Proceedings Paper:
Sudholt, D. orcid.org/0000-0001-6020-1646 (2018) On the Robustness of Evolutionary Algorithms to Noise: Refined Results and an Example Where Noise Helps. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2018). Genetic and Evolutionary Computation Conference (GECCO 2018), 15-19 Jul 2018, Kyoto, Japan. ACM . ISBN 978-1-4503-5618-3

https://doi.org/10.1145/3205455.3205595

On the Robustness of Evolutionary Algorithms to Noise: Refined Results and an Example Where Noise Helps

Dirk Sudholt
Department of Computer Science, University of Sheffield
Sheffield, United Kingdom

ABSTRACT

We present refined results for the expected optimisation time of the (1+1) EA and the (1+λ) EA on LeadingOnes in the prior noise model, where in each fitness evaluation the search point is altered before evaluation with probability p. Previous work showed that the (1+1) EA runs in polynomial time if $p = O((\log n)/n^2)$ and needs superpolynomial time if $p = \Omega((\log n)/n)$, leaving a huge gap for which no results were known. We close this gap by showing that the expected optimisation time is $\Theta(n^2) \cdot \exp(\Theta(pn^2))$, allowing for the first time to locate the threshold between polynomial and superpolynomial expected times at $p = \Theta((\log n)/n^2)$. Hence the (1+1) EA on LeadingOnes is much more sensitive to noise than previously thought. We also show that offspring populations of size $\lambda \geq 3.42 \log n$ can effectively deal with much higher noise than known before.

Finally, we present an example of a rugged landscape where prior noise can help to escape from local optima by blurring the landscape and allowing a hill climber to see the underlying gradient.

KEYWORDS

Evolutionary algorithms, noisy optimisation, runtime analysis, theory.

1 INTRODUCTION

Many real-world problems suffer from sources of uncertainty, such as noise in the fitness evaluation, changing constraints, or dynamic changes to the fitness function [18]. Evolutionary algorithms are well suited for dealing with these challenges, and have proven to work well in many applications to combinatorial problems [5].

However, our theoretical understanding of how evolutionary algorithms deal with noise is limited. It is often not clear how noise affects the performance of evolutionary algorithms, and how much noise an evolutionary algorithm can cope with. For evolution strategies in continuous optimisation there exists a rich body of work (see, e. g. [4, 17, 20] and the references therein), but there are only few rigorous theoretical analyses on the performance of noisy evolutionary optimisation in discrete spaces.

The first runtime analysis for discrete evolutionary algorithms in a noisy setting was given by Droste [10]. He considered a setting now known as *one-bit prior noise*, where with probability p a uniform random bit is flipped before evaluation. Hence, instead of returning the fitness of the evaluated search point, the fitness function may return the fitness of a random Hamming neighbour. He proved that, when $p = O((\log n)/n)$ the (1+1) EA can still optimise OneMax efficiently. But when $p = \omega((\log n)/n)$ the expected optimisation time becomes superpolynomial.

Gießen and Kötzing [15] studied a more general class of algorithms, including the (1+1) EA, (1+λ) EA, and (μ+1) EA on prior noise and *posterior noise*, where posterior noise means that noise is added to the fitness value. They presented an elegant approach that gives results in both noise models. They showed that the (1+1) EA on OneMax runs in expected time $O(n \log n)$ if $p = O(1/n)$, polynomial time if $p = O((\log n)/n)$, and superpolynomial time if $p = \omega((\log n)/n) \cap 1 - \omega((\log n)/n)$. The same results hold in the *bit-wise noise* model, where each bit is flipped independently before evaluation with probability p/n. For LeadingOnes they show a time bound of $O(n^2)$ if $p \leq 1/(6en^2)$ and an exponential lower bound if $p = 1/2$.

The authors also found that using parent populations in a (μ+1) EA can drastically improve robustness as survival selection removes one of the worst individuals, and a population increases the chances that a low-fitness individual will be correctly identified as having low fitness. Offspring populations also increase robustness as they amplify the probability that a clone of the current search point will be evaluated truthfully, thus lowering the chance of losing the best fitness. For LeadingOnes they showed a time bound for the (1+λ) EA of $O(\lambda n + n^2)$ if $p \leq 0.028/n$ and $72 \log n \leq \lambda = o(n)$.

Dang and Lehre [6] gave general results for prior and posterior noise in non-elitist EAs. The same authors [7] also considered noise resulting from only partially evaluating search points.

In terms of posterior noise, Sudholt and Thyssen [33] considered the performance of a simple ant colony optimiser (ACO) for computing shortest paths when path lengths are obscured by positive posterior noise modelling traffic delays. They showed that noise can make the ants risk-seeking, tricking them onto a suboptimal path and leading to exponential optimisation times. Doerr, Hota, and Kötzing [8] showed that this problem can be avoided if the parent is reevaluated in each iteration. Feldmann and Kötzing [12] further analysed the performance of fitness-proportional updates. Friedrich, Kötzing, Krejca, and Sutton [14] showed that the compact Genetic Algorithm and ACO [13] are both efficient under extreme Gaussian posterior noise, while a simple (μ+1) EA is not.

Prugel-Bennett, Rowe, and Shapiro [25] considered a population-based algorithm using only selection and crossover, and showed that the algorithm can optimise OneMax with a large amount of noise. Qian, Yu, and Zhou [27] showed that noise can be handled efficiently by combining reevaluation and threshold selection. Akimoto, Astete-Morales, and Teytaud [1] as well as Qian, Yu, Tang, Jin, Yao, and Zhou [26] showed that resampling can essentially eliminate the effect of noise.

Qian, Bian, Jiang, and Tang [28] studied the performance of the (1+1) EA on OneMax and LeadingOnes for a more general prior noise model with parameters (p, q): with probability p the search point is altered by flipping each bit with probability q. They studied two special cases: $(p, 1/n)$ meaning that with probability p a standard bit mutation is performed before evaluation and $(1, q)$, which

Table 1: Overview of results on the expected optimisation time on LeadingOnes with prior noise. Results in this work also hold for the general model $(p', q/n)$ with $p = p'q$ and $q \le 1$, which has not been studied before in this generality. Results for the (1+1) EA also hold for asymmetric one-bit noise p.

