First Steps Towards a Runtime Comparison of Natural and Artificial Evolution

Tiago Paixão
IST Austria
Am Campus 1, 3400
Klosterneuburg, Austria

Jorge Pérez Heredia
University of Sheffield
Sheffield, S1 4DP, United Kingdom

Dirk Sudholt
University of Sheffield
Sheffield, S1 4DP, United Kingdom

Barbora Trubenová
IST Austria
Am Campus 1, 3400
Klosterneuburg, Austria

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 their runtime 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 occurrence of new mutations is much longer than the time it takes for a new beneficial mutation to take over the population. In this situation, the population only contains copies of one genotype and evolution can be modelled as a (1+1)-type process where the probability of accepting a new genotype (improvements or worsenings) depends on the change in fitness.

We present an initial runtime analysis of SSWM, quantifying its performance for various parameters and investigating differences to the (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.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

Keywords

Runtime analysis, natural evolution, population genetics, theory, strong selection weak mutation regime

1. INTRODUCTION

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 runtime, and allow for rigorous bounds on the time required to reach a global optimum, or other well-specified high-fitness solutions. 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, 8, 14] as well as novel optimisation paradigms such as swarm intelligence [14] and artificial immune systems [9].

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 [22] 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 the population size, mutation rate, and selection strength are such that the time between occurrence of new mutations is long compared to the time a new genotype takes to replace the parent genotype [6]. Under these conditions, only one genotype is present in the population most of the time, and evolution occurs through "jumps" between different genotypes, corresponding to a new mutation replacing the resident genotype in the population. The relevant dynamics can then be characterized by a (1+1)-type stochastic process. This model is obtained as a limit of many other models, such as the Wright-Fisher model. One important aspect of this model is that new solutions are accepted with a probability $\frac{1-e^{-2\beta\Delta f}}{1-e^{-2N\beta\Delta f}}$ that depends on the fitness difference Δf between the new mutation and the resident genotype. Here

GECCO '15, July 11 - 15, 2015, Madrid, Spain

ISBN 978-1-4503-3472-3/15/07...$15.00

DOI: http://dx.doi.org/10.1145/2739480.2754758

N reflects the size of the underlying population, 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 [12] 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, solutions of lower fitness (worsenings) may be accepted with some positive probability. This is reminiscent of the Metropolis algorithm (Simulated Annealing with constant temperature) which can also accept worsenings (see, e. g. [10]). Second, 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 in a (1+1)-type algorithm referred to as SSWM, using common mutation operators from evolutionary algorithms. We then present first runtime analyses of this process. 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.

Our results are summarised as follows. For the simple function OneMax we show in Section 3 that with suitably large population sizes, when $N\beta \geq \frac{1}{2}\ln(11n)$, SSWM is an effective hill climber as it optimises OneMax in expected time $O((n \log n)/\beta)$. However, when the population size is by any constant factor smaller than this threshold, we encounter a phase transition and SSWM requires exponential time even on OneMax.

We then illustrate the particular features of the selection rule. In Section 4 we consider a function Cliff$_d$ where a fitness valley of Hamming distance d needs to be crossed. 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 Section 5 we illustrate on the function Balance [19] 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, 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. 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 probability of fixation and hence the absolute 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.

Some proofs were shortened or omitted due to space restrictions; we refer to a full version of this work [18].

2. PRELIMINARIES

We define the optimisation time of SSWM as the first generation where the optimum is accepted as new individual.

The model resembles the (1+1) EA in that it only maintains one genotype that may be replaced by mutated versions of it. Mutants 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. If $N = 1$, we get $p_{\text{fix}}(s) = 1$, meaning that any offspring will be accepted, and if $N \to \infty$, it will only accept solution for which $\Delta f > 0$. This expression was first derived by Kimura [12] and represents the *probability of fixation*, that is, the probability that a gene that is 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 absolute difference in fitness between genotypes, we include a parameter $\beta \in (0, 1]$ 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. This function has a sigmoid shape (strictly increasing—see [18, Lemma 15]) with limits $\lim_{\Delta f \to -\infty} p_{\text{fix}}(\Delta f) = 0$ and $\lim_{\Delta f \to \infty} p_{\text{fix}}(\Delta f) = 1$. 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. By bounding β to 1, we avoid artefactual results obtained by inflating the fitness differences between genotypes.

