Proceedings Paper:
Nallaperuma, S., Oliveto, P.S., Perez Heredia, J. et al. (1 more author) (2017) When is it Beneficial to Reject Improvements? In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 17). Genetic and Evolutionary Computation Conference (GECCO 17), 15/07/2017 - 19/07/2017, Berlin. ACM , pp. 1391-1398.

https://doi.org/10.1145/3071178.3071273

When is it Beneficial to Reject Improvements?

Samadhi Nallaperuma
University of Sheffield, Sheffield, United Kingdom

Pietro S. Oliveto
University of Sheffield, Sheffield, United Kingdom

Jorge Pérez Heredia
University of Sheffield, Sheffield, United Kingdom

Dirk Sudholt
University of Sheffield, Sheffield, United Kingdom

ABSTRACT

We investigate two popular trajectory-based algorithms from biology and physics to answer a question of general significance: when is it beneficial to reject improvements? A distinguishing factor of SSWM (Strong Selection Weak Mutation), a popular model from population genetics, compared to the Metropolis algorithm (MA), is that the former can reject improvements, while the latter always accepts them. We investigate when one strategy outperforms the other. Since we prove that both algorithms converge to the same stationary distribution, we concentrate on identifying a class of functions inducing large mixing times, where the algorithms will outperform each other over a long period of time. The outcome of the analysis is the definition of a function where SSWM is efficient, while Metropolis requires at least exponential time.

KEYWORDS

theory; evolutionary algorithms; non-elitism; Metropolis algorithm; sswm

ACM Reference format:
Samadhi Nallaperuma, Pietro S. Oliveto, Jorge Pérez Heredia, and Dirk Sudholt. 2017. When is it Beneficial to Reject Improvements?. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071273

1 INTRODUCTION

The Strong Selection Weak Mutation (SSWM) algorithm is a recent randomised search heuristic inspired by the popular model of biological evolution in the 'strong selection, weak mutation regime' [13]. The regime applies when mutations are rare and selection is strong enough such that new genotypes either replace the parent population or are lost completely before further mutations occur [5, 7].

The SSWM algorithm belongs to the class of trajectory-based search heuristics that evolve a single lineage rather than using a population. Amongst single trajectory algorithms, well-known ones are (randomised) local search, simulated annealing, the Metropolis algorithm (MA)—simulated annealing with fixed temperature—and simple classes of evolutionary algorithms such as the well-studied (1+1) EA and the (1+λ) EA. The main differences between SSWM and the (1+1) EA is that the latter only accepts new solutions if they are at least as good as the previous ones, while SSWM can reject improvements and it may also accept non-improving solutions with some probability. This characteristic may allow SSWM to escape local optima by gradually descending the slope leading to the optimum rather than relying on large, but rare, mutations to a point of high fitness far away.

A recent study has rigorously analysed the performance of SSWM in comparison with the (1+1) EA for escaping local optima [10]. The study only allowed SSWM to use local mutations such that the algorithm had to rely exclusively on its non-elitism to escape local optima, hence to highlight the differences between elitist and non-elitist strategies. A vast class of fitness functions, called fitness valleys, was considered. These valleys consist of paths between consecutive local optima where the mutation probability of going forward on the path is the same as going backwards. However, the valleys may have arbitrary length and arbitrary depth, where the length is measured by the hamming distance while the depth is the maximal fitness difference that has to be overcome.

The analysis revealed that the expected time of the (1+1) EA to cross the valley (i.e. escape the local optimum) is exponential in the length of the valley while the expected time for SSWM can be exponential in the depth of the valley.

The analysis revealed that other non-elitist trajectory-based algorithms such as the well-known Metropolis algorithm have the same asymptotic runtime as SSWM on fitness valleys, independent of lengths and depths. While it may not be surprising that both algorithms rely on non-elitism to descend the valleys, it is not necessarily obvious that the algorithms should have the same runtime on the valleys, because they differ significantly in the probability of accepting improving solutions. While Metropolis always accepts improvements, SSWM may reject an improving solution with a probability that depends on the difference between the quality of the new and the previous solution.

In this paper we investigate SSWM and Metropolis with the goal of identifying function characteristics for which the two algorithms perform differently. Given that the main difference between the two is that SSWM may reject improvements, we aim to identify a class of functions where it is beneficial to do so and, as a result, identify an example where SSWM outperforms Metropolis.

The roadmap is as follows. After introducing the algorithms precisely in the Preliminaries section, we show in Section 3 that our task is not trivial by proving that both algorithms converge to the same stationary distribution for equivalent parameters. While this result seems to have been known in evolutionary biology [15] we are not aware of a previous proof in the literature. In Section 4 we define a simple fitness function (i.e. 3 State Model) where two possible choices may be made from the initial point; one leading to a much larger fitness than the other. The idea is that, while Metropolis should be indifferent to the choice, SSWM should pick one choice more often than the other. Although this intuition is true, it turns out that, due to Metropolis' ability of escaping local optima, the mixing time for the 3 State Model is small and

GECCO '17, Berlin, Germany
2017. 978-1-4503-4920-8/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071273

afterwards the two algorithms behave equivalently as proven in the previous section. In Section 5 we extend the fitness function (i.e. 5 State Model) by adding two more states of extremely high fitness such that, once the algorithms have made their choice, the probability of escaping the local optima is very low. By tuning these high fitness points we can either reward or penalise a strategy that rejects small improvements. We capitalise on this by concatenating several 5 State models together and by defining a step function that requires that a high number of correct choices are made by the algorithm. Finally, we show that for appropriate fitness values of the different states, SSWM achieves the target and Metropolis doesn't with overwhelming probability. Along the way we complement our theoretical findings with experiments which help to understand the complete picture.

2 PRELIMINARIES

As mentioned in the introduction, we will be considering trajectory based heuristics. The following general scheme considers algorithms with local mutations i.e. only search points that differ in one bit can be sampled. However, the new individual will be accepted or rejected according to a probability function known as the acceptance probability p_{acc}.