Setting	Gießen and Kötzing [15]	Qian, Bian, Jiang, and Tang [28]	This work
(1+1) EA, one-bit noise p	$O(n^2)$ if $p \le 1/(6en^2)$ $2^{\Omega(n)}$ if $p = 1/2$	polynomial if $p = O((\log n)/n^2)$ superpolynomial if $p = \omega((\log n)/n) \cap o(1)$ exponential if $p = \Omega(1)$	
(1+1) EA, bit-wise noise $(p, 1/n)$		polynomial if $p = O((\log n)/n^2)$ superpolynomial if $p = \omega((\log n)/n) \cap o(1)$ exponential if $p = \Omega(1)$	$O\big(n^2 \cdot e^{O(pn^2)}\big)$ if $p \le 1 - \Omega(1)$ $\Omega\big(n^2 \cdot e^{\Omega(pn^2)}\big)$ if $p = O(1/n)$ (applies to all three (1+1) EA rows)
(1+1) EA, bit-wise noise $(1, p/n)$		polynomial if $p = O((\log n)/n^2)$ superpolynomial if $p = \omega((\log n)/n) \cap o(1)$ exponential if $p = \Omega(1)$	
(1+λ) EA, one-bit noise p	$O(\lambda n + n^2)$ if $p \le 0.028/n$ and $72 \log n \le \lambda = o(n)$		$O\big(n^2 \cdot e^{O(pn/\lambda)}\big)$ if $p \le 1/2$ and $3.42 \log n \le \lambda = O(n)$

is bit-wise noise with parameter q. For LeadingOnes they improve results from [15], showing that the (1+1) EA runs in polynomial expected time if $p = O((\log n)/n^2)$ and that it runs in superpolynomial time if $p = \omega((\log n)/n)$. This holds for one-bit noise with probability p, the $(p, 1/n)$ model and bit-wise noise with probability p/n (see Table 1). For bit-wise noise $(1, q)$ with parameter $q = \Omega(1/n)$ the expected time is exponential.

In this work we improve previous results for prior noise on the function $\text{LeadingOnes}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$, counting the number of leading ones in the bit string. This function is of particular interest as it represents a problem where decisions have to be made in sequence in order to reach the optimum, building up the components of a global optimum step by step. In the case of LeadingOnes, this is a prefix of ones that is being built up. Problems with similar features are found in combinatorial optimisation, for instances as worst-case examples for finding shortest paths [32].

Disruptive mutations can destroy a partial solution, leading to a large fitness loss, such that the algorithm is thrown back and may need a long time to recover. As such, LeadingOnes is a prime example of a problem that is very susceptible to noise.

We provide upper and lower bounds on the expected optimisation time of the (1+1) EA on LeadingOnes, showing that the expected time is in $\Theta(n^2) \cdot \exp(\Theta(pn^2))$, which is tight up to constant factors in the exponent of the term $\exp(\Theta(pn^2))$ that reflects the slowdown resulting from noise. This shows that the time is $\Theta(n^2)$ if $p = O(1/n^2)$, polynomial if $p = O((\log n)/n^2)$, and superpolynomial if $p = \omega((\log n)/n^2)$. This improves previous negative results that only showed superpolynomial times for $p = \omega((\log n)/n)$, which is by factor of n larger.

The upper bound (Section 3) is based on a very simple argument: estimating the probability that no noise will occur during a period of time long enough to allow the algorithm to find an optimum without experiencing any noise. The lower bound (Section 4) follows arguments from Rowe and Sudholt [31] who analysed the performance of the non-elitist algorithm (1,λ) EA on LeadingOnes.

In Section 5 we show an improved upper bound for the (1+λ) EA on LeadingOnes. Finally, in Section 6 we show that on the class of Hurdle problems [24], a class of rugged functions with many local optima on an underlying slope, noise helps to overcome local optima, allowing a simple hill climber to succeed that would otherwise fail with overwhelming probability.

2 PRELIMINARIES

Algorithm 1 shows the (1+λ) EA in the context of prior noise, which includes the (1+1) EA as a special case of $\lambda = 1$. Here noise(x) denotes a noisy version of a search point x, according to the given noise model. We assume that all applications of noise are independent. The (1+λ) EA creates λ independent offspring, evaluates their noisy fitness, and then picks a best offspring. This offspring is then compared against the parent, whose noisy fitness is evaluated in each generation. This means in particular that an offspring can replace a parent whose real fitness is higher if the parent is misevaluated to a lower noisy fitness, the offspring is misevaluated to a higher noisy fitness, or both.

Algorithm 1: (1+λ) EA with prior noise

Choose x uniformly at random.
while *termination criterion not met* **do**
  for $i = 1, \dots, \lambda$ **do**
    Create y_i by copying x and flipping each bit independently with probability $1/n$.
    Evaluate $f_i := f(\text{noise}(y_i))$.
  Choose $z \in P_t$ uniformly at random from $\arg\max\{f_1, \dots, f_\lambda\}$.
  if $f_z \ge f(\text{noise}(x))$ **then** $x = z$;

The optimisation time is defined as the number of fitness evaluations until a global optimum is found for the first time. We consider the following prior noise models from previous work; asymmetric noise is inspired by an asymmetric mutation operator [16].

One-bit noise(p) [10, 15]: with probability $1 - p$, noise(x) = x and otherwise noise(x) = x' where in x', compared to x, one bit chosen uniformly at random was flipped.

Bit-wise noise(p, q) [28]: with probability $1 - p$, noise(x) = x and otherwise noise(x) = x' where in x', compared to x, each bit was flipped independently with probability q.

Asymmetric one-bit noise(p) **[27]:** with probability $1-p$, $\text{noise}(x) = x$ and otherwise $\text{noise}(x) = x'$ where in x', compared to x, if $x \notin \{0^n, 1^n\}$, with probability $1/2$ a uniform random 0-bit is flipped, with probability $1/2$ a uniform random 1-bit is flipped, and if $x \in \{0^n, 1^n\}$ a uniform random bit is flipped.

We often write $(p, q/n)$ for bit-wise noise instead of (p, q) as then q plays a similar role to p in one-bit prior noise p, which allows for a more unified presentation of results.

Note that $\Pr(\text{noise}(x) \neq x) = p$ for one-bit noise and asymmetric one-bit noise, and for the bit-wise noise model $(p, q/n)$, $\Pr(\text{noise}(x) \neq x) = p(1-(1-q/n)^n) \leq pq$ by Bernoulli's inequality.

3 A SIMPLE AND GENERAL UPPER BOUND FOR DEALING WITH UNCERTAINTY

We first present a very simple result that applies in a general setting of optimisation under uncertainty (noise/dynamic changes/etc.). It is formulated for iterative algorithms that maintain a single search point, called *trajectory-based algorithms*, however it is easy to extend the definition to population-based algorithms as well.

Our approach is based on the worst-case median optimisation time, defined as follows.

Definition 3.1. For any trajectory-based algorithm $\mathcal{A}$ optimising a fitness function f let $T_{\mathcal{A},f}(x)$ be the random first hitting time of a global optimum when starting in x. We assume hereinafter that each initial search point x leads to a finite expectation.