We can then cast the SSWM regime as Algorithm 1, where the function mutate(x) can be either standard bit mutation (mutate all bits independently with probability $p_m = 1/n$, called *global mutations*) or flipping a single bit chosen uniformly at random (called *local mutations*). SSWM is valid 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.

Next, we derive upper and lower bounds for $p_{\text{fix}}(\Delta f)$ that will be useful throughout the manuscript.

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

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

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
```

3. SSWM ON ONEMAX

The function $\text{ONEMAX}(x) := \sum_{i=1}^{n} x_i$ 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 [21]. 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 effect, we first show a polynomial upper bound for the runtime of SSWM on ONEMAX. We then show that SSWM exhibits a phase transition on its runtime as a function of $N\beta$; changing this parameter by a constant factor 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. Oliveto and Witt [17] showed that the Simple Genetic Algorithm, which uses fitness-proportional selection, fails badly on ONEMAX even within exponential time, with a 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 2. *For any positive integer $k > 0$, let $\text{mut}(i, i \pm k)$ 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*

$$\text{mut}(i, i+k) \leq \left(\frac{n-i}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \cdot \frac{1.14}{k!}$$

$$\text{mut}(i, i-k) \leq \left(\frac{i}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \cdot \frac{1.14}{k!}.$$

The proof is omitted (see [18, Appendix]); it uses arguments from the proof of Lemma 2 in [21]. The second inequality follows immediately from the first one due to the symmetry $\text{mut}(i, i-k) = \text{mut}(n-i, n-i+k)$.

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, for all $1 \leq i \leq n$ the forward and backward drifts are*

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

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

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

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

LEMMA 3. *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*

$$\Delta^+(i) \geq \frac{n-i}{n} \left(1 - \frac{1}{n}\right)^{n-1} p_{\text{fix}}(1)$$

$$|\Delta^-(i)| \leq 1.14 \left(1 - \frac{1}{n}\right)^{n-1} \cdot \left(p_{\text{fix}}(-1) + e \cdot p_{\text{fix}}(-2)\right).$$

For local mutations the relations are as follows

$$\Delta^+(i) = \frac{n-i}{n} \cdot p_{\text{fix}}(1)$$

$$|\Delta^-(i)| \leq p_{\text{fix}}(-1).$$

Proof. For global mutations firstly we compute the lower bound for the forward drift,

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

where $\text{mut}(i, i+j)$ is the probability of mutation increasing the ONEMAX value by j and i is the number of ones of the current search point.

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

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. We can upper bound $\text{mut}(i, i-j)$ for the probability of flipping any j bits, which from Lemma 2 yields

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

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

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

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 for $N\beta \geq \frac{1}{2}\ln(11n)$, 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). The upper bound increases with $1/\beta$; this makes sense as for small values of β we have $p_{\text{fix}}(1) \approx 2\beta$ (cf. Lemma 1). In this regime absolute fitness differences are small and improvements are only accepted with a small probability.

Theorem 4. *If $N\beta \geq \frac{1}{2}\ln(11n)$ and $\beta \in (0,1]$ then the expected optimisation time of SSWM on* OneMax *with local or global mutations is $O\left(\frac{n\log n}{\beta}\right)$ for every initial search point.*

