(\frac{n-k}{n}\right)^j \cdot p_{\text{fix}}(j) \\ \leq & 1.14 \cdot \left(n^{\varepsilon/2-1}\right)^j \cdot p_{\text{fix}}(j). \end{aligned} \tag{6}$$

By Lemma 2 we bound $p_{\text{fix}}(j)$ from above by $\frac{2\beta j}{1-e^{-2N\beta j}}$. This gives

$$p_{k,k+j} \leq \left(n^{\varepsilon/2-1}\right)^j \cdot \frac{3\beta j}{1 - e^{-2N\beta j}}.$$

The forward drift $\Delta^+(k) = \sum_{j=1}^{n-k} j \cdot p_{k,k+j}$ is then bounded as follows

$$\begin{aligned} \Delta^+(k) \leq & \sum_{j=1}^{n-k} j \cdot \left(n^{\varepsilon/2-1}\right)^j \cdot \frac{3\beta j}{1 - e^{-2N\beta j}} \\ \leq & \frac{3\beta}{1 - e^{-2N\beta}} \sum_{j=1}^{\infty} j^2 \cdot \left(n^{\varepsilon/2-1}\right)^j. \end{aligned}$$

Since $x = n^{\varepsilon/2-1}$ becomes arbitrarily small for large n, for positive $x \leq 0.09$ we have the inequality $\sum_{j=1}^{\infty} j^2 \cdot x^j = \frac{x(1+x)}{(1-x)^3} \leq x(1 + 5x)$, which together with $N\beta \geq 1$ implies

$$\Delta^+(k) \leq \frac{3\beta}{1 - e^{-2}} \cdot n^{\varepsilon/2-1} \cdot \left(1 + 5n^{\varepsilon/2-1}\right). \tag{7}$$

On the other hand,

$$\begin{aligned}p_{k,k-1} \geq& \frac{k}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \cdot p_{\text{fix}}(-1) \geq \frac{n - n^{\varepsilon/2}}{en} \cdot p_{\text{fix}}(-1)\\ =&\frac{p_{\text{fix}}(-1)}{e} \cdot \left(1 - n^{\varepsilon/2-1}\right) \geq \frac{1}{e} \cdot \frac{2\beta}{e^{2N\beta}} \cdot \left(1 - n^{\varepsilon/2-1}\right),\end{aligned}$$

and using $e^{2N\beta} \leq e^{(1-\varepsilon)\ln n} = n^{1-\varepsilon}$,

$$p_{k,k-1} \geq \frac{2\beta \cdot n^{\varepsilon}}{en} \cdot \left(1 - n^{\varepsilon/2-1}\right). \tag{8}$$

The expected increase in the number of ones at state k, denoted $\Delta(k)$, is hence at most

$$\Delta(k) \leq \sum_{j=1}^{n-k} j \cdot p_{k,k+j} - p_{k,k-1} = -\Omega(p_{k,k-1})$$

as $\sum_{j=1}^{n-k} j \cdot p_{k,k+j} = \Delta^{+}(k) = O(\beta n^{\varepsilon/2-1})$ by (7) and $p_{k,k-1} = \Omega(\beta n^{\varepsilon-1})$ by (8). To establish the first condition (4), we need to show that the drift is $\Delta(k) \leq -\Omega(1-p_{k,k})$. We already know that $\Delta(k) = -\Omega(p_{k,k-1})$, hence we need to show that $p_{k,k-1} = \Omega(1 - p_{k,k})$. By Lemma 3, we have for all k and all $j \in \mathbb{N}$,

$$p_{k,k-j} = \text{mut}(k, k-j) \cdot p_{\text{fix}}(-j) \leq \frac{\text{mut}(k, k-1)}{j!} \cdot p_{\text{fix}}(-1) = \frac{p_{k,k-1}}{j!}. \tag{9}$$

Using (9), we get

$$\begin{aligned}1 - p_{k,k} = \sum_{j=1}^{n-k} p_{k,k+j} + \sum_{j=1}^{k} p_{k,k-j} \leq& \sum_{j=1}^{n-k} p_{k,k+j} + p_{k,k-1} \sum_{j=1}^{\infty} \frac{1}{j!}\\ =& \sum_{j=1}^{n-k} p_{k,k+j} + (e-1)p_{k,k-1}\\ \leq& \sum_{j=1}^{n-k} j \cdot p_{k,k+j} + (e-1)p_{k,k-1} = O(p_{k,k-1})\end{aligned}$$

where we again used that $\sum_{j=1}^{n-k} j \cdot p_{k,k+j} = \Delta^{+}(k) = O(\beta n^{\varepsilon/2-1})$ by (7) and $p_{k,k} = \Omega(\beta n^{\varepsilon-1})$ by (8).

Hence $p_{k,k-1} = \Omega(1 - p_{k,k})$ and

$$\Delta(k) \leq -\Omega(p_{k,k-1}) = -\Omega(1 - p_{k,k})$$

which establishes the first condition of the drift theorem (4).

The second condition (5) on exponentially decreasing transition probabilities follows, for all $j \in \mathbb{N}$, from $p_{k,k-j} \le p_{k,k-1}/(j!) \le 2p_{k,k-1}/2^j \le 2(1-p_{k,k})/2^j$ by (9) and

$$p_{k,k+j} \le \left(n^{\varepsilon/2-1}\right)^j \cdot \frac{3\beta j}{1-e^{-2N\beta j}} \le \left(n^{\varepsilon/2-1}\right)^j \cdot \frac{3\beta j}{1-e^{-2}}$$

[multiplying by $p_{k,k-1}/p_{k,k-1}$ and using (8)]

$$\begin{aligned}
\le & p_{k,k-1} \cdot \frac{\left(n^{\varepsilon/2-1}\right)^j \cdot \frac{3\beta j}{1-e^{-2}}}{\frac{2\beta \cdot n^{\varepsilon}}{en} \cdot \left(1-n^{\varepsilon/2-1}\right)} \\
= & p_{k,k-1} \cdot n^{-\varepsilon/2} \cdot \left(n^{\varepsilon/2-1}\right)^{j-1} \cdot \frac{e}{1-n^{\varepsilon/2-1}} \cdot \frac{3}{2} \cdot \frac{j}{1-e^{-2}}
\end{aligned}$$

(if n is large enough)

$$\begin{aligned}
\le & p_{k,k-1} \cdot n^{-\varepsilon/2} \cdot \left(n^{\varepsilon/2-1}\right)^{j-1} \cdot 5j \\
\le & p_{k,k-1} \cdot 2^{-j} \le \ (1-p_{k,k}) \cdot 2^{-j}
\end{aligned}$$

where the penultimate inequality holds since for $j = 1$ we get $5n^{-\varepsilon/2} \le 1/2$ for large enough n and for $j > 1$ we additionally use $j \cdot \left(n^{\varepsilon/2-1}\right)^{j-1} \le 2^{-j+1}$. This proves the second condition (5) for $\delta := 1$ and $r := 2$. Applying the drift theorem completes the proof. □

We remark that for many evolutionary algorithms, such as the (1+1) EA and the (1+λ) EA, a lower bound on OneMax transfers to all functions with a unique global optimum. The reason is that in these cases OneMax is an easiest function amongst those with a single optimum [30]. This generalisation does not apply to SSWM. In fact, OneMax is probably not the easiest function with a single global optimum, even when the fitness range is normalised to $[0, n]$. The reason here is that acceptance is determined by absolute fitness differences, and a convex fitness curve (beautifully illustrated in [32, Fig. 1]) might amplify fitness differences close to the optimum, in order to compensate for small probabilities of mutation creating fitness improvements. We leave the discovery of an easiest function for SSWM (with a unique optimum) as an open topic for future work.

4 On Traversing Fitness Valleys

We have shown that with the right parameters, SSWM is an efficient hill climber. On the other hand, in contrast to the (1+1) EA, SSWM can accept worse solutions with a probability that depends on the magnitude of the fitness decrease. This is reminiscent of the Metropolis algorithm—although the latter accepts every improvement with probability 1, whereas SSWM may reject improvements.