We define the worst-case expected optimisation time $E_{\mathcal{A},f}$ as

$$E_{\mathcal{A},f} := \max_x E(T_{\mathcal{A},f}(x))$$

Further define the median optimisation time $M_{\mathcal{A},f}$

$$M_{\mathcal{A},f}(x) := \min\{t \mid \Pr(T_{\mathcal{A},f}(x) \leq t) \geq 1/2\}$$

and the worst-case median optimisation time

$$M_{\mathcal{A},f} := \max_x M_{\mathcal{A},f}(x).$$

We omit subscripts if the context is clear. Applying Markov's inequality for all x, the median worst-case optimisation time is not much larger than the expected worst-case optimisation time.

Theorem 3.2. *For every $\mathcal{A}$ and every f, $M_{\mathcal{A},f} \leq 2E_{\mathcal{A},f}$.*

The following theorem gives an upper bound on the worst-case expected optimisation time under uncertainty, assuming we do know (an upper bound on) the median worst-case optimisation time in a setting without uncertainty.

Theorem 3.3. *Consider a setting where in each iteration a failure event may occur independently with probability $0 \leq p < 1$. Consider any function f on which an iterative algorithm $\mathcal{A}$ has worst-case median optimisation time M if $p = 0$. Then the worst-case expected optimisation time of $\mathcal{A}$ with failure probability p is at most*

$$2M(1-p)^{-M} \leq 2M \cdot e^{pM/(1-p)}.$$

The statement also holds if p is an upper bound on the probability of a failure and/or M is an upper bound on the described time.

Proof. By definition of the median worst-case optimisation time, if the algorithm experiences M steps without a failure, it will find an optimum with probability at least $1/2$ regardless of the initial search point. The probability that in a phase of M steps there will be no failure is at least $(1-p)^M$. Hence the expected waiting time for a phase of M steps without failures where the algorithm finds an optimum is at most $2M(1-p)^{-M}$ for every initial search point.

The inequality follows from $\frac{1}{1-p} = 1 + \frac{p}{1-p} \leq e^{p/(1-p)}$. □

In the setting of prior noise, Theorem 3.3 implies the following.

Theorem 3.4. *Consider an iterative algorithm $\mathcal{A}$ that evaluates up to ν search points in each iteration. For every function f on which $\mathcal{A}$ has worst-case median optimisation time M without prior noise, its worst-case expected optimisation time is at most*

$$2M(1-\nu p)^{-M} \leq 2M \cdot e^{\nu pM/(1-\nu p)}$$

for each of the following settings:

(1) one-bit prior noise with probability p,
(2) bit-wise prior noise $(p', q/n)$ with $q \leq 1$ and $p := p'q$, and
(3) asymmetric one-bit prior noise with probability p.

Proof. This follows immediately from Theorem 3.3 using the occurrence of noise as a failure event and a union bound over ν search points evaluated in each generation. □

For LeadingOnes Theorem 3.4 implies the following.

Theorem 3.5. *The expected optimisation time of the (1+1) EA with prior noise probability $p \leq 1 - \Omega(1)$ for any of the settings from Theorem 3.4, on LeadingOnes is*

$$O\big(n^2 \cdot e^{O(pn^2)}\big).$$

This is polynomial if $p = O((\log n)/n^2)$ and $O(n^2)$ if $p = O(1/n^2)$.

Proof. Follows from Theorem 3.4 with $\nu = 2$ (as the (1+1) EA evaluates parent and offspring in each generation), $2p/(1-2p) = O(p)$, and the fact that the worst-case expected optimisation time of the (1+1) EA on LeadingOnes is $O(n^2)$ [11], hence by Theorem 3.2 the worst-case median optimisation time is $M = O(n^2)$. □

Despite the simplicity of the above proofs, Theorem 3.5 matches, unifies and generalises the best known results [28, Theorems 4.1, 4.4, and 4.7] which only state that the expected optimisation time on LeadingOnes is polynomial if the noise parameter is $O((\log n)/n^2)$ in the models $(p, 1/n)$, $(1, q/n)$ and one-bit noise (see Table 1).

4 A MATCHING LOWER BOUND FOR THE (1+1) EA ON LEADINGONES

The arguments from Section 3 pessimistically assume that, once noise occurs, the algorithm needs to restart from scratch. For LeadingOnes, and problems with a similar structure, this is not far from the truth. An unlucky mutation can destroy a long prefix of leading ones and the fitness of the current search point can decrease significantly. We will see that then the algorithm comes close to having to start from scratch. Such an effect was already observed and made rigorous in the analysis of island models with migration [19], separable functions [9], and for the (1,λ) EA on LeadingOnes [31]; parts of this section closely follow [31, Proof of Theorem 12].

The main result of this section is the following.

Theorem 4.1. *For each of the settings described in Theorem 3.4 the expected optimisation time of the (1+1) EA on LeadingOnes is* $\Omega\big(n^2 \cdot e^{\Omega(pn^2)}\big)$ *if* $p = O(1/n)$ *and* $e^{\Omega(n)}$ *if* $p = \omega(1/n)$ *and* $p \leq 1 - \Omega(1)$. *This is superpolynomial for* $p = \omega((\log n)/n^2)$.

Along with Theorem 3.5, the case $p = O(1/n)$ gives a bound of $\Theta(n^2) \cdot \exp(\Theta(pn^2))$. The result is tight up constants in exponent of the term $\exp(\Theta(pn^2))$ that reflects the impact of noise.

Theorem 4.1 improves on the best known results, summarised in Table 1. Note that there is a gap of order $1/n$ between the noise parameter regime $p = \omega((\log n)/n)$ where times are known to be superpolynomial and the noise parameter regime $p = O((\log n)/n^2)$ that led to polynomial upper bounds in [28] and in Theorem 3.5.

Theorem 4.1 closes this gap by showing that superpolynomial times already occur for noise parameters $p = \omega((\log n)/n^2)$. This shows that the (1+1) EA on LeadingOnes is highly sensitive to noise, especially since the corresponding threshold for OneMax is at $p = \Theta((\log n)/n)$ [10, 15]. Theorem 4.1 also unifies and generalises the above results by giving a bound that holds for the whole range of noise parameters p, and for different prior noise models. In order to prove Theorem 4.1, we first analyse the probability of the fitness dropping significantly.

Lemma 4.2. *Consider the setting of Theorem 4.1 with a current LeadingOnes value of* i. *Then the probability that the LeadingOnes value decreases below* $i/2$ *in one generation is* $\Omega(p(1-p)i^2/n^2)$. *This is* $\Omega(p)$ *if* $i = \Omega(n)$ *and* $p \leq 1 - \Omega(1)$.