Proof. Introducing $N\beta \geq \frac{1}{2}\ln(11n)$ in the fixation probabilities, we get

$$\begin{aligned}p_{\text{fix}}(1) &= \frac{1-e^{-2\beta}}{1-e^{-2N\beta}} \geq \frac{1-e^{-2\beta}}{1-(11n)^{-1}} = \frac{11(1-e^{-2\beta})}{11-\frac{1}{n}}\\ p_{\text{fix}}(-1) &= \frac{e^{2\beta}-1}{e^{2N\beta}-1} \leq \frac{e^{2\beta}-1}{11n-1} \qquad (2)\\ p_{\text{fix}}(-2) &= \frac{e^{4\beta}-1}{e^{4N\beta}-1} \leq \frac{e^{4\beta}-1}{(11n)^2-1} = O(n^{-2}).\end{aligned}$$

Plugging these bounds into the results of Lemma 3

$$\Delta(i) \geq \frac{1}{e}\left[(n-i)\cdot\frac{11(1-e^{-2\beta})}{11n-1} - 1.14\frac{e^{2\beta}-1}{11n-1} - O(n^{-2})\right]$$

since we need a positive net drift even in the last step ($n-i=1$) we lower bound the function $\frac{11(1-e^{-2\beta})-1.14\cdot(e^{2\beta}+1)}{11n-1}$, which is concave for $\beta \in (0,1]$, with a straight line going through the points $\beta = 0$ and 1. Using $\frac{1}{5n}$ as a lower bound for the height of this second point we obtain

$$\geq \frac{1}{e}\cdot\left(\frac{n-i}{n}\cdot\frac{\beta}{5}\right) - O(n^{-2}) \geq \frac{\beta}{14}\cdot\frac{n-i}{n} \qquad (3)$$

for sufficiently large values of n.

Now we apply Johannsen's variable drift theorem [11] to the number of zeros. Using $h(z) := E(X_t - X_{t+1} \mid X_t = z)$ then

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

where z is the number of zeros, X_t the current state and T the optimisation time. Introducing $z_{\min} = 1$, $X_0 = n$ and

$$\Delta(i) \geq \frac{z\beta}{14n} = h(z)$$

we obtain an upper bound for the runtime

$$\begin{aligned}E(T \mid X_0) &\leq \frac{1}{h(1)} + \int_1^n \frac{1}{h(z)}dz \leq \frac{14n}{\beta} + \int_1^n \frac{14n}{\beta z}dz\\ &= \frac{14n}{\beta}(1+\log n) = O\left(\frac{n\log n}{\beta}\right).\end{aligned}$$ □

3.2 A Critical Threshold for SSWM on OneMax

The upper bound from Theorem 4 required $N\beta \geq \frac{1}{2}\ln(11n) = \frac{1}{2}\ln(n) + O(1)$. 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 $N\beta$ is by a factor of $1-\varepsilon$, for some constant $\varepsilon > 0$, smaller than $\frac{1}{2}\ln n$, 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 5. *If $1 \leq N\beta \leq \frac{1-\varepsilon}{2}\ln n$ 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 idea behind the proof of Theorem 5 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 [20] (an extension of the negative drift theorem [16] to stochastic processes with large self-loop probabilities). The proof of Theorem 5 applies this theorem with respect to the number of zeros on an interval of $[0, n^{\varepsilon/2}]$. A full proof can be found in [18].

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.

Jansen and Wegener [10] 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 [10, Section 6].

We consider a similar function, generalising a construction by Jägersküpper and Storch [7]: the function 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 Figure 1).

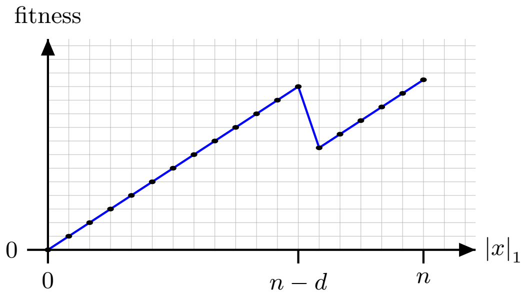

Figure 1: Sketch of the function $\textsc{Cliff}_d$**.**

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 6. *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 6, 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 7. *For all* $0 \leq i < j \leq n$,

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

The proof of this lemma is presented in [18].

Proof of Theorem 6. 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 7 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 $\frac{1}{n^d}(1-\frac{1}{n})^{n-d}$ 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 $e^{\Omega(d)}$ 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 8. *The expected optimisation time of SSWM with global mutations and* $\beta = 1, N = \frac{1}{2}\ln(11n)$ *on* $\textsc{Cliff}_d$ *with* $d = \omega(\log n)$ *is at most* $n^d/e^{\Omega(d)}$.