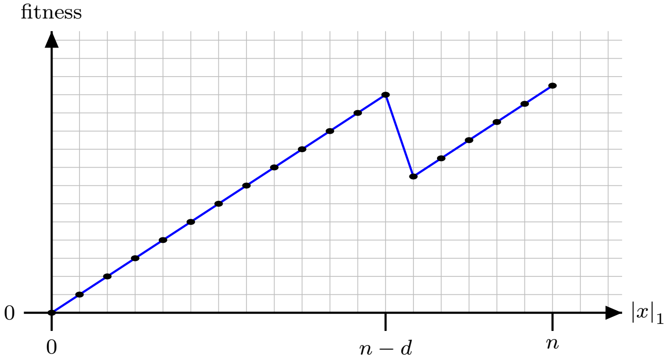

Fig. 3 Sketch of the function $\textsc{Cliff}_d$

Jansen and Wegener [12] compared the ability of the (1+1) EA and a Metropolis algorithm in crossing fitness valleys and found that both showed similar performance on *smooth integer* functions: functions where two Hamming neighbours have a fitness difference of at most 1 [12, Sect. 6].

We consider a similar function, generalising a construction by Jägersküpper and Storch [9]: the function $\textsc{Cliff}_d$ is defined such that non-elitist algorithms have a chance to jump down a "cliff" of height roughly d and to traverse a fitness valley of Hamming distance d to the optimum (see Fig. 3). Note that d may depend on n.

Definition 2 (Cliff)

$$\textsc{Cliff}_d(x) = \begin{cases} |x|_1 & \text{if } |x|_1 \leq n - d \\ |x|_1 - d + \frac{1}{2} & \text{otherwise} \end{cases}$$

where $|x|_1 = \sum_{i=1}^{n} x_i$ counts the number of ones.

The (1+1) EA typically optimises $\textsc{Cliff}_d$ through a direct jump from the top of the cliff to the optimum, which takes expected time $\Theta(n^d)$.

Theorem 8 *The expected optimisation time of the (1+1) EA on* $\textsc{Cliff}_d$*, for* $2 \leq d \leq n/2$*, is* $\Theta(n^d)$*.*

In order to prove Theorem 8, the following lemma will be useful for showing that the top of the cliff is reached with good probability. More generally, it shows that the conditional probability of increasing the number of ones in a search point to j, given it is increased to some value of j or higher, is at least $1/2$.

Lemma 9 *For all* $0 \leq i < j \leq n$*,*

$$\frac{\mathrm{mut}(i, j)}{\sum_{k=j}^{n} \mathrm{mut}(i, k)} \geq \frac{1}{2}.$$

The proof of this lemma is presented in the appendix.

Proof of Theorem 8 From any search point with $i < n - d$ ones, the probability of reaching a search point with higher fitness is at least $\frac{n-i}{en}$. The expected time for accepting a search point with at least $n - d$ ones is at most $\sum_{i=0}^{n-d-1} \frac{en}{n-i} = O(n \log n)$. Note that this is $O(n^d)$ since $d \geq 2$.

We claim that with probability $\Omega(1)$, the first such search point has $n - d$ ones: with probability at least $1/2$ the initial search point will have at most $n - d$ ones. Invoking Lemma 9 with $j := n - d$, with probability at least $1/2$ the top of the cliff is reached before any other search point with at least $n - d$ ones.

Once on the top of the cliff the algorithm has to jump directly to the optimum to overcome it. The probability of such a jump is $\frac{1}{n^d}\left(1 - \frac{1}{n}\right)^{n-d}$ and therefore the expected time to make this jump is $\Theta(n^d)$. □

SSWM with global mutations also has an opportunity to make a direct jump to the optimum. However, compared to the (1+1) EA its performance slightly improves when considering shorter jumps and accepting a search point of inferior fitness. The following theorem shows that for large enough cliffs, $d = \omega(\log n)$, the expected optimisation time is by a factor of $e^{\Omega(d)} = n^{\omega(1)}$ smaller than that of the (1+1) EA. Although both algorithms need a long time for large d, the speedup of SSWM is significant for large d.

Theorem 10 *The expected optimisation time of SSWM with global mutations and* $\beta = 1$, $N = \frac{1}{2}\ln(9n)$ *on* CLIFF_d *with* $d = \omega(\log n)$ *is at most* $n^d / e^{\Omega(d)}$.

Proof We define R as the expected time for reaching a search point with either $n - d$ or n ones, when starting with a worst possible non-optimal search point. Let T_{cliff} be the random optimisation time when starting with any search point of $n - d$ ones, hereinafter called the *top of the cliff* or a *local peak*. Then the expected optimisation time from any initial point is at most $R + \mathrm{E}(T_{\text{cliff}})$.

Let p_{success} be the probability that SSWM starting on top of the cliff will reach the optimum before reaching the top of the cliff again. We call the time period needed to reach the top of the cliff, or the global optimum, a *trial*. After the end of a trial, taking at most R expected generations, with probability $1 - p_{\text{success}}$ SSWM returns to the top of the cliff again, so

$$\mathrm{E}(T_{\text{cliff}}) \leq R + (1 - p_{\text{success}}) \cdot \mathrm{E}(T_{\text{cliff}}) \Leftrightarrow \mathrm{E}(T_{\text{cliff}}) \leq \frac{R}{p_{\text{success}}}. \qquad (10)$$

We first bound the worst-case time to return to the local peak or a global optimum as $R = O(n \log n)$. Let S_1 be the set of all search points with at most $n - d$ ones and $S_2 := \{0, 1\}^n \setminus S_1$. As long as the current search point remains within S_2, SSWM behaves like on OneMax. Here we have $2N\beta = \ln(9n)$ and by (2), along with $e^{2\beta} \cdot c = e^2 \cdot c < 9$ for a suitable constant $c > 1.2$ (e.g. $c := 1.21$), the condition $2(N - 1)\beta \geq \ln(cn)$ of Theorem 5 is satisfied. Repeating arguments from the proof of Theorem 5, in expected time $O\left(n \log n \cdot \left(1 + \frac{1}{2\beta}\right)\right) = O(n \log n)$ (as here $\beta = 1$)

SSWM either finds a global optimum or a search point in S_1. Likewise, as long as the current search point remains within S_1, SSWM essentially behaves like on OneMax and within expected time $O(n \log n)$ either the top of the cliff or a search point in S_2 is found.

SSWM can switch indefinitely between S_1 and S_2 within one trial, as long as no optimum nor a local peak is reached. Let X_t be the number of ones at time t, then the conditional probability of moving to a local peak—when from a search point with $i < n - d$ ones either a local peak or a non-optimal search point in S_2 is reached—is

$$\begin{aligned}
&\Pr(X_{t+1} = n-d \mid X_{t+1} \geq n-d, X_t = i < n-d)\\
&\quad= \frac{\Pr(X_{t+1} = n-d \mid X_t = i < n-d)}{\Pr(X_{t+1} \geq n-d \mid X_t = i < n-d)}\\
&\quad= \frac{\mathrm{mut}(i, n-d) \cdot p_{\mathrm{fix}}(n-d-i)}{\mathrm{mut}(i, n-d) \cdot p_{\mathrm{fix}}(n-d-i) + \sum_{k=n-d+1}^{n-1} \mathrm{mut}(i,k) \cdot p_{\mathrm{fix}}(k-i-d+1/2)}\\
&\quad= \frac{\mathrm{mut}(i, n-d)}{\mathrm{mut}(i, n-d) + \sum_{k=n-d+1}^{n-1} \mathrm{mut}(i,k) \cdot p_{\mathrm{fix}}(k-i-d+1/2)/p_{\mathrm{fix}}(n-d-i)}\\
&\quad\geq \frac{\mathrm{mut}(i, n-d)}{\sum_{k=n-d}^{n} \mathrm{mut}(i,k)}
\end{aligned}$$

as $p_{\mathrm{fix}}(n-d-i) \geq p_{\mathrm{fix}}(k-i-d+1/2)$ for all $n-d < k < n$. By Lemma 9, the above fraction is at least 1/2. Hence every time SSWM increases the number of ones to at least $n-d$, with probability at least 1/2 a local peak is found. This means that in expectation at most two transitions from S_1 to S_2 will occur before a local peak is reached, and the overall expected time spent in S_1 and S_2 is at most $R = O(1) \cdot O(n \log n)$.