Algorithm 1 General Trajectory Based Algorithm

Initialise $x \in \{0,1\}^n$
repeat
  $y \leftarrow$ flip uniformly at random one bit from x
  $\Delta f = f(y) - f(x)$
  Choose $r \in [0,1]$ uniformly at random
  if $r \leq p_{\text{acc}}(\Delta f)$ **then**
    $x \leftarrow y$
  end if
until stop

Two important characteristics of the acceptance probability are how detrimental and beneficial moves are dealt with. Elitist algorithms such as RLS will directly reject any worsening move and accept any improving search point. Hence, an elitist trajectory based algorithm will not be able to escape local optima.

To avoid this weakness, the algorithm must relax its selection strength. This is the case of the Metropolis [9] algorithm where detrimental moves are allowed with some probability, depending on the temperature $1/\alpha$. However, improvements will always be accepted regardless of their magnitude:

$$p_{\text{acc}}^{\text{MA}}(\Delta f) = \begin{cases} 1 & \text{if } \Delta f \geq 0 \\ e^{\alpha \Delta f} & \text{if } \Delta f < 0 \end{cases} \tag{1}$$

To investigate the other main characteristic of non-elitism, allowing the rejection of improvements, we will study a recently introduced algorithm [10, 13, 14] based on the so called SSWM evolutionary regime from Population Genetics (PG). Within this regime a new genotype will eventually take over of a population of size $N \in \mathbb{N}^+$ or become extinct according to the following expression, which depends on the fitness difference and a scaling factor $\beta \in \mathbb{R}^+$ [7]. To cast this regime as an algorithm we simple use the following acceptance probability in Algorithm 1.

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

The following figure presents an example of these two acceptance probabilities. We observe how both algorithms treat worsening moves similarly. The main difference arises when dealing with improvements. Unlike Metropolis, SSWM will prefer to keep the current search point against a small improvement (until $p_{\text{fix}} \geq 1/2$). However when the fitness difference is big enough the algorithm will be satisfied to move to the new solution. This is the crucial feature that we will be exploiting in the following sections.

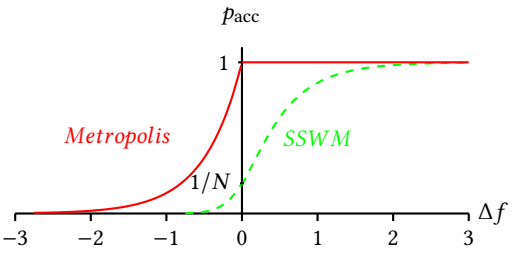

Figure 1: Acceptance probability for Metropolis (red solid line) and SSWM (green dashed line).

3 A COMMON STATIONARY DISTRIBUTION

We first show that SSWM and Metropolis have the same stationary distribution, starting by briefly recapping the foundations of Markov chain theory and mixing times (see, e. g. [1, 6, 8]). A Markov chain is called *irreducible* if every state can be reached from every other state. It is called *periodic* if certain states can only be visited at certain times; otherwise the chain is *aperiodic*. Markov chains that are both irreducible and aperiodic are called *ergodic* and they converge to a unique stationary distribution π.

Theorem 3.1. *Consider SSWM and Metropolis with local mutations over a Markov chain with states $x \in \{0,1\}^n$ and a fitness function $f : \{0,1\}^n \to \mathbb{R}$. Then the stationary distribution of such process will be*

$$\pi(x) = \frac{e^{\gamma f(x)}}{Z}$$

where $Z = \sum_{x \in \{0,1\}^n} e^{\gamma f(x)}$ and $\gamma = 2(N-1)\beta$ in the case of SSWM and $\gamma = \alpha$ for Metropolis.

Proof. First note that the acceptance probability of Metropolis has the following property $p_{\text{acc}}(\Delta f)/p_{\text{acc}}(-\Delta f) = e^{\gamma \Delta f}$, this relation is also true for SSWM with $\gamma = 2\beta(N-1)$ (Lemma 2 in [13]). The stationary condition for a distribution $\pi(x)$ can be written as (cf. Proposition 1.19 in [8])

$$\pi(x) \cdot p(x \to y) = \pi(y) \cdot p(y \to x), \quad \text{for all } x, y \in \{0,1\}^n$$

where $p(x \to y)$ is the probability of moving to state y given that the current state is x. Therefore

$$\pi(x) \cdot p(x \to y) = \frac{e^{\gamma f(x)}}{Z} \cdot \frac{1}{n} \cdot p_{\text{acc}}(f(y) - f(x))$$

$$= \frac{e^{\gamma f(x)}}{Z} \cdot \frac{1}{n} \cdot \frac{p_{\text{acc}}(f(y) - f(x))}{p_{\text{acc}}(f(x) - f(y))} \cdot p_{\text{acc}}(f(x) - f(y)),$$

since $p_{\text{acc}}(\Delta f)/p_{\text{acc}}(-\Delta f) = e^{\gamma \Delta f}$ we obtain

$$\begin{aligned}\pi(x) \cdot p(x \to y) &= \frac{e^{\gamma f(x)}}{Z} \cdot \frac{1}{n} \cdot e^{\gamma(f(y)-f(x))} \cdot p_{\text{acc}}(f(x) - f(y)) \\ &= \frac{e^{\gamma f(y)}}{Z} \cdot \frac{1}{n} \cdot p_{\text{acc}}(f(x) - f(y)) \\ &= \pi(y) \cdot p(y \to x).\end{aligned}$$

□

The distance between the current distribution and the stationary distribution is measured as follows by the *total variation distance*. For two distributions μ and ν on a state space Ω it is defined as

$$||\mu - \nu|| = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)| = \max_{A \subseteq \Omega} |\mu(A) - \nu(A)|.$$

Now the mixing time is defined as the first point in time where the total variation distance decreases below $1/(2e)$ (the constant $1/(2e)$ being a somewhat arbitrary choice in [18]).

Definition 3.2 (Mixing time [18]). Consider an ergodic Markov chain starting in x with stationary distribution π. Let $p_x^{(t)}$ denote the distribution of the Markov chain after t steps. Let $t_x(\varepsilon)$ be the time until the total variation distance between the current distribution and the stationary distribution has decreased to ε: $t_x(\varepsilon) = \min\{t : ||p_x^{(t)} - \pi|| \leq \varepsilon\}$. Let $t(\varepsilon) := \max_{x \in \Omega} t_x(\varepsilon)$ be the worst-case time until this happens.