Proof. Consider the following events. E_1: the offspring flips exactly one of the first $i/2$ bits, which has probability $\Omega(i/n)$. E_2: the parent is evaluated with prior noise flipping exactly one of the first $i/2$ bits, which happens with probability at least

(1) $i/2 \cdot p/n = \Omega(pi/n)$ for one-bit prior noise,
(2) $i/2 \cdot p' \cdot q/n \cdot (1-q/n)^{n-1} \geq p'qi/(2en) = pi/(2en) = \Omega(pi/n)$ for bit-wise prior noise $(p', q/n)$ with $p = p'q$, and
(3) $i/2 \cdot p/(2n) = \Omega(pi/n)$ for asymmetric one-bit noise, as with probability $p/2$, one of at most n 1-bits is flipped.

E_3: conditional on E_1 and E_2, the position of the bit flipped in the offspring is no smaller than the position of the flipped bit in the parent's noise. This has probability at least $1/2$ due to symmetry. E_4: the offspring is evaluated correctly (probability at least $1-p$).

If all these events happen, the offspring will appear to be no worse than the parent. Hence the offspring will survive, and its LeadingOnes value is at most $i/2$. Since all events are independent (or conditionally independent in the case of E_3), multiplying these probabilities implies the claim. □

As argued in [31] for the (1,λ) EA, such a fallback is not too detrimental per se as the (1+1) EA might recover from this easily. If the bits between $i/2$ and i have not been flipped during the mutation creating the accepted offspring, the previous leading ones can be easily recovered, in the best case by simply flipping the first 0-bit in the current search point. However, while waiting for such a mutation to happen, all bits between $i/2+1$ and i do not contribute to the fitness. So over time these bits are subjected to random mutations, which are likely to destroy many of the former leading ones. In other words, after a fallback previous leading ones are forgotten quickly.

The last fact was formalised in [19, Lemma 3] stated below. The lemma states that the probability distribution of a bit subjected to random mutations rapidly approaches a uniform distribution.

Lemma 4.3 (Adapted from Lässig and Sudholt [19]). *Let* $x^0, x^1, \ldots, x^t$ *be a sequence of random bit values such that* x^{j+1} *results from* x^j *by flipping the bit* x^j *independently with probability* $1/n$. *Then for every* $t \in \mathbb{N}$

$$\Pr(x^t = 1) \leq \frac{1}{2}\left(1 + \left(1 - \frac{2}{n}\right)^t\right).$$

We now say that the (1+1) EA *falls back* if, starting from a fitness at least $f^* := 2n/3$, the algorithm drops below a fitness of $n/2$. We speak of a *lasting fallback* if, additionally, the fitness remains below $n/2$ for at least $t_{\text{mix}} := n/2$ generations in which the offspring is accepted. Additionally, the initial search point is deemed a lasting fallback if its fitness is at most $n/2$.

The following lemma estimates probabilities for fallbacks and lasting fallbacks.

Lemma 4.4. *If* $p \leq 1 - \Omega(1)$ *and the current fitness is at least* f^*, *the probability of one generation yielding a fallback is* $\Omega(p)$. *Additionally, the probability of a fallback becoming a lasting fallback is* $\Omega(1)$.

Proof. The first statement follows from Lemma 4.2 as halving the current fitness results in a search point of fitness at most $n/2$.

A fallback becomes a lasting fallback if for t_{mix} generations after the fallback where the offspring is accepted, the first 0-bit never flips. Note that the offspring is accepted if and only if no leading ones are flipped, which is independent from the decision on the first 0-bit. The probability for the mentioned event is $(1-1/n)^{n/2} = \Omega(1)$. □

After a lasting fallback has occurred, the (1+1) EA with overwhelming probability needs some time in order to recover. Specifically, at least cn^2 generations are needed to increase the best fitness since the latest lasting fallback by at least $n/6$.

Lemma 4.5. *Let* t^* *be the latest generation where a fallback became a lasting fallback or* $t^* = 0$ *if no lasting fallback occurred. Let* B_t *be the best fitness found since generation* t^*. *With probability* $1 - e^{-\Omega(n)}$, *for a small constant* $c > 0$, $B_{t+cn^2} < B_t + n/6$.

Proof. A lasting fallback implies that at any generation from t^*, all bits at positions $\{B_t + 1, \ldots, n\}$ have been subjected to mutation at least $t_{\text{mix}} = n/2$ times. Every mutation flips each of these bits independently with probability $1/n$, leaving the bits in a random state. We apply the *principle of deferred decisions* [21, page 9] and determine the current bit value for these bits at the time these bits first have a chance to become part of the leading ones in an offspring. By Lemma 4.3 we know that then the probability such a bit is set to 1 is at most

$$\frac{1}{2}\left(1 + \left(1 - \frac{2}{n}\right)^{n/2}\right) \leq \frac{1}{2}\left(1 + \frac{1}{e}\right) = \frac{e+1}{2e}.$$

A necessary condition for increasing the best fitness by at least $n/6$ in cn^2 generations, c a positive constant chosen later, is that either

(1) among cn^2 mutations at least $2cn$ mutations lead to an improvement in fitness or

(2) during at most $2cn$ fitness improvements the total fitness gain is at least $n/6$.

The probability that a mutation leads to a fitness improvement is always at most $1/n$ as the first 0-bit needs to be flipped. By standard Chernoff bounds, the probability for the first event is at most $e^{-\Omega(n)}$. The total fitness gain is given by the number of improvements—at most $2cn$—plus a sum of up to $2cn$ geometric random variables to account for additional bits gained ("free riders"). By Theorem 5 in [3], we get that the probability of a fitness gain of $n/6$ is $e^{-\Omega(n)}$, provided that c is small enough. □

Lemma 4.6. *Let $c > 0$ be any constant. Within cn^2 generations where the current fitness is larger than f^*, a lasting fallback occurs with probability at least $1 - e^{-\Omega(pn^2)}$.*

Proof. The probability of a fallback occurring is $\Omega(p)$, and then it becomes lasting with probability $\Omega(1)$. Note that the time until a fallback potentially becomes a lasting fallback (whether it does or not) is not counted towards the cn^2 generations from the statement as during this time the fitness is smaller than f^*.

So the probability that no lasting fallback occurs is at most

$$(1 - \Omega(p))^{cn^2} \le e^{-\Omega(pn^2)}. \quad \square$$

Now we prove Theorem 4.1.