The remainder of the proof now shows a lower bound on p_{success}, the probability of a trial being successful. A sufficient condition for a successful trial is that the following events occur: the next mutation creates a search point with $n-d+k$ ones, for some integer $1 \leq k \leq d$ chosen later, this point is accepted, and from there the global optimum is reached before returning to the top of the cliff.

We estimate the probabilities for these events separately in order to get an overall lower bound on the probability of a trial being successful.

From any local peak there are $\binom{d}{k}$ search points at Hamming distance k that have $n-d+k$ ones. Considering only such mutations, the probability of a mutation increasing the number of ones from $n-d$ by k is at least

$$\begin{aligned}
\mathrm{mut}(n-d, n-d+k) &\geq \frac{1}{n^k} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \cdot \binom{d}{k}\\
&\geq \frac{1}{en^k} \cdot \left(\frac{d}{k}\right)^k.
\end{aligned}$$

The probability of accepting such a move is

$$p_{\text{fix}}(k-d+1/2) = \frac{e^{2\beta(d-k-1/2)}-1}{e^{2N\beta(d-k-1/2)}-1} \geq \frac{e^{2(d-k-1/2)}-1}{(9n)^{(d-k-1/2)}}.$$

We now fix $k := \lfloor d/e \rfloor$ and estimate the probability of making and accepting a jump of length k:

$$\begin{aligned}
&\text{mut}(n-d, n-d+k) \cdot p_{\text{fix}}(k-d+1/2)\\
&\geq \frac{1}{en^k} \cdot \left(\frac{d}{k}\right)^k \cdot \frac{e^{2(d-k-1/2)}-1}{(9n)^{(d-k-1/2)}}\\
&= \Omega\left(n^{-d+1/2} \cdot \left(\frac{d}{k}\right)^k \cdot \left(\frac{e^2}{9}\right)^{d-k}\right)\\
&= \Omega\left(n^{-d+1/2} \cdot \left(e^{1/e} \cdot \left(\frac{e^2}{9}\right)^{1-1/e}\right)^d\right)\\
&= \Omega\left(n^{-d+1/2} \cdot \left(\frac{5}{4}\right)^d\right).
\end{aligned}$$

Finally, we show that, if SSWM does make this accepted jump, with high probability it climbs up to the global optimum before returning to a search point in S_1. To this end we work towards applying the negative drift theorem to the number of ones in the interval $[a := \lceil n-d+k/2 \rceil, b := n-d+k]$ and show that, since we start in state b, a state a or less is unlikely to be reached in polynomial time.

We first show that the drift is typically equal to that on OneMax. For every search point with more than a ones, in order to reach S_1, at least $k/2$ bits have to flip. Until this happens, SSWM behaves like on OneMax and hence reaches either a global optimum or a point in S_1 in expected time $O(n \log n)$. The probability for a mutation flipping at least $k/2$ bits is at most $1/(k/2)! = (\log n)^{-\Omega(\log n)} = n^{-\Omega(\log \log n)}$, so the probability that this happens in expected time $O(n \log n)$ is still $n^{-\Omega(\log \log n)}$.

Assuming such jumps do not occur, we can then use drift bounds from the analysis of OneMax for states with at least a ones. From the proof of Theorem 5 and (3) we know that the drift at i ones for $\beta = 1$ is at least

$$\Delta(i) \geq \Omega\left(\frac{n-i}{n}\right).$$

As in the proof of Theorem 6, we denote by $p_{i,j}$ the transition probability from a state with i ones to one with j ones. The probability of decreasing the current state is at most $p_{\text{fix}}(-1) = O(1/n)$ due to Lemma 2. The probability of increasing the current state is at most $(n-i)/n$ as a necessary condition is that one out of $n-i$ zeros needs to flip. Hence for $i \leq b$, which implies $n-i = \omega(1)$, the self-loop probability is at least

$$p_{i,i} \geq 1 - O\left(\frac{1}{n}\right) - \frac{n-i}{n} = 1 - O\left(\frac{n-i}{n}\right).$$

Together, we get $\Delta(i) \geq \Omega(1 - p_{i,i})$, establishing the first condition (4) of Theorem 7.

Note that $p_{\text{fix}}(1) = \frac{1-e^{-2}}{1-1/n} = \Omega(1)$, hence

$$1 - p_{i,i} \geq p_{i,i+1} \geq \frac{n-i}{en} \cdot p_{\text{fix}}(1) = \Omega\left(\frac{n-i}{n}\right). \tag{11}$$

The second condition (5) follows for improving jumps from i to $i + j$, $j \geq 1$, from Lemma 3 and (11):

$$p_{i,i+j} \leq \left(\frac{n-i}{n}\right)^j \cdot \frac{1}{j!} \cdot p_{\text{fix}}(j) \leq \frac{n-i}{n} \cdot \frac{1}{j!} \leq (1 - p_{i,i}) \cdot \frac{O(1)}{2^j}.$$

For backward jumps we get, for $1 \leq j \leq k/2$, and n large enough,

$$p_{i,i-j} \leq p_{\text{fix}}(-j) \leq \frac{e^{2j}}{e^{2Nj} - 1} = \frac{e^{2j}}{(9n)^j - 1} \leq 2^{-j}.$$

Now Theorem 7 can be applied with $r = O(1)$ and $\delta = 1$ and it yields that the probability of reaching a state of a or less in $n^{\omega(1)}$ steps is $n^{-\omega(1)}$.

This implies that following a length-k jump, a trial is successful with probability $1 - n^{-\omega(1)}$. This establishes $p_{\text{success}} := \Omega\left(n^{-d+1/2} \cdot \left(\frac{5}{4}\right)^d\right)$. Plugging this into (10), adding time R for the time to reach the top of the cliff initially, and using that $O(n^{1/2} \log n) \cdot (4/5)^d = e^{-\Omega(d)}$ for $d = \omega(\log n)$ yields the claimed bound. □

5 SSWM Outperforms (1+1) EA on Balance

Finally, we investigate a feature that distinguishes SSWM from the (1+1) EA as well as the Metropolis algorithm: the fact that larger improvements are more likely to be accepted than smaller improvements.

To this end, we consider the function BALANCE, originally introduced by Rohlfshagen et al. [27] as an example where rapid dynamic changes in dynamic optimisation can be beneficial. The function has also been studied in the context of stochastic ageing by Oliveto and Sudholt [21] and it goes back to an earlier idea by Witt [36].

In its static (non-dynamic) form, BALANCE can be illustrated by a two-dimensional plane, whose coordinates are determined by the number of leading ones (LO) in the first half of the bit string, and the number of ones in the second half, respectively. The former has a steeper gradient than the latter, as the leading ones part is weighted by a factor of n in the fitness (see Fig. 4).

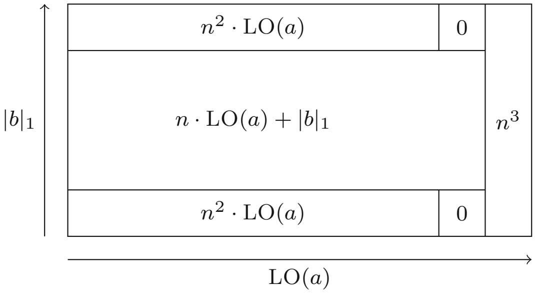
$n^2 \cdot \mathrm{LO}(a)$
0
$|b|_1$
$n \cdot \mathrm{LO}(a) + |b|_1$
n^3
$n^2 \cdot \mathrm{LO}(a)$
0
$\mathrm{LO}(a)$

Fig. 4 Visualisation of BALANCE [27]

Definition 3 (BALANCE [27]) Let $a, b \in \{0,1\}^{n/2}$ and $x = ab \in \{0,1\}^n$. Then

$$\textsc{Balance}(x) = \begin{cases} n^3 & \text{if } \mathrm{LO}(a) = n/2, \text{ else} \\ |b|_1 + n \cdot \mathrm{LO}(a) & \text{if } n/16 < |b|_1 < 7n/16, \text{ else} \\ n^2 \cdot \mathrm{LO}(a) & \text{if } |a|_0 > \sqrt{n}, \text{ else} \\ 0 & \text{otherwise} \end{cases}$$

where $|x|_1 = \sum_{i=1}^{n/2} x_i$, $|x|_0$ is a number of zeros and $\mathrm{LO}(x) := \sum_{i=1}^{n/2} \prod_{j=1}^{i} x_j$ counts the number of leading ones.