The mixing time t_{mix} of the Markov chain is then defined as $t_{\text{mix}} := t(1/(2e))$.

After the mixing time, both algorithms will be close to the stationary distribution, hence any differing behaviour can only be shown before the mixing time. In the following, we aim to construct problems where the mixing time is large, such that SSWM and Metropolis show different performance over a long period of time. In particular, we seek to identify a problem where the expected first hitting time of SSWM is less than the mixing time.

4 A 3 STATE MODEL

We first introduce a fitness function defined on 2 bits. We will analyse the behaviour of SSWM and Metropolis on this function, before proceeding (in Section 5.1) to concatenate n copies of the fitness function to create a new function where SSWM drastically outperforms Metropolis.

The idea is simple: we start in a search point of low fitness, and are faced with two improving moves, one with a higher fitness than the other. This construction requires 3 search points, encoded on 2 bits; the 4th possible bitstring will have a fitness of $-\infty$, making it inaccessible for both Metropolis and SSWM.

Considering the 3 relevant nodes of the Markov Chain, they form a valley structure tunable through two parameters a and b representing the fitness difference between the minimum and the local and global optimum respectively. This model is inspired by the Muller Dobzshansky incompatibilities [12] in population genetics.

Definition 4.1 (3 State Model). For any $b > a > 0$ and a bit-pair $\{0,1\}^2$ the 3 state model $f_3^{a,b}$ assigns fitness as follows:

$$\begin{aligned} f_3^{a,b}(01) &= a, && \text{(state 1)} \\ f_3^{a,b}(00) &= 0, && \text{(state 2)} \\ f_3^{a,b}(10) &= b, && \text{(state 3)} \end{aligned}$$

and $f_3^{a,b}(11) = -\infty$.

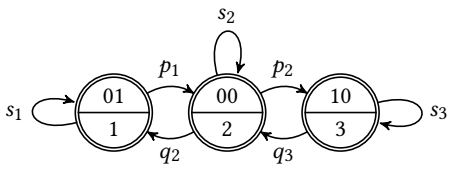

Figure 2: Accessible states of the 3 State Model by SSWM and Metropolis.

The main idea behind this construction is that Metropolis is indifferent to the choice of the local optimum (fitness $a > 0$) and the global optimum (fitness $b > a$), hence it will make either choice from state 00 with probability 1/2. SSWM, on the other hand, when parameterised accordingly, may reject a small improvement of fitness a more often than it would reject a larger improvement of $b > a$. Hence we expect SSWM to reach the global optimum with a probability larger than 1/2 in just a relevant step (an iteration excluding self-loops). We make this rigorous in the following.

Since the analysis has similarities with the classical Gambler's Ruin problem (see e.g. [3]) we introduce similar concepts to the ruin probability and the expected duration of the game.

Definition 4.2 (Notation). Consider a Markov Chain with only local probabilities

$$P(X_{t+1} = i \mid X_t = j) = \begin{cases} q_i & \text{if } j = i - 1 \\ s_i = 1 - q_i - p_i & \text{if } j = i \\ p_i & \text{if } j = i + 1 \\ 0 & \text{if } j \notin \{i-1, i, i+1\}. \end{cases}$$

Then, we define absorbing probabilities ρ_i as the probability of hitting state k before state 1 starting from i. Equivalently, we define expected absorbing times $\mathrm{E}(T_{k \vee 1} \mid i)$ as the expected hitting time for either state 1 or k starting from i.

Note that this definition may differ from the standard use of *absorbing* within Markovian processes. In our case the state k has an absorbing probability, but the state itself is not absorbing since the process may move to other states. The following lemma derives a closed form for the just defined absorbing probability, both for the general scheme 1 and for two specific algorithms. The obtained expression of $\rho_2 = p_2/(p_2 + q_2)$ is simply the conditional probability of moving to the global optimum p_2 given that the process has moved, hence the factor $1 - s_2$ in the denominator.

Theorem 4.3. *Consider any trajectory based algorithm that fits in Algorithm 1 on $f_3^{a,b}$ starting from state 2. Then the absorbing probability of state 3 is*

$$\rho_2 = \frac{p_2}{p_2 + q_2}.$$

And for Metropolis and SSWM ($N \geq 2$) it is

$$\rho_2^{\text{MA}} = \frac{1}{2} \qquad \rho_2^{\text{SSWM}} = \frac{p_{\text{fix}}(b)}{p_{\text{fix}}(b) + p_{\text{fix}}(a)} > \frac{1}{2}.$$

Proof. Let us start expressing the absorbing probability with a recurrence relation: $\rho_2 = p_2\rho_3 + q_2\rho_1 + (1 - p_2 - q_2)\rho_2$. Using the boundary conditions $\rho_3 = 1$ and $\rho_1 = 0$ we can solve the previous equation yielding $\rho_2 = p_2/(p_2 + q_2)$.

The result for Metropolis follows from introducing $p_2 = q_2$ since both probabilities lead to a fitness improvement. For SSWM the mutational component of p_2 and q_2 cancels out, yielding only the acceptance probabilities. Finally the lower bound of $1/2$ is due to state 3 having a fitness $b > a$. □

Note that SSWM's ability to reject improvements resembles a strategy of *steepest ascent* [16]: since the probability of accepting a large improvement is larger than the probability of accepting a small improvement, SSWM tends to favour the largest uphill gradient. Metropolis, on the other hand, follows the first slope it finds, resembling a *first (or greedy) ascent* strategy.

However, despite these different behaviours, we know from Theorem 3.1 that both algorithms will eventually reach the same state. This seems surprising in the light of Theorem 4.3 where the probabilities of reaching the local versus global optimum from the minimum are potentially very different.

This seeming contradiction can be explained by the fact that Metropolis is able to undo bad decisions by leaving the local optimum and going back to the starting point. Furthermore, leaving the local optimum has a much higher probability than leaving the global optimum. In the light of the previous discussion, Metropolis' strategy in local optima resembles that of a *shallowest descent*: it tends to favour the smallest downhill gradient. This allows Metropolis to also converge to the stationary distribution by leaving local optimal states.