Proof of Theorem 4.1. With probability $1 - 2^{-\Omega(n)}$ the initial search point has fitness less than $n/2$, so the (1+1) EA starts with a lasting fallback. As the fitness after initialisation and after every lasting fallback is at most $n/2$, by Lemma 4.5, reaching a fitness of at least f^* from there takes time at least cn^2 with overwhelming probability, for a suitably small constant $c > 0$. Applying Lemma 4.5 every time the fitness increases to at least f^*, the (1+1) EA does not find an optimum within the next cn^2 generations where the fitness is at least f^*, with overwhelming probability. But by Lemma 4.6 during these cn^2 generations another lasting fallback occurs, with overwhelming probability. We iterate this argument until a failure occurs. The largest failure probability is $e^{-\Omega(pn^2)}$ if $p = O(1/n)$, hence in expectation we can iterate this argument at least $e^{\Omega(pn^2)}$ times, each iteration taking time at least cn^2 (from the time it takes to reach fitness f^* after a lasting fallback). If $p = \omega(1/n)$, the largest failure probability is $e^{-\Omega(n)}$ and in expectation we can iterate this argument for $e^{\Omega(n)}$ generations. Together, this proves the claim. □

5 IMPROVED RESULTS FOR OFFSPRING POPULATIONS

The general Theorem 3.3 can also be used in the context of offspring populations in the (1+λ) EA, in order to quantify the robustness of evolutionary algorithms with offspring populations to noise. Offspring populations can reduce the probability of the current fitness decreasing. This can happen in two different ways:

(1) the current search point may be misevaluated as having a poor fitness, and then be replaced by an offspring that is worse than the parent in real fitness or
(2) the current search point may be replaced by an offspring where mutation has led to poor real fitness, but noise happens to misevaluate the offspring as having a high fitness, thus replacing its parent. Here noise essentially needs to make the same bit-flips as the preceding mutation to cover up the effect of mutation.

The first failure can be avoided if there is a clone of the current search point where no prior noise has occurred. A large offspring population can amplify this probability.

Lemma 5.1. *Consider the (1+λ) EA in a prior noise model where $\Pr(\text{noise}(y) \neq y) \le p$ for all search points y. Then for all current search points x the probability that all copies of x among parent and offspring are affected by noise is at most*

$$p\left(1 - \left(1 - \frac{1}{n}\right)^n (1-p)\right)^\lambda = p\left(\frac{e - (1-p)}{e}\right)^\lambda \cdot \exp(O(\lambda/n)).$$

Proof. Let $q := (1 - 1/n)^n$ abbreviate the probability of creating a clone of the parent for an offspring. The probability of creating exactly i clones is $\binom{\lambda}{i} q^i (1-q)^{\lambda - i}$, and then the probability that all $i+1$ copies of x (including the parent) are affected by noise is p^{i+1}. Hence the sought probability is

$$\begin{aligned}\sum_{i=0}^{\lambda} \binom{\lambda}{i} q^i (1-q)^{\lambda-i} p^{i+1} &= p \sum_{i=0}^{\lambda} \binom{\lambda}{i} (pq)^i (1-q)^{\lambda-i} \\ &= p(1 - q + pq)^\lambda \\ &= p(1 - q(1-p))^\lambda\end{aligned}$$

where we have used the binomial theorem in the penultimate equality. Plugging in $(1-1/n)^n$ for q yields the claimed result. The second bound follows from $(1-1/n)^n = (1-1/n)(1-1/n)^{n-1} \ge (1-1/n) \cdot 1/e$ and straightforward calculations turning the $1 - 1/n$ term into a $\exp(O(\lambda/n))$ factor. Details are omitted due to space restrictions. □

Theorem 5.2. *Consider any of the settings from Theorem 3.4, except for asymmetric bit-wise noise*[1]*. The expected number of function evaluations for the (1+λ) EA with prior noise parameter $p \le 1/2$ on LeadingOnes with $\log_{\frac{e}{e-1/2}}(n) \le \lambda = O(n)$ is*

$$O\left(n^2 \cdot e^{O(pn/\lambda)}\right).$$

This is polynomial if $p = O((\lambda \log n)/n)$ and $O(n^2)$ if $p = O(\lambda/n)$.

The exponent is smaller compared to the upper bound for the (1+1) EA by a factor of order λn, and thus the threshold for p for which polynomial times are guaranteed increases by the same factor. The threshold between polynomial and superpolynomial times could be higher as we do not have a corresponding lower bound.

Theorem 5.2 improves and generalises the best known result for the (1+λ) EA [15, Corollary 24] which requires $p = O(1/n)$ and $\lambda \ge 72 \log n$ and gives a time bound of $O(\lambda n + n^2)$. This is $O(n^2)$ as the authors also assume $\lambda = o(n)$. Our result covers the whole parameter range for p up to $1/2$ and also identifies a functional relationship between p and λ that guarantees robustness to noise.

Proof of Theorem 5.2. We estimate the probability of the following failure events in order to apply a union bound later on.

[1]We exclude asymmetric bit-wise noise as the probability of flipping a 1-bit may be $\omega(1/n)$ in case there are $o(n)$ leading ones, and only $o(n)$ 1-bits in total. We cannot exclude that this happens, though it seems highly unlikely in the light of Lemma 4.3.

Failure event E_1: all copies of the current search point are affected by noise. By Lemma 5.1, this probability is at most

$$p_1 := O\left(p\left(\frac{e-(1-p)}{e}\right)^{\lambda}\right) \leq O\left(p\left(\frac{e-1/2}{e}\right)^{\lambda}\right) = O\left(\frac{p}{n}\right).$$

Failure event E_2: the best offspring is evaluated as having the parent's fitness, and the offspring y chosen to replace the parent carries disruptive mutations that were undone by noise, i. e. $\textsc{LeadingOnes}(y) < \textsc{LeadingOnes}(\text{noise}(y)) = \textsc{LeadingOnes}(x)$. The probability for this to happen is at most

$$p_2 := \frac{p}{n}$$

as noise has to flip at least one specific bit.

Failure event E_3: there is an offspring y that carries disruptive mutations, but is being evaluated as being better than the parent, i. e. $\textsc{LeadingOnes}(y) < \textsc{LeadingOnes}(x)$ and $\textsc{LeadingOnes}(\text{noise}(y)) > \textsc{LeadingOnes}(x)$. For each offspring where mutation flips one of the leading ones, two events may occur: if mutation flips the first 0-bit, noise in an offspring has to undo all mutations of the leading ones. This has probability at most p/n^2. Otherwise, noise has to undo all mutations of the leading ones and flip the first 0-bit at the same time. This is impossible under one-bit noise, and has probability at most p/n^2 under bit-wise noise. Along with a union bound over these two events and λ offspring,

$$p_3 \leq \frac{2p\lambda}{n^2} = O\left(\frac{p}{n}\right).$$

As long as no failure occurs, the current fitness of the (1+λ) EA cannot decrease. We now show that, conditional on no failure occurring, the expected worst-case number of generations of the (1+λ) EA is bounded by $O(n + n^2/\lambda) = O(n^2/\lambda)$.