The function is constructed in such a way that all points with a maximum number of leading ones are global optima, whereas increasing the number of ones in the second half beyond a threshold of $7n/16$ (or decreasing it below a symmetric threshold of $n/16$) leads to a trap, a region of local optima that is hard to escape from.

Rohlfshagen et al. [27, Theorem 3] showed the following lower bound for the (1+1) EA. The statement is specialised to non-dynamic optimisation and slightly strengthened by using a statement from their proof.

Theorem 11 ([27]) *With probability $\Omega(1)$ the (1+1) EA on* BALANCE *reaches a trap, and then needs at least $n^{\sqrt{n}}$ further generations in expectation to find an optimum from there. The expected optimisation time of the (1+1) EA is thus $\Omega(n^{\sqrt{n}})$.*

We believe that the probability bound $\Omega(1)$ can be strengthened to $1 - e^{-\Omega(n^{1/2})}$ with a more detailed analysis, which would show that the (1+1) EA gets trapped with an overwhelming probability.

We next show that SSWM with high probability finds an optimum in polynomial time. For appropriately small β we have sufficiently many successes on the LO-part such that we find an optimum before the ONEMAX-part reaches the region of local optima. This is because for small β the probability of accepting small improvements is small. The fact that SSWM for $\beta < 1$ is slower than the (1+1) EA on ONEMAX by a factor of $O(1/\beta)$ turns into an advantage over the (1+1) EA on BALANCE.

The following lemma shows that SSWM effectively uses elitist selection on the LO-part of the function in a sense that every decrease is rejected with overwhelming probability.

Lemma 12 *For every $x = ab$ with $n/16 < |b|_1 < 7n/16$ and $\beta = n^{-3/2}$ and $N\beta = \ln n$, the probability of SSWM with local or global mutations accepting a mutant $x' = a'b'$ with $\mathrm{LO}(a') < \mathrm{LO}(a)$ and $n/16 < |b'|_1 < 7n/16$ is $O(n^{-n})$.*

Proof The loss in fitness is at least $n - (|b'|_1 - |b|_1) \geq n/2$. The probability of SSWM accepting such a loss is at most

$$p_{\text{fix}}(-n/2) \leq \frac{1 - e^{-2\beta(-n/2)}}{1 - e^{-2N\beta(-n/2)}} \leq \frac{e^{\beta n}}{e^{N\beta n} - 1}.$$

Assuming $\beta = n^{-3/2}$ and $N\beta = \ln n$, this is at most

$$\frac{e^{\frac{\sqrt{n}}{n}}}{n^n - 1} \leq \frac{e}{n^n - 1} = O(n^{-n}).$$

□

The following lemma establishes the optimisation time of the SSWM algorithm on either the OneMax or the LO-part of Balance.

For global mutations we restrict our considerations to *relevant steps*, defined as steps where no leading ones in the first half of the bit string is flipped. The probability of a relevant step is always at least $(1-1/n)^{n/2} \approx e^{-1/2}$. When using local mutations, all steps are defined as relevant.

Lemma 13 *Let $\beta = n^{-3/2}$ and $N\beta = \ln n$. With probability $1 - e^{-\Omega(n^{1/2})}$, SSWM with either local or global mutations either optimises the LO part or reaches the trap (a set of all search points with fitness $n^2 \cdot \mathrm{LO}(a)$) within*

$$T := \frac{n^2}{4} \cdot \frac{1}{p_{\text{fix}}(n - \sqrt{n})} \cdot \left(1 + n^{-1/4}\right)$$

relevant steps.

Proof We use the method of typical runs [34]: we consider the typical behaviour of the algorithm, and show that events where the algorithm deviates from a typical run are very unlikely. A union bound over all such failure events proves the claimed probability bound.

Consider a relevant step, implying that global mutations will leave all leading ones intact. With probability $1/n$ a local or global mutation will flip the first 0-bit. This increases the fitness by $k \cdot n - \Delta_{\text{OM}}$, where Δ_{OM} is the difference in the OneMax-value of b caused by this mutation and k is the number of consecutive 1-bits following the first 0-bit, after mutation. The latter bits are called *free riders* and it is well known (see [15, Lemma 1 and proof of Theorem 2]) that the number of free riders follows a geometric distribution with parameter 1/2, only capped by the number of bits to the end of the bit string a.

The probability of flipping at least $\sqrt{n}$ bits in one global mutation is at most $1/(\sqrt{n})! = e^{-\Omega(\sqrt{n})}$ and the probability that this happens at least once in T relevant steps is still of the same order (using that $T = \text{poly}(n)$ as $p_{\text{fix}}(n - \sqrt{n}) \geq$

$1/N \geq 1/\text{poly}(n)$). We assume in the following that this does not happen, which allows us to assume $\Delta_{\text{OM}} \leq \sqrt{n}$. We also assume that the number of leading ones is never decreased during non-relevant steps as the probability of accepting such a fitness decrease is $O(n^{-n})$ by Lemma 12 and the expected number of non-relevant steps before T relevant steps have occurred is $O(T)$.

We have now restricted our attention to runs in which the number of leading ones can never decrease, and any increase by mutation is accepted with probability at least $p_{\text{fix}}(n - \sqrt{n})$. In a relevant step, the probability of increasing the number of leading ones is hence at least $1/n \cdot p_{\text{fix}}(n - \sqrt{n})$ and the expected number of such improvements in

$$T := \frac{n^2}{4} \cdot \frac{1}{p_{\text{fix}}(n - \sqrt{n})} \cdot (1 + n^{-1/4})$$

relevant steps is at least $n/4 + n^{3/4}/4$.

Now, a LO-value of $n/2$ is reached if (event A) in T relevant steps at least $n/4 + n^{3/4}/8$ improvements happen and if (event B) the first $n/4 + n^{3/4}/8$ improvements lead to a total of at least $n/4 - n^{3/4}/8$ free riders (unless the number of leading ones hits $n/2$). Note that these two events are independent, as improvements are due to the current mutation and the number of free riders is due to the uniform random distribution of bits following the first 0-bit [15, Lemma 1].

By Chernoff bounds [5] , the probability that the typical event A does not occur, that is, less than $n/4+n^{3/4}/8$ improvements happen, is $e^{-\Omega(n^{1/2})}$. For the same reason also the probability of event B not occurring is $e^{-\Omega(n^{1/2})}$. Taking the union bound over all rare failure probabilities proves the claim. □

We now show that the OneMax part is not optimised before the LO part.

Lemma 14 *Let $\beta = n^{-3/2}$, $N\beta = \ln n$, and T be as in Lemma 13. The probability that SSWM starting with a_0b_0 such that $n/4 \leq |b_0|_1 \leq n/4 + n^{3/4}$ creates a search point ab with $|b|_1 \leq n/16$ or $|b|_1 \geq 7n/16$ in T relevant steps is $e^{-\Omega(n^{1/2})}$.*

It will become obvious that in T relevant steps SSWM typically makes a progress of $O(n)$ on the OneMax part. The proof of Lemma 14 requires a careful and delicate analysis to show that the constant factors are small enough such that the stated thresholds for $|b|_1$ are not surpassed.

Proof of Lemma 14 We only prove that a search point with $|b|_1 \geq 7n/16$ is unlikely to be reached with the claimed probability. The probability for reaching a search point with $|b|_1 \leq n/16$ is clearly no larger, and a union bound for these two events leads to a factor of 2 absorbed in the asymptotic notation.

The proof is divided into two parts: we first estimate the increase of $|b|_1$ in steps where the number of leading ones in a does not change. We refer to these as *regular steps*. Steps where mutation increases the number of leading ones in a are called *special steps*; during these steps every mutation of b is accepted as the fitness gain through additional leading ones guarantees that any change in b will be accepted. In the following, we first show that the progress in $|b|_1$ in regular steps is close to $1.14n/9$

and then we show that the progress in special steps is bounded by $O(n^{3/4})$ with high probability.