We show that the mixing time is asymptotically equal to the probability of accepting a move leaving the local optimum, state 1. Note that asymptotic notation is used with respect to said probability, as the problem size is fixed to 2 bits. To be able to bound the mixing time using Theorem 1.1 in [2], we consider *lazy* versions of SSWM and Metropolis: algorithms that with probability $1/2$ execute a step of SSWM or MA, respectively, and otherwise produce an idle step. This behaviour can also be achieved for the original algorithms by appending two irrelevant bits to the encoding of $f_3^{a,b}$.

Another assumption is that the algorithm parameters are chosen such that $\pi(3) \geq 1/2$. This is a natural assumption as state 3 has the highest fitness, and it is only violated in case the temperature is extremely high.

Theorem 4.4. *The mixing time of lazy SSWM and lazy Metropolis on $f_3^{a,b}$ is $\Theta(1/p_{\text{acc}}(-a))$, provided $b > a > 0$ are chosen such that $\pi(3) \geq 1/2$.*

Proof. We use the transition probabilities from Figure **??**. According to Theorem 1.1 in [2], if $\pi(3) \geq 1/2$ then the mixing time of the lazy algorithms is of order $\Theta(t)$ where

$$t = \frac{1}{p_1} + \frac{\pi(1) + \pi(2)}{\pi(2)p_2}$$

As $p_1 = 1/2 \cdot p_{\text{acc}}(-a)$ this proves a lower bound $\Omega(1/p_{\text{acc}}(-a))$. For the upper bound, we bound t from above as follows, using $\pi(1)p_1 = \pi(2)q_2$ (the stationary distribution is reversible):

$$\begin{aligned} t &= \frac{1}{p_1} + \frac{\pi(1) + \pi(2)}{\pi(2)p_2} \\ &= \frac{1}{p_1} + \frac{\pi(1)}{\pi(2)p_2} + \frac{1}{p_2} \\ &= \frac{1}{p_1} + \frac{q_2}{p_2} \cdot \frac{1}{p_1} + \frac{1}{p_2} \leq \frac{3}{p_1} \end{aligned}$$

as $q_2/p_2 = p_{\text{acc}}(a)/p_{\text{acc}}(b) \leq 1$ and $p_2 \geq p_1$. Recalling that $p_1 = 1/2 \cdot p_{\text{acc}}(-a)$ completes the proof. □

4.1 Experiments

We performed experiments to see the analysed dynamics more clearly. In the case of SSWM we considered different population sizes $N = (10, 100)$ and scaling parameter values $\beta = (0.01, 0.1)$. For Metropolis we choose a temperature of $1/\alpha$, such that $\alpha = 2(N-1)\beta$. This choice was made according to Theorem 3.1 such that both algorithms have the same stationary distribution. The algorithms are run for 10000 iterations. The fitness values for states representing local and global optimum are chosen as $a = 1$ and $b = 10$ respectively. We record the average and standard deviations of the number of components in the local and global optimum for 50 runs.

The experimental results show that in general SSWM outperforms Metropolis in considered settings (Figure 3 (left)). However, this effect decreases with the capability of Metropolis to accept negative improvements. For example as seen in Figure 3 (right) the two algorithms are similar in performance when the temperature is high for Metropolis.

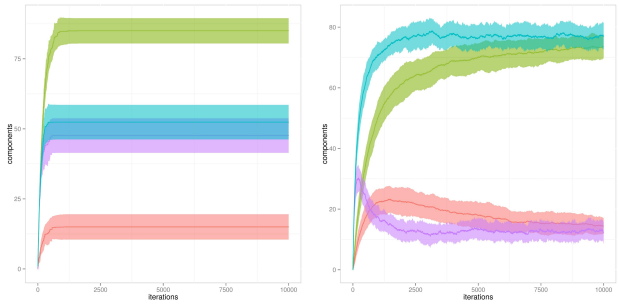

Figure 3: Performance of SSWM with $N = 100$ and $\beta = 0.1$ (left) and $N = 10$ and $\beta = 0.01$ (right) on the 3 State Model. For Metropolis the temperature was chosen such that $\alpha = 2(N-1)\beta$ in both cases. The average number of components (± one standard deviation) in the global and local optimum are plotted for SSWM and for Metropolis with colours red, green, purple and cyan respectively.

This coincides with our theoretical observation that the mixing time is inversely proportional to $p_{\text{acc}}(-a)$, which in turn depends on a and the parameters of SSWM and Metropolis. If the temperature is low (large α), the algorithms show a different behaviour before the mixing time, whereas if the temperature is high (small α), the algorithms quickly reach the same stationary distribution.

5 A 5 STATE MODEL

We saw in the previous section how two algorithms with different selection operators displayed the same limit behaviour. Moreover the mixing time was small for both algorithms despite the asymmetric valley structure of the function. This asymmetry favoured moving towards the steepest slope, landscape feature from which SSWM benefits and Metropolis is indifferent. However this feature also implied that it was easier climbing down from the shallowest slope, and Metropolis successfully exploits this feature to recover from wrong decisions.

Making use of this results we build a new function where the previous local optimum will now be a transition point between the valley and the new local optimum. We will assign an extremely large fitness to this new search point, this way we *lock in* bad decisions made by any of the two algorithms. In the same way, if the algorithm moved to the previous global optimum we offer a new search point with the highest fitness.

Definition 5.1 (5 State Model). For any $M' > M \gg b > a > 0$ and a search point $x \in \{0,1\}^3$ the 5 state model $f_5^{M,a,b,M'}$ assigns fitness as follows

$$\begin{aligned} f_5^{M,a,b,M'}(011) &= M, && \text{(state 1)}\\ f_5^{M,a,b,M'}(001) &= a, && \text{(state 2)}\\ f_5^{M,a,b,M'}(000) &= 0, && \text{(state 3)}\\ f_5^{M,a,b,M'}(100) &= b, && \text{(state 4)}\\ f_5^{M,a,b,M'}(110) &= M' && \text{(state 5)} \end{aligned}$$

and $f_5^{M,a,b,M'}(010) = f_5^{M,a,b,M'}(101) = f_5^{M,a,b,M'}(111) = -\infty$.