The probability of one offspring increasing the current fitness is at least $(1-p)/(en)$ as it suffices to flip the first 0-bit and not to flip any of the other bits, and to have the offspring being evaluated correctly. The probability that this happens in at least one of the λ offspring and the parent is evaluated correctly is at least

$$(1-p)\left(1-\left(1-\frac{1-p}{en}\right)^{\lambda}\right) \geq \frac{(1-p)^2\lambda/(en)}{1+(1-p)\lambda/(en)} = \Omega\left(\frac{\lambda}{n}\right)$$

where the inequality follows from [2, Lemma 6]. The expected time to increase the best fitness is thus $O(n/\lambda)$, and since the fitness only has to be increased at most n times, an upper bound of $O(n^2/\lambda)$ generations follows, for every initial search point. The same bound also holds for the worst-case median optimisation time by Theorem 3.2.

Now the result follows from applying Theorem 3.3 with a time bound of $O(n^2/\lambda)$ and a failure probability bound of $p_1 + p_2 + p_3 = O(p/n)$, and multiplying the number of generations by λ for the number of function evaluations. □

6 AN EXAMPLE WHERE NOISE HELPS

The final contribution of this paper is to show that noise can be beneficial for escaping from local optima. To this end, we consider a known class of functions that lead to a highly rugged fitness landscape with an underlying gradient pointing towards the location of the global optimum. Such landscapes are known as "big valley" structures, which is an important characteristic of many hard problems from combinatorial optimisation [23, 30].

Prügel-Bennett defined such a class of problems known as Hurdle problems [24] as an example function where genetic algorithms with crossover outperform hill climbers. Hurdle functions are functions of unitation, that is, they only depend on the number of 1-bits. The fitness is given as

$$\textsc{Hurdle}(x) = -\left\lceil \frac{|x|_0}{w} \right\rceil - \frac{|x|_0 \bmod w}{w}$$

where $|x|_0$ denotes the number of 0-bits in x and w is a parameter called *hurdle width* that defines the distance between subsequent peaks. A sketch of the function is shown in Figure 1.

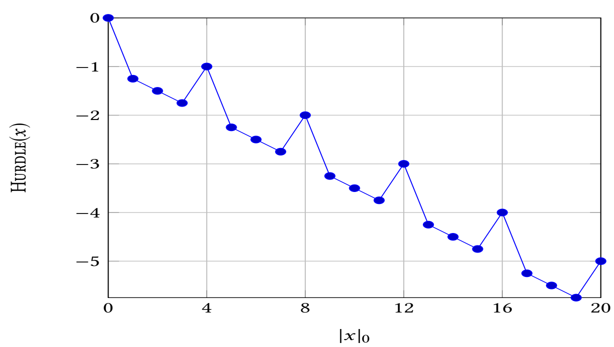

Figure 1: Sketch of a Hurdle function with hurdle width $w = 4$ and problem size $n = 20$.

Here all search points with $i \bmod w = 0$ zeros are local optima, and all search points with j zeros, $i - w < j < i$, have worse fitness. Hence an evolutionary algorithm needs to flip at least w bits in order to find a search point of better fitness. Nguyen and Sudholt [22] proved that the (1+1) EA has expected time $\Theta(n^w)$ if $2 \leq w \leq n/2$.

In the following, we consider the well-known algorithm *Randomised Local Search (RLS)*, which works like the (1+1) EA, but only flips exactly one bit in each mutation. It is obvious that RLS has infinite expected time on any Hurdle function with non-trivial hurdle width $w \geq 2$, and Nguyen and Sudholt [22] showed via Chernoff bounds that local searchers get stuck in a non-optimal local optimum with probability $1 - 2^{-\Omega(n)}$ if $w \leq (1 - \Omega(1))n/2$.

However, prior noise can help to escape from such a local optimum: RLS with one-bit prior noise can misevaluate either the parent or the offspring, which allows the algorithm to accept a search point with $i \bmod w = w - 1$ ones. Then it can climb to the next local optimum from there, until the global optimum is found. This is made precise in the following theorem.

Theorem 6.1. *The expected optimisation time of RLS with one-bit prior noise $p \leq 1/(6n)$ on Hurdle with hurdle width $w \geq 2\log(n)$ is $O(n^2/(pw^2) + n\log n)$.*

Note that in particular for $p = \Theta(1/n)$ and $w = \Omega(n/\sqrt{\log n})$ this is $O(n \log n)$. Then RLS is as efficient as on the underlying function OneMax without any hurdles.

Proof of Theorem 6.1. The algorithm can escape from a local optimum with i zeros, $i \bmod w = 0$, if the offspring has $i-1$ zeros (probability i/n) and additionally

(1) the offspring is misevaluated as having i zeros (probability $p \cdot (n-i+1)/n$) or
(2) the parent is misevaluated as having $i-1$ zeros (probability pi/n).

The probability of the union of these events is $p(n-i+1)/n + pi/n - p^2 i(n-i+1)/n^2 = p(1 + 1/n - pi(n-i+1)/n^2) \geq p(1-p)$ as the event of both offspring and parent being misevaluated as described is counted twice in the enumeration. Together, the probability of escaping from a local optimum with i zeros is at least $p(1-p)i/n$.

We now define a potential function g such that $g(i)$ estimates or overestimates the expected optimisation time from a state with i zeros, bar constant factors. Let $a_i := 2^{(i \bmod w)-w+1}$, then

$$g(i) := \begin{cases} 0 & \text{if } i = 0, \\ g(i-1) + \frac{n}{ip(1-p)} & \text{if } i > 0, i \bmod w = 0, \\ g(i-1) + \frac{n}{i} + a_i \frac{n^2}{i^2 p(1-p)^3} & \text{otherwise.} \end{cases}$$

Note that $g(i) \leq g(n)$, with $g(n)$ being composed of the following sums. The additive terms $\frac{n}{i}$ for all $i > 0, i \bmod w > 0$ sum up to at most $\sum_{i=1}^{n} \frac{n}{i} = O(n \log n)$. For each hurdle with a peak at i zeros, $g(n)$ contains an additive term $\frac{n}{ip(1-p)}$ as well as terms

$$\sum_{j=1}^{w-1} 2^{j-w+1} \frac{n^2}{(i-w+j)^2 p(1-p)^3} \leq O(1) \cdot \frac{n^2}{i^2 p(1-p)^3}$$

as $\sum_{d=0}^{i-1} 2^{-d} i^2/(i-d)^2 = O(1)$. Adding up the terms for each hurdle with $w, 2w, 3w, \dots, (n/w)w$ zeros yields

$$\begin{aligned} g(i) \leq g(n) &= O\left(n \log n + \sum_{j=1}^{n/w} \left(\frac{n}{jwp(1-p)} + \frac{n^2}{(jw)^2 p(1-p)^3}\right)\right) \\ &= O\left(n \log n + \frac{n}{wp(1-p)} \sum_{j=1}^{n/w} \frac{1}{j} + \frac{n^2}{w^2 p(1-p)^3} \sum_{j=1}^{n/w} \frac{1}{j^2}\right) \\ &= O\left(n \log n + \frac{n \log(n/w)}{wp} + \frac{n^2}{w^2 p}\right) \\ &= O\left(n \log n + \frac{n^2}{w^2 p}\right) \end{aligned}$$

where the penultimate line follows from $\sum_{j=1}^{n/w} 1/j^2 \leq \sum_{j=1}^{\infty} 1/j^2 = \pi^2/6 = O(1)$ and in the last line we used $\log(n/w) = O(n/w)$ to absorb the middle term. We show in the following that the potential decreases in expectation by $\Omega(1)$.