Proof sketch for Theorem 8. 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_{peak} be the random optimisation time when starting with any search point of $n-d$ ones, hereinafter called a *peak*. Then the expected optimisation time from any initial point is at most $R + \mathrm{E}(T_{\text{peak}})$. Let p_{success} be the probability of SSWM starting in a peak will reach the optimum before reaching a peak again. We call such a time period a *trial*. After the end of a trial, taking at most R expected generations, with probability $1 - p_{\text{success}}$ SSWM returns to a peak again, so

$$\mathrm{E}(T_{\text{peak}}) \leq R + (1 - p_{\text{success}}) \cdot \mathrm{E}(T_{\text{peak}})$$
$$\Leftrightarrow \mathrm{E}(T_{\text{peak}}) \leq \frac{R}{p_{\text{success}}}. \tag{4}$$

We first bound the worst-case time to return to a 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 essentially behaves like on OneMax. Repeating arguments from the proof of Theorem 4, in expected time $O((n \log n)/\beta) = O(n \log n)$ (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 a peak 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 or peak is reached. The conditional probability of creating a peak—when from a search point with $i < n-d$ ones either a peak or a non-optimal search point in S_2 is reached—is

$$\frac{\text{mut}(i, n-d) \cdot p_{\text{fix}}(n-d-i)}{\sum_{k=n-d}^{n-1} \text{mut}(i,k) \cdot p_{\text{fix}}(k-i-d+1/2)} \geq \frac{\text{mut}(i,j)}{\sum_{k=j}^{n} \text{mut}(i,k)}$$

as $p_{\text{fix}}(n-d-i) \geq p_{\text{fix}}(k-i-d+1/2)$ for all $n-d < k < n$. By Lemma 7, the above fraction is at least 1/2. Hence SSWM in expectation only makes $O(1)$ transitions from S_1 to S_2, 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 next mutation creates a search point with $n-d+k$ ones, for some integer $1 \leq k \leq d$, that this point is accepted, and that from there the global optimum is reached before returning to a peak.

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 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

$$\text{mut}(n-d, n-d+k) \geq \frac{1}{n^k} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \cdot \binom{d}{k}$$
$$\geq \frac{1}{en^k} \cdot \left(\frac{d}{k}\right)^k.$$

The probability of accepting such a move is

$$p_{\text{fix}}(k-d+1/2) = \frac{e^{2\beta(d-k-1/2)}-1}{e^{2N\beta(d-k-1/2)}-1} \geq \frac{e^{2(d-k-1/2)}-1}{(11n)^{(d-k-1/2)}}.$$

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

$$\begin{aligned}
&\text{mut}(n-d, n-d+k) \cdot p_{\text{fix}}(k-d+1/2) \\
&\geq \frac{1}{en^k} \cdot \left(\frac{d}{k}\right)^k \cdot \frac{e^{2(d-k-1/2)}-1}{(11n)^{(d-k-1/2)}} \\
&= \Omega\left(n^{-d+1/2} \cdot \left(\frac{d}{k}\right)^k \cdot \left(\frac{e^2}{11}\right)^{d-k}\right) \\
&= \Omega\left(n^{-d+1/2} \cdot \left(e^{1/e} \cdot \left(\frac{e^2}{11}\right)^{1-1/e}\right)^d\right) \\
&= \Omega\left(n^{-d+1/2} \cdot \left(\frac{10}{9}\right)^d\right).
\end{aligned}$$

Finally, we show that, if SSWM does make this accepted jump, with probability $1 - n^{-\omega(1)}$ it climbs up to the global optimum before returning to a search point in S_1. To this end we apply the negative drift theorem with self-loops [20] 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. Details of this application are given in [18].

This implies that following a length-k jump, a trial is successful with probability $1 - n^{-\omega(1)}$. This establishes $p_{\text{success}} := \Omega\left(n^{-d+1/2} \cdot \left(\frac{10}{9}\right)^d\right)$. Plugging this into (4), adding time R for the time to reach the peak initially, and using that $O(n^{1/2} \log n) \cdot (9/10)^d = e^{-\Omega(d)}$ 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, Lehre, and Yao [19] 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 [15].