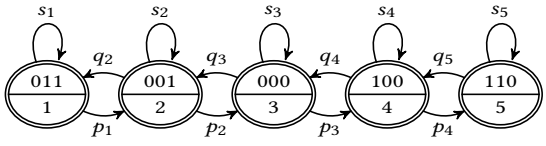

Figure 4: Accessible states of the 5 State Model by SSWM and Metropolis.

Let us consider the Markov chain with respect to the above model. For simplicity we refer to states with the numbers 1-5 as in the above description.

Again, we will compute the absorbing probability for the global optimum (state 5 or 110 of the Markov Chain). Note that by choosing very large values of M and M', we can make the mixing time arbitrarily large, as then the expected time to leave state 1 or state 5 becomes very large, and so does the mixing time.

For simplicity we introduce the following conditional transition probabilities Q_i and P_i for each state i as

$$P_i := \frac{p_i}{p_i + q_i} \qquad Q_i := \frac{q_i}{p_i + q_i}. \tag{3}$$

By using this notation the following lemma derives a neat expression for the absorption probability $\rho_3 = P_3P_4/(Q_2Q_3 + P_3P_4)$. This formula can be understood in terms of events that can occur in 2 iterations starting from state 3. Since Q and P are conditioning on the absence of self-loops there will be only 4 events after 2 iterations, whose probabilities will be $\{Q_3Q_2, Q_3P_2, P_2Q_4, P_3P_4\}$. Therefore the expression $\rho_3 = P_3P_4/(Q_2Q_3 + P_3P_4)$ is just the success probability over the probability space.

Lemma 5.2. *Consider any trajectory based algorithm that fits in Algorithm 1 on $f_5^{M,a,b,M'}$ starting from the node 3. Then the absorbing probability for state 5 is*

$$\rho_3 = \frac{P_3P_4}{Q_2Q_3 + P_3P_4}.$$

Proof. Firstly we compute the absorbing probabilities,

$$\begin{aligned} \rho_1 &= 0\\ \rho_2 &= p_2\rho_3 + q_2\rho_1 + (1 - p_2 - q_2)\rho_2\\ \rho_3 &= p_3\rho_4 + q_3\rho_2 + (1 - p_3 - q_3)\rho_3\\ \rho_4 &= p_4\rho_5 + q_4\rho_3 + (1 - p_4 - q_4)\rho_4\\ \rho_5 &= 1 \end{aligned}$$

which can be rewritten using P_i and Q_i from equation (3) and the two boundary conditions as

$$\begin{aligned} \rho_2 &= P_2\rho_3\\ \rho_3 &= P_3\rho_4 + Q_3\rho_2\\ \rho_4 &= P_4 + Q_4\rho_3. \end{aligned}$$

Solving the previous system for ρ_3 yields $\rho_3 = P_3 \cdot (P_4 + Q_4\rho_3) + Q_3P_2\rho_3$ which after solving for ρ_3 leads to

$$\rho_3 = \frac{P_3P_4}{1 - Q_3P_2 - P_3Q_4}$$

introducing $Q_3 = 1 - P_3$, $P_2 = 1 - Q_2$ and $Q_4 = 1 - P_4$ in the denominator yields the claimed statement. □

Now we apply the previous general result for the two studied heuristics. First, for Metropolis one would expect the absorbing probability to be $1/2$ since it does not distinguish between improving moves of different magnitudes. However it comes at as a surprise that this probability will always be greater than $1/2$. The reason is again due to the fitness dependant acceptance probability of detrimental moves.

Theorem 5.3. *Consider MA starting from state 3 on $f_5^{M,a,b,M'}$, then the absorbing probability for state 5 is*

$$\rho_3^{\mathrm{MA}} = \frac{1 + e^{-\alpha a}}{2 + e^{-\alpha a} + e^{-\alpha b}} \geq \frac{1}{2}.$$

Proof. First let us compute the two conditional probabilities

$$Q_2 = \frac{1}{1 + e^{-\alpha a}}, \qquad P_4 = \frac{1}{1 + e^{-\alpha b}}.$$

Now we invoke Lemma 5.2 but with $P_3 = Q_3 = 1/2$ since Metropolis does not distinguish slope gradients, hence

$$\begin{aligned} \rho_3 &= \frac{P_4}{Q_2 + P_4}\\ &= \frac{1/\left(1 + e^{-\alpha b}\right)}{1/(1 + e^{-\alpha a}) + 1/\left(1 + e^{-\alpha b}\right)}\\ &= \frac{1 + e^{-\alpha a}}{2 + e^{-\alpha a} + e^{-\alpha b}}. \end{aligned}$$

Finally, using $\Delta f_3^2 \leq \Delta f_3^4$, it follows that $\rho_3^{\mathrm{MA}} \geq 1/2$. □

Finally, for SSWM we were able to reduce the complexity of the absorbing probability to just the two intermediate points (states 2 and 4) between the valley (state 3) and the two optima (states 1 and 5). The obtained expression is reminiscent of the absorbing probability on the 3 State Model (Theorem 4.3). However, it is important to note that a and b were the fitness of the optima in $f_3^{a,b}$ and now they refer to the transition nodes between the valley and the optima.