For $0 < i \bmod w < w-1$, the potential decreases by $g(i) - g(i-1)$ if mutation creates a search point with $i-1$ zeros and the mutant is evaluated correctly (probability at least $i/n \cdot (1-p)$). It is increased by $g(i+1) - g(i)$ only if mutation creates a search point with $i+1$ zeros (probability $(n-i)/n \leq 1$) and either the parent or the offspring is misevaluated (probability at most $2p$), as otherwise the offspring will be rejected. Thus for all i with $i \bmod w \notin \{0, w-1\}$, using $a_{i+1} = 2a_i$,

$$\begin{aligned} &E(g(X_t) - g(X_{t+1}) \mid X_t = i, i \bmod w \notin \{0, w-1\}) \\ &\geq \frac{i}{n}(1-p)(g(i) - g(i-1)) - 2p(g(i+1) - g(i)) \\ &= \frac{i}{n}(1-p)\left(\frac{n}{i} + a_i \frac{n^2}{i^2 p(1-p)^3}\right) - 2p\left(\frac{n}{i+1} + a_{i+1} \frac{n^2}{(i+1)^2 p(1-p)^3}\right) \\ &\geq 1 - p + (1-p)a_i \frac{n}{ip(1-p)^3} - 2p\left(\frac{n}{i} + 2a_i \frac{n^2}{i^2 p(1-p)^3}\right) \\ &= 1 - p - \frac{2pn}{i} + \frac{a_i n}{ip(1-p)^3}\left(1 - p - \frac{4pn}{i}\right). \end{aligned}$$

As $p \leq 1/(6n)$, the bracket is at least $1 - 1/(6n) - 2/3 \geq 0$, hence the drift is at least

$$\begin{aligned} &E(g(X_t) - g(X_{t+1}) \mid X_t = i, i \bmod w \notin \{0, w-1\}) \\ &\geq 1 - p - \frac{2pn}{i} \geq 1 - \frac{1}{6n} - \frac{1}{3} \geq \frac{1}{2}. \end{aligned}$$

For $i \bmod w = 0$, the potential is decreased by $g(i) - g(i-1) = \frac{n}{ip(1-p)}$ with probability at least $p(1-p)i/n$, and it is increased by $g(i+1) - g(i)$ only if either the parent or the offspring is misevaluated and the offspring increases the number of zeros. The probability of an increase is bounded by $2p$. Thus

$$\begin{aligned} &E(g(X_t) - g(X_{t+1}) \mid X_t = i, i \bmod w = 0) \\ &\geq \frac{n}{ip(1-p)} \cdot \frac{ip(1-p)}{n} - 2p(g(i+1) - g(i)) \\ &= 1 - 2p(g(i+1) - g(i)) \\ &= 1 - 2p \cdot \left(\frac{n}{i+1} + 2^{-w+2} \cdot \frac{n^2}{(i+1)^2 p(1-p)^3}\right) \\ &\geq 1 - 2pn - 2^{-w+3} \cdot \frac{n^2}{i^2(1-p)^3} \end{aligned}$$

and using $p \leq 1/(6n)$ and $w \geq 2 \log n$ this is at least

$$\geq \frac{2}{3} - \frac{8}{w^2(1-p)^3} \geq \frac{2}{3} - o(1).$$

For $i \bmod w = w-1$ the potential is decreased by $g(i) - g(i-1)$ if mutation decreases the number of zeros and both parent and offspring are evaluated truthfully. The potential is increased by $g(i+1) - g(i)$ only if mutation creates a search point with $i+1$ zeros (probability at most 1). Thus

$$\begin{aligned} &E(g(X_t) - g(X_{t+1}) \mid X_t = i, i \bmod w = w-1) \\ &\geq \frac{i(1-p)^2}{n} \cdot (g(i) - g(i-1)) - (g(i+1) - g(i)) \\ &= \frac{i(1-p)^2}{n} \cdot \left(\frac{n}{i} + \frac{n^2}{i^2 p(1-p)^3}\right) - \frac{n}{(i+1)p(1-p)} \\ &= (1-p)^2 + \frac{n}{ip(1-p)} - \frac{n}{(i+1)p(1-p)} \\ &\geq (1-p)^2 = 1 - O(1/n). \end{aligned}$$

Now standard additive drift arguments yield a $O(g(n))$ bound. □

The reason why prior noise is helpful is that, intuitively speaking, it can "smooth out" the fitness landscape, blurring rugged peaks and allowing the algorithm to see the underlying gradient. Hence noise can be useful for problems with a *big valley* structure. This effect has been observed in continuous spaces before [29] where it was termed "annealing of peaks". In discrete spaces the only other examples the author is aware of showing a positive effect of noise are deceptive functions and needle-in-a-haystack functions [27].

7 CONCLUSIONS

We have presented a simple method for proving upper bounds under several prior noise models, based on estimating the probability that during the median worst-case optimisation time no noise occurs. Despite its simplicity, it matches and generalises the best known results [28] and provides a unified approach for one-bit noise, bit-wise noise, and asymmetric bit-wise noise. Along with our negative result for LeadingOnes, the expected optimisation time of the (1+1) EA on LeadingOnes is $\Theta(n^2) \cdot \exp(\Theta(pn^2))$ for one-bit noise p, asymmetric one-bit noise p, and bit-wise noise $(p', q/n)$ where $q \leq 1$ and $p = p'q$. This confirms that the threshold between polynomial and superpolynomial times is $p = \Theta((\log n)/n^2)$.

Offspring populations can cope with noise up to $p \leq 1/2$ if the population size is at least $\lambda \geq \log_{\frac{e}{e-1/2}}(n) \approx 3.42 \log n$. We obtained an upper bound of $O\big(n^2 \cdot e^{O(pn/\lambda)}\big)$, guaranteeing polynomial expected times for $p = O((\lambda \log n)/n)$. An open problem is whether the upper bound is tight in the same sense as for the (1+1) EA.