Bounding the progress in regular steps: note that for $\beta = n^{-3/2}$ we have

$$p_{\text{fix}}(n - \sqrt{n}) \geq \frac{2\beta(n - \sqrt{n})}{1 + 2\beta(n - \sqrt{n})} \geq 2\beta n \cdot (1 - O(n^{-1/2})).$$

Hence

$$T \leq \frac{n^2}{4} \cdot \frac{1}{2\beta n} \cdot \left(1 + O(n^{-1/2})\right) = \frac{n}{8\beta} \cdot \left(1 + O(n^{-1/2})\right).$$

We call a relevant step *improving* if the number of ones in b increases and the step is accepted.

We first consider only steps where the number of leading ones stays the same. Then the probability that the OneMax value increases from k by j, adapting Lemma 3 to a string of length $n/2$, is at most

$$p_j \leq \left(\frac{n/2 - k}{n}\right)^j \cdot \frac{1.14}{j!} \cdot p_{\text{fix}}(j)$$

(using $n/2 - k \leq n/4$)

$$\begin{aligned}
&\leq \frac{1.14 \cdot 4^{-j}}{j!} \cdot p_{\text{fix}}(j) \leq \frac{1.14 \cdot 4^{-j}}{j!} \cdot \frac{2\beta j}{1 - e^{-2N\beta j}} \\
&\leq 2.28\beta \cdot 4^{-j} \cdot \frac{1}{1 - e^{-2N\beta j}} =: p_j.
\end{aligned}$$

In the following, we work with pessimistic transition probabilities p_j. Note that for all $j \geq 1$

$$\frac{p_j}{p_1} = 4^{-(j-1)} \cdot \frac{1 - e^{-2N\beta}}{1 - e^{-2N\beta j}} \leq 4^{-(j-1)}.$$

Let p^+ denote (a lower bound on) the probability of an improving step, then

$$p^+ \leq \sum_{j=1}^{\infty} p_j \leq p_1 \cdot \sum_{j=1}^{\infty} 4^{-(j-1)} = p_1 \cdot \frac{4}{3}.$$

The conditional probability of advancing by j, given an improving step, is then

$$\frac{p_j}{p^+} \leq 4^{-(j-1)} \cdot \frac{p_1}{p^+} = \left(1 - \frac{3}{4}\right)^{j-1} \cdot \frac{3}{4},$$

which corresponds to a geometric distribution with parameter $3/4$.

Now, by Chernoff bounds, the probability of having more than $S := (1 + n^{-1/4}) \cdot p^+ \cdot T$ improving steps in T relevant steps is $e^{-\Omega(n^{1/2})}$. Using a Chernoff bound for geometric random variables [5, Theorem 1.14], the probability of S improving steps yielding a total progress of at least $(1 + n^{-1/4}) \cdot 4/3 \cdot S$ is $e^{-\Omega(n^{1/2})}$.

If none of these rare events happen, the progress is at most

$$\begin{aligned}
&(1 + O(n^{-1/4})) \cdot \frac{4}{3} \cdot p^+ \cdot T \\
&\quad = (1 + O(n^{-1/4})) \cdot \frac{16}{9} \cdot p_1 \cdot T \\
&\quad = (1 + O(n^{-1/4})) \cdot \frac{16}{9} \cdot \frac{2.28\beta}{4} \cdot \frac{1}{1 - e^{-2N\beta}} \cdot \frac{n}{8\beta} \cdot \left(1 + O(n^{-1/2})\right) \\
&\quad = (1 + O(n^{-1/4})) \cdot \frac{1.14}{9} \cdot \frac{1}{1 - n^{-2}} \cdot n \cdot \left(1 + O(n^{-1/2})\right) \\
&\quad \leq (1 + O(n^{-1/4})) \cdot \frac{1.14}{9} \cdot n.
\end{aligned}$$

Bounding the progress in special steps: we also have at most $n/2$ steps where the number of leading ones increases. If the number of leading ones increases by $\delta \geq 1$, the fitness increase is $\delta n + |b'|_1 - |b|_1$. Hence the above estimations of jump lengths are not applicable. These special steps are unorthodox as the large fitness increase makes it likely that any mutation on the OneMax part is accepted. We show that the progress on the OneMax part across all special steps is $O(n^{3/4})$ with high probability.

We grant the algorithm an advantage if we assume that, after initialising with $|b|_1 \geq n/4$, no search point with $|b|_1 < n/4$ is ever reached.[1] Under this assumption we always have at least as many 1-bits as 0-bits in b, and mutation in expectation flips at least as many 1-bits to 0 as 0-bits to 1.

Then the progress in $|b|_1$ in one special step increasing the number of leading ones by $d \geq 1$ can be described as follows. Imagine a matching (pairing) between all bits in b such that each pair contains at least one 1-bit. Let X_i denote the random change in $|b|_1$ by the i-th pair. If the pair has two 1-bits, $X_i \leq 0$ with probability 1. Otherwise, we have $X_i = 1$ if the 0-bit in the pair is flipped, the 1-bit in the pair is not flipped, and the mutant is accepted (which depends on the overall $|b|_1$-value in the mutant). The potential fitness increase is at most $dn + n/2$ as the range of $|b|_1$-values is $n/2$. Likewise, we have $X_i = -1$ if the 0-bit is not flipped, the 1-bit is flipped, and the mutant is accepted (which again depends on the overall $|b|_1$-value in the mutant). The fitness increase is at least $dn - n/2$. With the remaining probability we have $X_i = 0$. Hence for global mutations (for local mutations simply drop the $1 - 1/n$ term) the total progress in a special step increasing $\mathrm{LO}(a)$ by d is stochastically dominated by a sum of independent variables $Y_1, \ldots, Y_{n/4}$ where $\Pr(Y_i = \pm 1) = 1/n \cdot (1 - 1/n) \cdot p_{\text{fix}}(dn \pm n/2)$ and $Y_i = 0$ with the remaining probability.

[1] Otherwise, we restart our considerations from the first point in time where $|b|_1 \geq n/4$ again, replacing T with the number of remaining steps. With overwhelming probability we will then again have $|b|_1 \leq n/4 + n^{3/4}$.

There is a bias towards increasing the number of ones due to differences in the arguments of p_{fix}: $\mathrm{E}(Y_i) = 1/n \cdot (1 - 1/n) \cdot (p_{\text{fix}}(dn + n/2) - p_{\text{fix}}(dn - n/2))$. Using the definition of p_{fix} and preconditions $\beta = n^{-3/2}$, $N\beta = \ln n$, the bracket is bounded as

$$\begin{aligned}
&p_{\text{fix}}(dn + n/2) - p_{\text{fix}}(dn - n/2)\\
&\quad = \frac{1 - e^{-2dn^{-1/2} - n^{-1/2}}}{1 - n^{-2dn+n}} - \frac{1 - e^{-2dn^{-1/2} + n^{-1/2}}}{1 - n^{-2dn-n}}\\
&\quad = (1 + o(1)) \left(\left(1 - e^{-2dn^{-1/2} - n^{-1/2}}\right) - \left(1 - e^{-2dn^{-1/2} + n^{-1/2}}\right)\right)\\
&\quad = (1 + o(1)) \cdot e^{-2dn^{-1/2}} \left(e^{n^{-1/2}} - e^{-n^{-1/2}}\right)\\
&\quad \le (1 + o(1)) \cdot e^{-2dn^{-1/2}} \left((1 + 2n^{-1/2}) - (1 - n^{-1/2})\right)\\
&\quad = (1 + o(1)) \cdot e^{-2dn^{-1/2}} \cdot 3n^{-1/2}
\end{aligned}$$

where in the last inequality we have used $1 + x \le e^x$ for all x and $e^x \le 1 + 2x$ for $0 \le x \le 1$.

Note that the expectation, and hence the bias, is largest for $d = 1$, in which case we get, using $e^{-2dn^{-1/2}} \le e^{-2n^{-1/2}} \le 1$,

$$\mathrm{E}(Y_i) \le (1 + o(1)) \cdot 1/n \cdot (1 - 1/n) \cdot 3n^{-1/2} \le 4n^{-3/2}$$

for n large enough.