Theorem 5.4. *Consider SSWM ($N \geq 2$) starting from state 3 on $f_5^{M,a,b,M'}$, then the absorbing probability of state 5 is*

$$\rho_3^{\text{SSWM}} \geq \frac{p_{\text{fix}}(b)}{p_{\text{fix}}(b) + p_{\text{fix}}(a)} > \frac{1}{2}.$$

Proof. Let us start computing the probabilities required by Lemma 5.2.

$$P_4 = \frac{1}{1 + p_{\text{fix}}(-b)/p_{\text{fix}}(M' - b)} \qquad Q_2 = \frac{1}{1 + p_{\text{fix}}(-a)/p_{\text{fix}}(M - a)}$$

$$P_3 = \frac{1}{1 + p_{\text{fix}}(a)/p_{\text{fix}}(b)} \qquad Q_3 = \frac{1}{1 + p_{\text{fix}}(b)/p_{\text{fix}}(a)}$$

Let us now focus on the term $Q_2Q_3/(P_3P4)$:

$$\frac{Q_2Q_3}{P_3P_4} = \frac{\left(1 + \frac{p_{\text{fix}}(-b)}{p_{\text{fix}}(M'-b)}\right)}{\left(1 + \frac{p_{\text{fix}}(-a)}{p_{\text{fix}}(M-a)}\right)} \cdot \frac{\left(1 + \frac{p_{\text{fix}}(a)}{p_{\text{fix}}(b)}\right)}{\left(1 + \frac{p_{\text{fix}}(b)}{p_{\text{fix}}(a)}\right)}$$

the last term is of the form $(1 + x)/(1 + 1/x) = x$, hence it can be highly simplified to just $p_{\text{fix}}(a)/p_{\text{fix}}(b)$, yielding

$$\frac{Q_2Q_3}{P_3P_4} = \frac{\left(1 + \frac{p_{\text{fix}}(-b)}{p_{\text{fix}}(M'-b)}\right)}{\left(1 + \frac{p_{\text{fix}}(-a)}{p_{\text{fix}}(M-a)}\right)} \cdot \frac{p_{\text{fix}}(a)}{p_{\text{fix}}(b)}$$

since $0 < p_{\text{fix}}(-b) < p_{\text{fix}}(-a) < p_{\text{fix}}(M - a) < p_{\text{fix}}(M' - b) < 1$, we can bound $p_{\text{fix}}(-b)/p_{\text{fix}}(M' - b) \leq p_{\text{fix}}(-a)/p_{\text{fix}}(M - a)$ to obtain

$$\frac{Q_2Q_3}{P_3P_4} \leq \frac{\left(1 + \frac{p_{\text{fix}}(-a)}{p_{\text{fix}}(M-a)}\right)}{\left(1 + \frac{p_{\text{fix}}(-a)}{p_{\text{fix}}(M-a)}\right)} \cdot \frac{p_{\text{fix}}(a)}{p_{\text{fix}}(b)} = \frac{p_{\text{fix}}(a)}{p_{\text{fix}}(b)}.$$

Introducing this in Lemma 5.2 leads to

$$\rho_3 = \frac{1}{1 + Q_2Q_3/(P_3P_4)} \geq \frac{1}{1 + p_{\text{fix}}(a)/p_{\text{fix}}(b)} = \frac{p_{\text{fix}}(b)}{p_{\text{fix}}(b) + p_{\text{fix}}(a)}.$$

Finally, using $b > a$ we obtain the lower bound of $1/2$. □

5.1 An Example Where SSWM Outperforms Metropolis

We now consider a smaller family of problems $f_5^{M,1,10,M'}$ and create an example where SSWM outperforms Metropolis. In this simpler yet general scenario we can compute the optimal temperature for Metropolis that will maximise the absorbing probability ρ_3^{MA}.

Lemma 5.5. *Consider Metropolis on $f_5^{M,1,10,M'}$ starting from state 3. Then for any parameter $\alpha \in \mathbb{R}^+$ the absorbing probability ρ_3^{MA} of state 5 can be bounded as*

$$\rho_3^{\text{MA}}(\alpha) \leq \rho_3^{\text{MA}}(\alpha^*) < 0.63$$

where $\alpha^* = 0.312\ldots$ is the optimal value of α.

Proof. Introducing the problem settings ($a = 1$ and $b = 10$) in the absorbing probability from Theorem 5.3 yields

$$\rho_3^{\text{MA}}(\alpha) = \frac{1 + e^{-\alpha}}{2 + e^{-\alpha} + e^{-10\alpha}}$$

whose derivative is

$$\frac{d\rho_3^{\text{MA}}(\alpha)}{d\alpha} = \frac{e^{9\alpha}(10e^{\alpha} - e^{10\alpha} + 9)}{\left(e^{9\alpha} + 2e^{10\alpha} + 1\right)^2}.$$

By solving numerically this equation for $d(\rho_3^{\text{MA}}(\alpha))/d\alpha = 0$ with $\alpha > 0$ we obtain an optimal value of $\alpha^* = 0.312071\ldots$ which yields the maximum value of $\rho_3^{\text{MA}}(\alpha^*) = 0.623881\ldots$

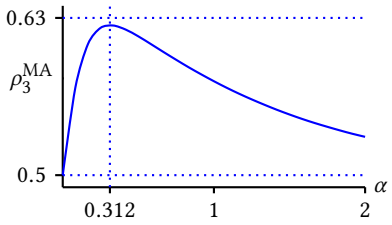

Figure 5: Absorbing probability of Metropolis on the 5-state model. □

Now that we have shown the optimal parameter for Metropolis, we will find parameters such that SSWM outperforms Metropolis. To obtain this we must make use of SSWM's ability of rejecting improvements. We wish to identify a parameter setting such that small improvements ($\Delta f = a = 1$) are accepted with small probabilities, while large improvements ($\Delta f = b = 10$) are accepted with a considerably higher probability. The following graph shows p_{fix} for different values of β. While for large β, $p_{\text{fix}}(1)$ and $p_{\text{fix}}(10)$ are similar, for smaller values of β there is a significant difference. Furthermore we can see that $p_{\text{fix}}(1) \leq 1/2$ i.e. the algorithm will prefer to stay in the current point, rather than moving to the local optimum.

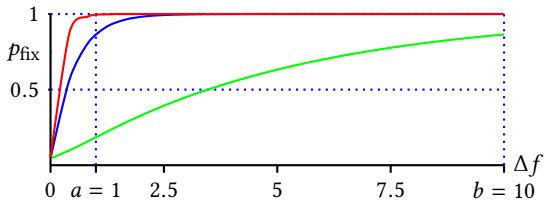

Figure 6: Acceptance probability of SSWM with $N = 20$ and $\beta = (0.2, 2, 5)$ for the (green, blue, red) curves.

In the following lemma we identify a range of parameters for which the desired effect occurs. The results hold for arbitrary population size, apart from the limit case $N = 1$ where SSWM becomes a pure random walk. The scaling factor β is the crucial parameter; only small values up to 0.33 will give a better performance than Metropolis.