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 Figure 2).

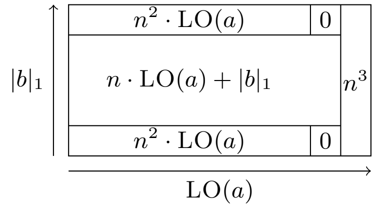

Figure 2: Visualisation of Balance [19].

Definition 3 (Balance [19]). *Let $a, b \in \{0,1\}^{n/2}$ and $x = ab \in \{0,1\}^n$. Then,* $\text{Balance}(x) =$

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

where $|x|_1 = \sum_{i=1}^{n/2} x_i$, $|x|_0$ is a number of zeros and $\text{LO}(x) := \sum_{i=1}^{n/2} \prod_{j=1}^{i} x_j$ 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, Lehre, and Yao [19, Theorem 3] showed the following lower bound for the (1+1) EA, specialised to non-dynamic optimisation:

Theorem 9 ([19]). *The expected optimisation time of the (1+1) EA on* Balance *is $n^{\Omega(n^{1/2})}$.*

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 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 10. *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 $\text{LO}(a') < \text{LO}(a)$ 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

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

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

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

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 11. *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 (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. 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 this bit position after mutation. The latter bits are called *free riders* and it is well known (see [13, 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 $T = \mathrm{poly}\,(n)$ as $p_{\text{fix}}(n - \sqrt{n}) \geq 1/N \geq 1/\mathrm{poly}\,(n)$). 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 10 and the expected number of non-relevant steps before T relevant steps have occurred is $O(T)$.

We now have that 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 $n/4 + n^{3/4}/4$. By Chernoff bounds [5], the probability that less than $n/4 + n^{3/4}/8$ improvements happen is $e^{-\Omega(n^{1/2})}$. Also the probability that during this number of improvements less than $n/4 - n^{3/4}/8$ free riders occur is $e^{-\Omega(n^{1/2})}$. If these two rare events do not happen, a LO-value of $n/2$ is reached before time T. Taking the union bound over all rare failure probabilities proves the claim. □

We now show that the ONEMAX part is not optimized before the LO part.

LEMMA 12. *Let $\beta = n^{-3/2}$, $N\beta = \ln n$, and T be as in Lemma 11. The probability that SSWM starting with $a_0 b_0$ 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 12 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 sketch for Lemma 12. 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.

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 2 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

$$\begin{aligned} &(1 + O(n^{-1/4})) \cdot \frac{4}{3} \cdot p^+ \cdot T \\ &= (1 + O(n^{-1/4})) \cdot \frac{16}{9} \cdot p_1 \cdot T \\ &\leq (1 + O(n^{-1/4})) \cdot \frac{1.14}{9} \cdot n. \end{aligned}$$

We also have at most $n/2$ steps where the number of leading ones increases. If the number of leading ones increases by $d \geq 1$, the fitness increase is $dn + |b'|_1 - |b|_1$. Hence the above estimations of jump lengths are not applicable. We call these *special* steps; they are unorthodox as the large fitness increase makes it likely that any mutation on the OneMax part is accepted. Note that this applies to mutations increasing $|b|_1$ as well as to those decreasing it. By similar arguments as used before, in at most $n/2$ special steps, the net gain in the number of ones is at most $O(n^{3/4})$ with probability $1 - e^{-\Omega(n^{1/2})}$. For details we refer to [18].