The total progress in all m special steps is hence stochastically dominated by a sequence of $m \cdot n/4$ random variables Y_i as defined above, with $d := 1$. Invoking Lemma 17 (basically a Hoeffding bound on the non-zero outcomes of the variables), stated in the appendix, with $\delta := n^{3/4}$, the total progress in all special steps is at most $\delta + m \cdot n/4 \cdot \mathrm{E}(Y_i) = \delta + O(n^{1/2}) = O(n^{3/4})$ with probability $1 - e^{-\Omega(n^{1/2})}$.

Hence the net gain in the number of ones in all special steps is at most $n^{3/4} + O(mn/4 \cdot n^{-3/2}) = O(n^{3/4})$ with probability $1 - e^{-\Omega(n^{1/2})}$.

Together with all regular steps, the progress on the OneMax part is at most $1.14n/9 + O(n^{3/4})$, which for large enough n is less than the distance $7n/16 - (n/4 + n^{3/4})$ to reach a point with $|b|_1 \ge 7n/16$ from initialisation. This proves the claim. □

Finally, we put the previous lemmas together into our main theorem that establishes that SSWM can optimise Balance in polynomial time.

Theorem 15 *With probability* $1 - e^{-\Omega(n^{1/2})}$ *SSWM with* $\beta = n^{-3/2}$ *and* $N\beta = \ln n$ *optimises* Balance *in time* $O(n/\beta) = O(n^{5/2})$.

Proof By Chernoff bounds, the probability that for the initial solution $x_0 = a_0 b_0$ we have $n/4 - n^{3/4} \le |b_0|_1 \le n/4 + n^{3/4}$ is $1 - e^{-\Omega(n^{1/2})}$. We assume pessimistically that $n/4 \le |b_0|_1 \le n/4 + n^{3/4}$. Then Lemma 14 is in force, and with probability $1 - e^{-\Omega(n^{1/2})}$ within T relevant steps, T as defined in Lemma 13, SSWM does not reach a trap or a search point with fitness 0. Lemma 13 then implies that with probability $1 - e^{-\Omega(n^{1/2})}$ an optimal solution with $n/2$ leading ones is found.

The time bound follows from the fact that $T = O(n/\beta)$ and that, again by Chernoff bounds, we have at least T relevant steps in $3T$ iterations of SSWM, with probability $1 - e^{-\Omega(n^{1/2})}$. □

6 Conclusions

The field of evolutionary computation has matured to the point where techniques can be applied to models of natural evolution. Our analyses have demonstrated that runtime analysis of evolutionary algorithms can be used to analyse a simple model of natural evolution, opening new opportunities for interdisciplinary research with population geneticists and biologists.

Our conclusions are highly relevant for biology, and open the door to the analysis of more complex fitness landscapes in this field and to quantifying the efficiency of evolutionary processes in more realistic scenarios of evolution. One interesting aspect of our results is that they impose conditions on population size (N) and strength of selection (β) which represent fundamental limits to what is possible by natural selection. We hope that these results may inspire further research on the similarities and differences between natural and artificial evolution.

From a computational perspective, we have shown that SSWM can overcome obstacles such as posed by Cliff_d and Balance in different ways to the (1+1) EA, due to its non-elitistic selection mechanism. We have seen how the probability of accepting a mutant can be tuned to enable hill climbing, where fitness-proportional selection fails, as well as tunnelling through fitness valleys, where elitist selection fails. For Balance we showed that SSWM can take advantage of information about the steepest gradient. The selection rule in SSWM hence seems to be a versatile and useful mechanism. Future work could investigate its usefulness in the context of population-based evolutionary algorithms.

Acknowledgments The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 618091 (SAGE). The authors thank the anonymous reviewers for their many constructive comments.

Appendix

This appendix contains proofs that were omitted from the main part.

Lemma 16 *p_{fix} is monotonic for all $N \geq 1$ and strictly increasing for $N > 1$*

Proof If $N = 1$, $p_{\text{fix}}(\beta\Delta f) = 1$. In order to show that $p_{\text{fix}}(\Delta f)$ is monotonically increasing for $N > 1$ we show that $\frac{\partial\, p_{\text{fix}}(\Delta f)}{\partial\, \Delta f} > 0$ for all $\Delta f \neq 0$. The derivative is

$$\frac{\partial\, p_{\text{fix}}(\Delta f)}{\partial\, \Delta f} = \frac{2\beta \cdot e^{-2\beta\Delta f}}{1 - e^{-2N\beta\Delta f}} - \frac{2N\beta \cdot e^{-2N\beta\Delta f}(1 - e^{-2\beta\Delta f})}{(1 - e^{-2N\beta\Delta f})^2}.$$

Dividing the derivative by 2β and multiplying by $(1 - e^{-2N\beta\Delta f})^2$, which for $\Delta f \neq 0$ is always positive, we get

$$\begin{aligned}\frac{(1 - e^{-2N\beta\Delta f})^2}{2\beta}\frac{\partial\, p_{\text{fix}}(\Delta f)}{\partial\, \Delta f} &= e^{-2\beta\Delta f}\left(1 - e^{-2N\beta\Delta f}\right) - N \cdot e^{-2N\beta\Delta f}\left(1 - e^{-2\beta\Delta f}\right)\\ &= e^{-2(N+1)\beta\Delta f}\left(e^{2N\beta\Delta f} + N - 1 - Ne^{2\beta\Delta f}\right).\end{aligned}$$

Hence $\frac{\partial\, p_{\text{fix}}(\Delta f)}{\partial\, \Delta f} > 0$ if and only if

$$e^{2N\beta\Delta f} + N - 1 > Ne^{2\beta\Delta f}. \tag{12}$$

If $N = 1$ then $e^{2N\beta\Delta f} + N - 1 = Ne^{2\beta\Delta f}$, and from comparing the derivatives w.r.t. N of both sides of (12):

$$\frac{\partial}{\partial\, N}\left(e^{2N\beta\Delta f} + N - 1\right) = 2\beta\Delta f e^{2N\beta\Delta f} + 1 \tag{13}$$

$$\frac{\partial}{\partial\, N} Ne^{2\beta\Delta f} = e^{2\beta\Delta f} \tag{14}$$

we see that for $N = 1$ the derivative (13) is larger than derivative (14). Since expression (13) is an increasing positive function of N for any $\beta\Delta f$ while expression (14) is constant in N, the inequality (12) is established for $N > 1$ and hence $\frac{\partial\, p_{\text{fix}}(\Delta f)}{\partial\, \Delta f} > 0$. □

Restating Lemma 3 *For any positive integer $k > 0$, let $\mathrm{mut}(i, i \pm k)$ for $0 \leq i \leq n$ be the probability that a global mutation of a search point with i ones creates an offspring with $i \pm k$ ones. Then*

$$\begin{aligned}\mathrm{mut}(i, i+k) &\leq \left(\frac{n-i}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \cdot \frac{1.14}{k!}\\ \mathrm{mut}(i, i-k) &\leq \left(\frac{i}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \cdot \frac{1.14}{k!}\\ \mathrm{mut}(i, i-k) &\leq \frac{\mathrm{mut}(i, i-1)}{k!}.\end{aligned}$$

Proof of Lemma 3 During this proof we use that $\binom{a}{b} = 0$ for $b > a$. We follow the proof of Lemma 2 in [30]. An offspring with $i + k$ 1-bits is created if and only if there is an integer $j \in \mathbb{N}_0$ such that j 1-bits flip and $k + j$ 0-bits flip.

$$\begin{aligned}\mathrm{mut}(i, i+k) =& \sum_{j=0}^{n} \binom{i}{j}\binom{n-i}{k+j}\left(\frac{1}{n}\right)^{k+2j}\left(1-\frac{1}{n}\right)^{n-k-2j}\\ =& \left(\frac{1}{n}\right)^{k}\left(1-\frac{1}{n}\right)^{n-k} \cdot \sum_{j=0}^{n} \binom{i}{j}\binom{n-i}{k+j}\left(\frac{1}{n-1}\right)^{2j}.\end{aligned}$$