Lemma 5.6. *Consider SSWM on $f_5^{M,1,10,M'}$ starting from state 3. Then for $\beta \in (0, 0.33]$ and $N \geq 2$ the absorbing probability ρ_3^{SSWM} of state 5 is at least 0.64.*

Proof. Using the bound on ρ_3^{SSWM} from Theorem 5.4 with $a = 1$ and $b = 10$ we obtain

$$\rho_3^{\text{SSWM}} \geq \frac{p_{\text{fix}}(10)}{p_{\text{fix}}(1) + p_{\text{fix}}(10)} = \frac{1}{1 + p_{\text{fix}}(1)/p_{\text{fix}}(10)}$$

We want to show that $\rho_3^{\text{SSWM}} \geq 0.64$, which is equivalent to $p_{\text{fix}}(1)/p_{\text{fix}}(10) \leq 1/0.64 - 1 = 9/16$. Using the bounds $p_{\text{fix}}(1) \leq 2\beta/(1-e^{-2N\beta})$ and $p_{\text{fix}}(10) \geq 20\beta/(1+20\beta)$ (see Lemma 1 from [13]) we obtain

$$\begin{aligned}\frac{p_{\text{fix}}(1)}{p_{\text{fix}}(10)} &\leq \frac{2\beta}{1-e^{-2N\beta}} \cdot \frac{1+20\beta}{20\beta} \\ &= \frac{1+20\beta}{10\left(1-e^{-2N\beta}\right)} \\ &\leq \frac{1+20\beta}{10\left(1-e^{-4\beta}\right)}\end{aligned}$$

where in the last step we have used $N \geq 2$. The obtained expression is always increasing with $\beta > 0$, hence we just need to find the value β^* for when it crosses our threshold value of $9/16$. Solving this numerically we found that $\beta^* = 0.332423\ldots$ then the statement will be true for β values up to this cut off point. □

Now that we have derived parameter values for which SSWM has a higher absorbing probability on the 5 State Model than Metropolis for any temperature setting $1/\alpha$ (Lemma 5.5), we are ready to construct a function where SSWM considerably outperforms Metropolis. We first define a concatenated function

$$f(X) = \sum_{i=1}^{n} f_5^{M,a,b,M'}(x_i)$$

consisting of n copies of the 5 State Model (i.e. n components) x_i with $1 \leq i \leq n$, such that the concatenated function $f(x)$ returns the sum of the fitnesses of the individual components. Note that $3n$ bits are used in total. To ensure that the algorithms take long expected times to escape from each local optimum we set $M = n$ and $M' = 2n$ for each component x_i, apart from keeping $a = 1$ and $b = 10$, for which the absorbing probabilities from Lemmata 5.5 and 5.6 hold. Furthermore, we assume $2\beta(N-1) = \Omega(1)$ to ensure that SSWM remains in states 1 or 5 for a long time.

Theorem 5.7. *The expected time for SSWM and Metropolis to reach either the local or global optimum of all the components of $f(x)$ is $O(n \log n)$. With overwhelming probability $1 - e^{-\Omega(n)}$, SSWM with positive constant $\beta < 0.33$ and $N \geq 2$ has optimised correctly at least $(639/1000)n$ components while Metropolis with optimal parameter $\alpha = 0.312\ldots$ has optimised correctly at most $(631/1000)n$ components. The expected time for either algorithm to increase (or decrease) further the number of correctly optimised components by one is at least $e^{\Omega(n)}$.*

Proof. The expected time to reach either of the states 5 or 1 on the single-component 5 State Model is a constant c for both algorithms. Hence, the first statement follows from an application of the coupon collector where each coupon has to be collected c times [11]. The second statement follows by straightforward applications of Chernoff bounds using that each component is independent and, pessimistically, that SSWM optimises each one correctly with probability $640/1000$ (i.e., Lemma 5.6) and Metropolis with probability $630/1000$ (i.e., Lemma 5.5). The final statement follows because both algorithms with parameters $\Omega(1)$ accept a new solution, that is $\Omega(n)$ worse, only with exponentially small probability. □

As the absorbing probabilities of SSWM and Metropolis are both constants, with that of SSWM being higher than that of MA, we expect SSWM to achieve a higher fitness. We can amplify these potentially small differences by transforming our fitness function f with a step function $g(f(X))$ returning 1 if at least a certain number of components are optimised correctly (i.e. state 110 is found) and 0 otherwise:

$$g(f(X)) := \begin{cases} 0 & \text{if } f(X) \leq 1.635n^2 \\ 1 & \text{otherwise} \end{cases}$$

We use this to compose a function h where with overwhelming probability SSWM is efficient while Metropolis is not:

$$h(X) = f(X) \cdot (1 - g(f(X))) + 2nM' \cdot g(f(X))$$

Note that $h(X) = f(X)$ while the step function $g(X)$ returns 0, and h attains a global optimum if and only if $g(X) = 1$. Our analysis transfers to the former case.

Corollary 5.8. *In the setting described in Theorem 5.7, SSWM finds an optimum on $h(X)$ in expected time $O(n \log n)$, while Metropolis requires $e^{\Omega(n)}$ steps with overwhelming probability.*

Obviously, by swapping the values of M and M' in f, the function would change into one where preferring improvements of higher fitness is deceiving. As a result, SSWM would, with overwhelming probability, optimise at least 63.9% of the components incorrectly. Although Metropolis would optimise more components correctly than SSWM, it would still be inefficient on h.

5.2 Experiments

We performed experiments to study the performance of SSWM and Metropolis on the 5 State Model under several parameter settings. The experimental setting is similar to that of the 3 State Model. For all the considered scenarios SSWM had at least 70 of the components in global optimum while Metropolis had 50 on average. Results for two sample parameter settings are shown in Figure 7.

We also plot the step function $g(f(X))$ as this is the most crucial term in $h(X)$. The respective plots for $g(f(X))$ function suggest that SSWM outperforms Metropolis on the 5 State Model (see Figure 8). For SSWM it has value 1 for both parameter settings at most after 4000 iterations while Metropolis has 0 throughout the considered time span of 5000 iterations.