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 $|b|_1 \geq 7n/16$ 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 13. *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 $n/4 - n^{3/4} \leq |b_0|_1 \leq n/4 + n^{3/4}$ is $1 - e^{-\Omega(n^{1/2})}$. We assume pessimistically that $n/4 \leq |b_0|_1 \leq n/4 + n^{3/4}$. Then Lemma 12 is in force, and with probability $1 - e^{-\Omega(n^{1/2})}$ within T relevant steps, T as defined in Lemma 11, SSWM does not reach a trap or a search point with fitness 0. Lemma 11 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 $T = O(n/\beta)$ 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, and uncovered constraints on biologically relevant parameters such as population size and strength of selection, imposing fundamental limits to what is possible by natural selection.

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 GECCO reviewers for their many constructive comments.

7. REFERENCES

[1] A. Auger and B. Doerr, editors. *Theory of Randomized Search Heuristics – Foundations and Recent Developments.* Number 1 in Series on Theoretical Computer Science. World Scientific, 2011.

[2] E. Chastain, A. Livnat, C. Papadimitriou, and U. Vazirani. Algorithms, games, and evolution. *Proceedings of the National Academy of Sciences*, 111(29):10620–10623, July 2014.

[3] K. Chatterjee, A. Pavlogiannis, B. Adlam, and M. A. Nowak. The time scale of evolutionary innovation. *PLoS Computational Biology*, 10(9), Sept. 2014.

[4] D. Corus, D.-C. Dang, A. V. Eremeev, and P. K. Lehre. Level-based analysis of genetic algorithms and other search processes. In *PPSN 2014*, pages 912–921. Springer, 2014.

[5] B. Doerr. Analyzing randomized search heuristics: Tools from probability theory. In *[1]*, pages 1–20. World Scientific, 2011.

[6] J. H. Gillespie. Molecular evolution over the mutational landscape. *Evolution*, 38(5):1116–1129, 1984.

[7] J. Jägersküpper and T. Storch. When the plus strategy outperforms the comma strategy and when not. In *Proc. of IEEE FOCI 2007*, pages 25–32. IEEE, 2007.

[8] T. Jansen. *Analyzing Evolutionary Algorithms. The Computer Science Perspective.* Springer, 2013.

[9] T. Jansen, P. S. Oliveto, and C. Zarges. On the analysis of the immune-inspired B-Cell algorithm for the Vertex Cover problem. In *Proc. of ICARIS 2011*, pages 117–131. Springer, 2011.

[10] T. Jansen and I. Wegener. A comparison of simulated annealing with a simple evolutionary algorithm on pseudo-Boolean functions of unitation. *Theoretical Computer Science*, 386(1-2):73–93, 2007.

[11] D. Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany and the Max-Planck-Institut für Informatik, 2010.

[12] M. Kimura. On the probability of fixation of mutant genes in a population. *Genetics*, 47(6):713–719, 1962.

[13] P. K. Lehre and C. Witt. Black-box search by unbiased variation. *Algorithmica*, 64(4):623–642, 2012.

[14] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity.* Springer, 2010.

[15] P. S. Oliveto and D. Sudholt. On the runtime analysis of stochastic ageing mechanisms. In *Proc. of GECCO 2014*, pages 113–120. ACM Press, 2014.

[16] P. S. Oliveto and C. Witt. Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica*, 59(3):369–386, 2011.

[17] P. S. Oliveto and C. Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 545:2–19, 2014.

[18] T. Paixão, J. Pérez Heredia, D. Sudholt, and B. Trubenová. First steps towards a runtime comparison of natural and artificial evolution. *ArXiv e-prints*, 2015. Available from http://arxiv.org/abs/1504.06260.

[19] P. Rohlfshagen, P. K. Lehre, and X. Yao. Dynamic evolutionary optimisation: an analysis of frequency and magnitude of change. In *Proc. of GECCO '09*, pages 1713–1720. ACM Press, 2009.

[20] J. E. Rowe and D. Sudholt. The choice of the offspring population size in the (1,λ) evolutionary algorithm. *Theoretical Computer Science*, 545:20–38, 2014.

[21] D. Sudholt. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 17(3):418–435, 2013.

[22] L. G. Valiant. Evolvability. *J. ACM*, 56(1):3:1–3:21, 2009.