Using $\binom{n-i}{k+j} = \frac{1}{(k+j)!} \cdot (n-i) \cdot (n-i-1) \cdot \ldots \cdot (n-i-k-j+1) \le \frac{1}{(k+j)!} \cdot (n-i)^k \cdot (n-i-1)^j$, this is at most

$$\mathrm{mut}(i, i+k) \le \left(\frac{1}{n}\right)^{k}\left(1-\frac{1}{n}\right)^{n-k} \cdot \sum_{j=0}^{n} \frac{(n-i)^k}{j!(k+j)!} \cdot \left(\frac{i(n-i-1)}{(n-1)^2}\right)^{j}.$$

It is easy to see that $\frac{i(n-i-1)}{(n-1)^2} \le \frac{1}{4}$ for all i, as the maximum is attained for $i = \frac{n}{2} - \frac{1}{2}$. Hence we get an upper bound of

$$\mathrm{mut}(i, i+k) \le \left(\frac{n-i}{n}\right)^{k}\left(1-\frac{1}{n}\right)^{n-k} \cdot \sum_{j=0}^{n} \frac{4^{-j}}{j!(k+j)!}.$$

Using $(k+j)! \ge k!(j+1)!$ for all $k \in \mathbb{N}$, $j \in \mathbb{N}_0$,

$$\begin{aligned}\mathrm{mut}(i, i+k) \le& \left(\frac{n-i}{n}\right)^{k}\left(1-\frac{1}{n}\right)^{n-k} \cdot \frac{1}{k!}\sum_{j=0}^{\infty} \frac{4^{-j}}{j!(j+1)!}\\ \le& \left(\frac{n-i}{n}\right)^{k}\left(1-\frac{1}{n}\right)^{n-k} \cdot \frac{1.14}{k!}.\end{aligned}$$

The proof for mutations decreasing the number of ones follows immediately due to the symmetry $\mathrm{mut}(i, i-k) = \mathrm{mut}(n-i, n-i+k)$.

Finally, we prove the third claim:

$$\begin{aligned}\mathrm{mut}(i, i-k) =& \binom{i}{k} n^{-k}\left(1-\frac{1}{n}\right)^{n-k}\\ =& \frac{i(i-1) \cdot \ldots \cdot (i-k+1)}{k!} \cdot n^{-k}\left(1-\frac{1}{n}\right)^{n-k}\end{aligned}$$

$$\begin{aligned}
&\leq \frac{i}{n} \cdot \left(\frac{n-1}{n}\right)^{k-1} \cdot \frac{1}{k!}\left(1-\frac{1}{n}\right)^{n-k} \\
&= \frac{i}{n} \cdot \left(1-\frac{1}{n}\right)^{n-1} \cdot \frac{1}{k!} = \frac{\operatorname{mut}(i, i-1)}{k!}.
\end{aligned}$$

□

Restating Lemma 9 *For all* $0 \leq i < j \leq n$,

$$\frac{\operatorname{mut}(i, j)}{\sum_{k=j}^{n} \operatorname{mut}(i, k)} \geq \frac{1}{2}.$$

Proof of Lemma 9 The proof consists of two parts:

(1) The probability of improving by $j - i = k$ bits is at least twice as large as the probability of improving by $k + 1$ bits, i.e. $\operatorname{mut}(i, i + k) \geq 2 \operatorname{mut}(i, i + k + 1)$ for any $0 \leq i < j \leq n$.

(2) We use (1) to prove that $\frac{\operatorname{mut}(i,j)}{\sum_{m=j}^{n} \operatorname{mut}(i,m)} \geq \frac{1}{2}$.

Part 1) The probability to improve by k bits is

$$\operatorname{mut}(i, i+k) = \sum_{l=0}^{n} \binom{i}{l}\binom{n-i}{k+l}\left(\frac{1}{n}\right)^{k+2l}\left(1-\frac{1}{n}\right)^{n-k-2l}$$

while the probability to improve by $k + 1$ bits is

$$\operatorname{mut}(i, i+k+1) = \sum_{l=0}^{n} \binom{i}{l}\binom{n-i}{k+l+1}\left(\frac{1}{n}\right)^{k+2l+1}\left(1-\frac{1}{n}\right)^{n-k-2l-1}.$$

We want to show that the following is true:

$$\begin{aligned}
&\operatorname{mut}(i, i+k) \geq 2 \operatorname{mut}(i, i+k+1) \\
&\Leftrightarrow \sum_{l=0}^{n} \binom{i}{l}\binom{n-i}{k+l}\left(\frac{1}{n}\right)^{k+2l}\left(1-\frac{1}{n}\right)^{n-k-2l} \\
&\geq 2\sum_{l=0}^{n} \binom{i}{l}\binom{n-i}{k+l+1}\left(\frac{1}{n}\right)^{k+2l+1}\left(1-\frac{1}{n}\right)^{n-k-2l-1} \\
&\Leftrightarrow \sum_{l=0}^{n} \binom{i}{l}\binom{n-i}{k+l}(n-1)^{n-k-2l} \geq 2\sum_{l=0}^{n} \binom{i}{l}\binom{n-i}{k+l+1}(n-1)^{n-k-2l-1} \\
&\Leftrightarrow \sum_{l=0}^{n} \frac{i!(n-i)!}{l!(i-l)!}\frac{(n-1)^{n-k-2l}}{(n-i-k-l-1)!(k+l)!}\left[\frac{1}{(n-i-k-l)} - \frac{2}{(n-1)(k+l+1)}\right] \geq 0.
\end{aligned}$$

This holds if following holds for any $0 \le l \le n$

$$\left[\frac{1}{(n-i-k-l)} - \frac{2}{(n-1)(k+l+1)}\right] \ge 0$$
$$\Leftrightarrow (n-1)(k+l+1) \ge 2(n-i-k-l).$$

This is true for any $k \ge 1$ (thus for any $0 \le i < j \le n$).

Part 2) Using the above inequality $\mathrm{mut}(i, i+k) \ge 2\,\mathrm{mut}(i, i+k+1)$ we can bound every possible improvement better than k from above by

$$\mathrm{mut}(i, i+k+l) \le \left(\frac{1}{2}\right)^l \mathrm{mut}(i, i+k)$$

for any $0 \le l \le n-i-k$. This can also be written as

$$\mathrm{mut}(i, j+l) \le \left(\frac{1}{2}\right)^l \mathrm{mut}(i, j)$$

for any $0 \le l \le n-j$. This leads to

$$\begin{aligned}\frac{\mathrm{mut}(i,j)}{\sum_{m=j}^{n}\mathrm{mut}(i,m)} &= \frac{\mathrm{mut}(i,j)}{\sum_{l=0}^{n-j}\mathrm{mut}(i,j+l)}\\ &\ge \frac{\mathrm{mut}(i,j)}{\sum_{l=0}^{n-j}\left(\frac{1}{2}\right)^l \mathrm{mut}(i,j)}\\ &= \frac{1}{\sum_{l=0}^{n-j}\left(\frac{1}{2}\right)^l} = \frac{1}{2-\frac{1}{2^{n-j}}} \ge \frac{1}{2}\end{aligned}$$

which proves Lemma 9. □

Lemma 17 *Consider independent random variables $Y_1, \ldots, Y_t$ where*

$$Y_i = \begin{cases} 1 & \text{with probability } p\\ 0 & \text{with probability } 1-p-r\\ -1 & \text{with probability } r\end{cases}$$

then for $Y = \sum_{i=1}^{t} Y_i$ we have $E(Y) = t(p-r)$ and for every $0 \le \delta \le t(p+r)$

$$P(Y \ge E(Y) + \delta) \le e^{-\Omega(t(p+r))} + e^{-\Omega\left(\frac{\delta^2}{t(p+r)}\right)}.$$

Proof We imagine Y_i to be drawn in a two-step process: in a first draw with probability $1-p-r$ we set $Y_i = 0$. Otherwise, we have $Y_i \ne 0$ and a second random experiment determines whether $Y_i = 1$ or $Y_i = -1$.

We define indicator variables $X_i \in \{0, 1\}$ for the first experiment: $X_i = 1$ if $Y_i \neq 0$. Then $X = \sum_{i=1}^{t} X_i$ gives the number of events where $Y_i \neq 0$. Furthermore, let $Z_j \in \{-1, +1\}$ be the outcome of the j-th instance of the second-type experiment (such an experiment only happens when the first draw determined $Y_i \neq 0$), and $Z = \sum_{j=1}^{X} Z_j$ be the sum of these variables. Since Z, in comparison to Y, excludes all summands of value 0, we have $Z = Y$ and hence $\mathrm{E}(Z) = \mathrm{E}(Y) = t(p - r)$.