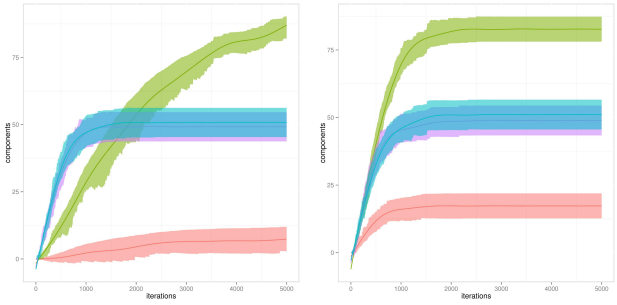

Figure 7: Performance of SSWM with $N = 100$ and $\beta = 0.1$ (left) and $N = 10$ and $\beta = 0.01$ (right) on the 5-state model. For Metropolis the temperature was chosen such that $\alpha = 2(N-1)\beta$ in both cases. The average number of components ($\pm$ one standard deviation) in the global and local optimum are plotted for SSWM and for Metropolis with colours red, green, purple and cyan respectively.

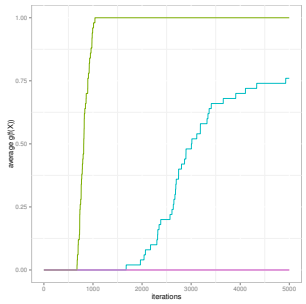

Figure 8: SSWM with $N = 10$ and $\beta = 0.01$ (in blue) and $N = 100$ and $\beta = 0.1$ (in green) and Metropolis with $\alpha = 2(N-1)\beta$ (purple) on $g(f(X))$.

6 CONCLUSIONS AND FUTURE WORK

We have presented a rigorous comparison of the non-elitist SSWM and Metropolis algorithms. Their main difference is that SSWM may reject improving solutions while Metropolis always accepts them. Nevertheless, we prove that both algorithms have the same stationary distribution, and they may only have considerably different performance on optimisation functions where the mixing time is large.

Our analysis on a 3 State Model highlights that a simple function with a local optimum of low fitness and a global optimum of high fitness does not allow the required large mixing times. The reason is that, although Metropolis initially chooses the local optimum more often than SSWM, it still escapes quickly. As a result we designed a 5 State Model which "locks" the algorithms to their initial choices. By amplifying the function to contain several copies of the 5 State Model we achieve our goal of defining a step function where SSWM is efficient while Metropolis requires exponential time with overwhelming probability, independent from its temperature parameter.

Given the similarities between SSWM and other particularly selective strategies such as steepest ascent, in future work we will analyse when these algorithms have different behaviours. This also relates to previous work where the choice of the pivot rule was investigated in local search and memetic algorithms [4, 17, 19].

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) and from the EPSRC under grant agreement no EP/M004252/1.

REFERENCES

[1] D. Aldous and J. Fill. 2017. *Reversible Markov Chains and Random Walks on Graphs.* Monograph in preparation, https://www.stat.berkeley.edu/~aldous/RWG/book.html.
[2] Guan-Yu Chen and Laurent Saloff-Coste. 2013. On the mixing time and spectral gap for birth and death chains. *Latin American Journal of Probability and Mathematical Statistics* X (2013), 293–321.
[3] William Feller. 1968. *An introduction to probability theory and its applications.* Wiley.
[4] Christian Gießen. 2013. Hybridizing Evolutionary Algorithms with Opportunistic Local Search. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO '13).* ACM, 797–804.
[5] John H. Gillespie. 1984. Molecular Evolution Over the Mutational Landscape. *Evolution* 38, 5 (1984), 1116–1129.
[6] Mark Jerrum and Alistair Sinclair. 1996. The Markov chain Monte Carlo method: an approach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems.* PWS Publishing, 482–520.
[7] Motoo Kimura. 1962. On the Probability of Fixation of Mutant Genes in a Population. *Genetics* 47, 6 (1962), 713–719.
[8] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. 2008. *Markov Chains and Mixing Times.* American Mathematical Society.
[9] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092.
[10] Pietro S. Oliveto, Tiago Paixão, Jorge Pérez Heredia, Dirk Sudholt, and Barbora Trubenová. 2016. When Non-Elitism Outperforms Elitism for Crossing Fitness Valleys. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16).* ACM, New York, NY, USA, 1163–1170.
[11] Pietro S. Oliveto and Xin Yao. 2011. Runtime Analysis of Evolutionary Algorithms for Discrete Optimization. In *Theory of Randomized Search Heuristics: Foundations and Recent Developments.* World Scientific Publishing Co., Inc.
[12] H. A. Orr. 1995. The population genetics of speciation: the evolution of hybrid incompatibilities. *Genetics* 139 (1995), 1805–1813. Issue 4.
[13] Tiago Paixão, Jorge Pérez Heredia, Dirk Sudholt, and Barbora Trubenová. 2016. Towards a Runtime Comparison of Natural and Artificial Evolution. *Algorithmica* (2016).
[14] Jorge Pérez Heredia, Barbora Trubenová, Dirk Sudholt, and Tiago Paixão. 2016. Selection Limits to Adaptive Walks on Correlated Landscapes. *Genetics* (2016).
[15] Guy Sella and Aaron E Hirsh. 2005. The application of statistical physics to evolutionary biology. *Proceedings of the National Academy of Sciences of the United States of America* 102, 27 (2005), 9541–9546.
[16] J. E. Smith. 2007. Coevolving Memetic Algorithms: A Review and Progress Report. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37, 1 (2007), 6–17.
[17] Dirk Sudholt. 2011. Hybridizing Evolutionary Algorithms with Variable-Depth Search to Overcome Local Optima. *Algorithmica* 59, 3 (2011), 343–368.
[18] Dirk Sudholt. 2011. Using Markov-Chain Mixing Time Estimates for the Analysis of Ant Colony Optimization. In *Proceedings of the 11th Workshop on Foundations of Genetic Algorithms (FOGA 2011).* ACM Press, 139–150.
[19] Kuai Wei and Michael J. Dinneen. 2014. Runtime Analysis to Compare Best-improvement and First-improvement in Memetic Algorithms. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO '14).* ACM, 1439–1446.