Finally, we showed that on the Hurdle problem class, a highly rugged problem with a clear "big valley" structure, prior noise is helpful as it allows RLS to escape from local optima and to follow the underlying gradient.

ACKNOWLEDGMENTS

The author thanks the anonymous reviewers for their many thorough and constructive comments that helped to improve the paper.

REFERENCES

[1] Y. Akimoto, S. Astete-Morales, and O. Teytaud. Analysis of runtime of optimization algorithms for noisy functions over discrete codomains. *Theor. Comput. Sci.*, 605:42–50, 2015. doi: 10.1016/j.tcs.2015.04.008.

[2] G. Badkobeh, P. K. Lehre, and D. Sudholt. Black-box complexity of parallel search with distributed populations. In *Proc. of FOGA '15*, pages 3–15. ACM Press, 2015.

[3] S. Baswana, S. Biswas, B. Doerr, T. Friedrich, P. P. Kurur, and F. Neumann. Computing single source shortest paths using single-objective fitness functions. In *Proc. of FOGA '09*, pages 59–66. ACM Press, 2009.

[4] H.-G. Beyer. Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanics and Engineering*, 186(2):239 – 267, 2000. doi: https://doi.org/10.1016/S0045-7825(99)00386-2.

[5] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2): 239–287, 2009. doi: 10.1007/s11047-008-9098-4.

[6] D.-C. Dang and P. K. Lehre. Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In *Proc. of FOGA '15*, pages 62–68. ACM, 2015. doi: 10.1145/2725494.2725508.

[7] D.-C. Dang and P. K. Lehre. Runtime analysis of non-elitist populations: From classical optimisation to partial information. *Algorithmica*, 75(3):428–461, 2016. doi: 10.1007/s00453-015-0103-x.

[8] B. Doerr, A. Hota, and T. Kötzing. Ants easily solve stochastic shortest path problems. In *Proc. of GECCO '12*, pages 17–24, 2012. doi: 10.1145/2330163.2330167.

[9] B. Doerr, D. Sudholt, and C. Witt. When do evolutionary algorithms optimize separable functions in parallel? In *Proc. of FOGA 2013*, pages 51–64. ACM, 2013.

[10] S. Droste. Analysis of the (1+1) EA for a noisy OneMax. In *Proc. of GECCO 2004*, pages 1088–1099. Springer, 2004.

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

[12] M. Feldmann and T. Kötzing. Optimizing expected path lengths with ant colony optimization using fitness proportional update. In *Proc. of FOGA '13*, pages 65–74. ACM, 2013. doi: 10.1145/2460239.2460246.

[13] T. Friedrich, T. Kötzing, M. S. Krejca, and A. M. Sutton. Robustness of ant colony optimization to noise. In *Proc. of GECCO '15*, pages 17–24. ACM, 2015. doi: 10.1145/2739480.2754723.

[14] T. Friedrich, T. Kötzing, M. S. Krejca, and A. M. Sutton. The compact genetic algorithm is efficient under extreme gaussian noise. *IEEE Transactions on Evolutionary Computation*, 21(3):477–490, 2017. doi: 10.1109/TEVC.2016.2613739.

[15] C. Gießen and T. Kötzing. Robustness of populations in stochastic environments. *Algorithmica*, 75(3):462–489, 2016. doi: 10.1007/s00453-015-0072-0.

[16] T. Jansen and D. Sudholt. Analysis of an asymmetric mutation operator. *Evolutionary Computation*, 18(1):1–26, 2010.

[17] M. Jebalia, A. Auger, and N. Hansen. Log-linear convergence and divergence of the scale-invariant (1+1)-ES in noisy environments. *Algorithmica*, 59(3):425–460, 2011. doi: 10.1007/s00453-010-9403-3.

[18] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317, 2005. doi: 10.1109/TEVC.2005.846356.

[19] J. Lässig and D. Sudholt. Design and analysis of migration in parallel evolutionary algorithms. *Soft Computing*, 17(7):1121–1144, 2013.

[20] S. Meyer-Nieberg and H.-G. Beyer. Why noise may be good: Additive noise on the sharp ridge. In *Proc. of GECCO '08*, pages 511–518. ACM, 2008. doi: 10.1145/1389095.1389192.

[21] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[22] P. T. H. Nguyen and D. Sudholt. Memetic algorithms beat evolutionary algorithms on the class of hurdle problems. In *Proc. of GECCO '18*, 2018. To appear.

[23] G. Ochoa and N. Veerapen. Deconstructing the big valley search space hypothesis. In *Proc. of EvoCOP 2016*, pages 58–73. Springer, 2016.

[24] A. Prügel-Bennett. When a genetic algorithm outperforms hill-climbing. *Theoretical Computer Science*, 320(1):135 – 153, 2004.

[25] A. Prugel-Bennett, J. Rowe, and J. Shapiro. Run-time analysis of population-based evolutionary algorithm in noisy environments. In *Proc. of FOGA '15*, pages 69–75. ACM, 2015. doi: 10.1145/2725494.2725498.

[26] C. Qian, Y. Yu, K. Tang, Y. Jin, X. Yao, and Z.-H. Zhou. On the effectiveness of sampling for evolutionary optimization in noisy environments. *Evolutionary Computation*, . doi: 10.1162/EVCO_a_00201. To appear.

[27] C. Qian, Y. Yu, and Z.-H. Zhou. Analyzing evolutionary optimization in noisy environments. *Evolutionary Computation*, . doi: 10.1162/EVCO_a_00170. To appear.

[28] C. Qian, C. Bian, W. Jiang, and K. Tang. Running time analysis of the (1+1)-EA for Onemax and Leadingones under bit-wise noise. In *Proc. of GECCO '17*, pages 1399–1406. ACM, 2017. doi: 10.1145/3071178.3071347.

[29] S. Rana, L. D. Whitley, and R. Cogswell. Searching in the presence of noise. In *Proc. of PPSN IV*, pages 198–207. Springer, 1996.

[30] C. R. Reeves. Landscapes, operators and heuristic search. *Annals of Operations Research*, 86(0):473–490, 1999.

[31] 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. doi: http://dx.doi.org/10.1016/j.tcs.2013.09.036.

[32] D. Sudholt and C. Thyssen. Running time analysis of ant colony optimization for shortest path problems. *Journal of Discrete Algorithms*, 10:165–180, 2012. doi: DOI:10.1016/j.jda.2011.06.002.

[33] D. Sudholt and C. Thyssen. A simple ant colony optimizer for stochastic shortest path problems. *Algorithmica*, 64(4):643–672, 2012.