It is easy to see that $(X < 2\mathrm{E}(X)) \wedge (Z < \mathrm{E}(Z) + \delta \mid X < 2\mathrm{E}(X)) \Rightarrow (Y < \mathrm{E}(Y) + \delta)$. Therefore

$$P(Y \geq \mathrm{E}(Y) + \delta) \leq P(X \geq 2\mathrm{E}(X)) + P(Z \geq \mathrm{E}(Z) + \delta \mid X < 2\mathrm{E}(X)).$$

Now we apply a Chernoff bound to X and a Hoeffding bound to Z under the condition of not having more non-zero variables than $2\mathrm{E}(X)$:

$$\begin{aligned} P(Y \geq \mathrm{E}(Y) + \delta) &\leq e^{-\frac{1}{3}\mathrm{E}(X)} + e^{-\frac{\delta^2}{4\mathrm{E}(X)}} \\ &= e^{-\Omega(\mathrm{E}(X))} + e^{-\Omega\left(\frac{\delta^2}{\mathrm{E}(X)}\right)} \\ &= e^{-\Omega(t(p+r))} + e^{-\Omega\left(\frac{\delta^2}{t(p+r)}\right)}. \end{aligned}$$

□

References

1. Auger, A., Doerr, B. (eds.): Theory of Randomized Search Heuristics-Foundations and Recent Developments. Series on Theoretical Computer Science, vol. 1. World Scientific, Singapore (2011)
2. Chastain, E., Livnat, A., Papadimitriou, C., Vazirani, U.: Algorithms, games, and evolution. Proc. Natl. Acad. Sci. **111**(29), 10620–10623 (2014)
3. Chatterjee, K., Pavlogiannis, A., Adlam, B., Nowak, M.A.: The time scale of evolutionary innovation. PLoS Comput. Biol. **10**(9), 1–7 (2014)
4. Corus, D., Dang, D.-C., Eremeev, A.V., Lehre, P.K.: Level-based analysis of genetic algorithms and other search processes. In: Parallel Problem Solving from Nature (PPSN), Springer, Berlin, pp. 912–921 (2014)
5. Doerr, B.: Analyzing Randomized Search Heuristics: Tools from Probability Theory. In: [1], pp. 1–20. World Scientific, Singapore (2011)
6. Eiben, A .E., Smith, J .E.: Introduction to Evolutionary Computing, 2nd edn. Springer, Berlin (2015)
7. Ewens, W.J.: Mathematical Population Genetics 1: Theoretical Introduction, 2nd edn. Springer, New York (2004)
8. Gillespie, J.H.: Molecular evolution over the mutational landscape. Evolution **38**(5), 1116–1129 (1984)
9. Jägersküpper, J., Storch, T.: When the plus strategy outperforms the comma strategy and when not. In: Proceedings of IEEE Foundations of Computational Intelligence (FOCI 2007), pp. 25–32. IEEE (2007)
10. Jansen, T.: Analyzing Evolutionary Algorithms. The Computer Science Perspective. Springer, Berlin (2013)
11. Jansen, T., Oliveto, P.S., Zarges, C.: On the analysis of the immune-inspired B-Cell algorithm for the Vertex Cover problem. In: Proceedings of the International Conference on Artificial Immune Systems (ICARIS '11), Springer, Berlin, pp. 117–131 (2011)
12. Jansen, T., Wegener, I.: A comparison of simulated annealing with a simple evolutionary algorithm on pseudo-Boolean functions of unitation. Theor. Comput. Sci. **386**(1–2), 73–93 (2007)
13. Johannsen, D.: Random Combinatorial Structures and Randomized Search Heuristics. Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany and the Max-Planck-Institut für Informatik (2010)
14. Kimura, M.: On the probability of fixation of mutant genes in a population. Genetics **47**(6), 713–719 (1962)

15. Lehre, P.K., Witt, C.: Black-box search by unbiased variation. Algorithmica **64**(4), 623–642 (2012)
16. Neumann, F.: Expected runtimes of evolutionary algorithms for the Eulerian cycle problem. Comput. Oper. Res. **35**(9), 2750–2759 (2008)
17. Neumann, F., Oliveto, P.S., Witt, C.: Theoretical analysis of fitness-proportional selection: landscapes and efficiency. In: Proceedings of the 2009 Genetic and Evolutionary Computation Conference (GECCO '09), pp. 835–842, ACM (2009)
18. Neumann, F., Wegener, I.: Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. Theor. Comput. Sci. **378**(1), 32–40 (2007)
19. Neumann, F., Witt, C.: Runtime analysis of a simple ant colony optimization algorithm. Algorithmica **54**(2), 243–255 (2009)
20. Neumann, F., Witt, C.: Bioinspired Computation in Combinatorial Optimization—Algorithms and Their Computational Complexity. Springer, Berlin (2010)
21. Oliveto, P.S., Sudholt, D.: On the runtime analysis of stochastic ageing mechanisms. In: Proceedings of the 2014 Genetic and Evolutionary Computation Conference (GECCO '14), ACM Press, pp. 113–120 (2014)
22. Oliveto, P.S., Witt, C.: Simplified drift analysis for proving lower bounds in evolutionary computation. Algorithmica **59**(3), 369–386 (2011)
23. Oliveto, P.S., Witt, C.: On the runtime analysis of the simple genetic algorithm. Theor. Comput. Sci. **545**, 2–19 (2014)
24. Paixão, T., Badkobeh, G., Barton, N., Çörüş, D., Dang, D.-C., Friedrich, T., Lehre, P.K., Sudholt, D., Sutton, A.M., Trubenová, B.: Toward a unifying framework for evolutionary processes. J. Theor. Biol. **383**, 28–43 (2015)
25. Paixão, T., Pérez Heredia, J., Sudholt, D., Trubenová, B.: First steps towards a runtime comparison of natural and artificial evolution. In: Proceedings of the 2015 Genetic and Evolutionary Computation Conference (GECCO '15), pp. 1455–1462, ACM (2015)
26. Reichel, J., Skutella, M.: Evolutionary algorithms and matroid optimization problems. Algorithmica **57**(1), 187–206 (2010)
27. Rohlfshagen, P., Lehre, P.K., Yao, X.: Dynamic evolutionary optimisation: an analysis of frequency and magnitude of change. In: Proceedings of the 2009 Genetic and Evolutionary Computation Conference (GECCO '09), ACM Press, pp. 1713–1720 (2009)
28. Rowe, J.E., Sudholt, D.: The choice of the offspring population size in the $(1,\lambda)$ evolutionary algorithm. Theor. Comput. Sci. **545**, 20–38 (2014)
29. Scharnow, J., Tinnefeld, K., Wegener, I.: The analysis of evolutionary algorithms on sorting and shortest paths problems. J. Math. Modell. Algorithms **3**(4), 349–366 (2004)
30. Sudholt, D.: A new method for lower bounds on the running time of evolutionary algorithms. IEEE Trans. Evol. Comput. **17**(3), 418–435 (2013)
31. Sudholt, D., Thyssen, C.: A simple ant colony optimizer for stochastic shortest path problems. Algorithmica **64**(4), 643–672 (2012)
32. Traulsen, A., Iwasa, Y., Nowak, M.A.: The fastest evolutionary trajectory. J. Theor. Biol. **249**(3), 617–623 (2007)
33. Valiant, L .G.: Evolvability. J. ACM **56**(1), 3:1–3:21 (2009)
34. Wegener, I.: Methods for the analysis of evolutionary algorithms on pseudo-boolean functions. In: Sarker, R., Mohammadian, M., Yao, X. (eds.) Evolutionary Optimization, volume 48 of International Series in Operations Research & Management Science, chapter 14. Kluwer Academic Publishers, Dordrecht, pp. 349–369 (2003)
35. Witt, C.: Worst-case and average-case approximations by simple randomized search heuristics. In: Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science (STACS '05), Springer, Berlin, pp. 44–56 (2005)
36. Witt, C.: Population size versus runtime of a simple evolutionary algorithm. Theor. Comput. Sci. **403**(1), 104–120 (2008)