eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Analysing the Robustness of Evolutionary Algorithms to Noise: Refined Runtime Bounds and an Example Where Noise is Beneficial

Dirk Sudholt[1] ⬡

Abstract

We analyse the performance of well-known evolutionary algorithms, the $(1 + 1)$ EA and the $(1 + \lambda)$ EA, in the prior noise model, where in each fitness evaluation the search point is altered before the evaluation with probability p. We present refined results for the expected optimisation time of these algorithms on the function LEADINGONES, where bits have to be optimised in sequence. Previous work showed that the $(1 + 1)$ EA on LEADINGONES runs in polynomial expected time if $p = O((\log n)/n^2)$ and needs superpolynomial expected 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 $\Theta(n^2) \cdot \exp(\Theta(\min\{pn^2, n\}))$ for all $p \leq 1/2$, 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 surprisingly sensitive to noise. 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. We prove that in this particular setting noise can have a highly beneficial effect on performance.

Keywords Evolutionary algorithms · Noisy optimisation · Robustness · Runtime analysis · Theory · Uncertainty

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

1   University of Sheffield, Sheffield S1 4DP, UK

                      ⨊ Springer

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 [30]. Evolutionary algorithms are well suited for dealing with these challenges due to their use of a population, and because they can often recover quickly from setbacks resulting from noise or dynamic changes. They have proven to work well in many applications to combinatorial problems [6].

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, 29, 36] 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 [18] in the context of a simple algorithm called $(1 + 1)$ EA on the well-known function ONEMAX $(x) := \sum_{i=1}^{n} x_i$, which simply counts the number of bits set to 1. He considered a setting now known as *one-bit prior noise*, where with probability p a uniformly 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 [25] studied a more general class of algorithms, including the $(1 + 1)$ EA, the $(1 + \lambda)$ EA that generates λ new solutions (offspring) in parallel and picks the best one, and the $(\mu + 1)$ EA that keeps a population of μ search points. They considered prior noise and *posterior noise*, where posterior noise means that noise is added to the fitness value, and 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. They also considered the function LEADINGONES $(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$ that counts the length of the longest prefix that only contains bits set to 1. 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 $(\mu + 1)$ 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 $(1 + \lambda)$ EA of $O(\lambda n + n^2)$ if $p \leq 0.028/n$ and $72 \log n \leq \lambda = o(n)$. Note that their bound simplifies to $O(n^2)$ since $\lambda = o(n)$.

Dang and Lehre [9] gave general results for prior and posterior noise in non-elitist evolutionary algorithms, that is, evolutionary algorithms where the best fitness in the population may decrease. The same authors [10] also considered noise resulting from only partially evaluating search points.

In terms of posterior noise, Sudholt and Thyssen [56] 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 et al. [14] showed that this problem can be avoided if the parent is reevaluated in each iteration. Feldmann and Kötzing [20] further analysed the performance of fitness-proportional updates. Friedrich et al. [22] showed that the compact Genetic Algorithm and ACO [21] are both efficient under extreme Gaussian posterior noise, while a simple $(\mu + 1)$ EA is not.

Prugel-Bennett et al. [46] 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 et al. [51] showed that noise can be handled efficiently by combining reevaluation and threshold selection. Akimoto et al. [1] as well as Qian et al. [50] showed that resampling can essentially eliminate the effect of noise.

Qian et al. [48] 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 is bit-wise noise with parameter q. For LEADINGONES they improve results from [25], 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.

Very recently, Bian et al. [5] considered the general noise model (p, q) for ONEMAX and LEADINGONES and showed that for LEADINGONES the $(1 + 1)$ EA needs polynomial expected time if $p = O((\log n)/n^2)$ or $pq = O((\log n)/n^3)$. It needs superpolynomial time if $p = \omega((\log n)/n)$ and $pq = \omega((\log n)/n^2)$.

In this work we improve previous results for prior noise on the function LEADINGONES. Recall that LEADINGONES $(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$ counts 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 [3, Sect. 4]. Multiobjective variants like LOTZ are popular example functions in the theory of evolutionary multiobjective optimisation [7, 16, 23, 24, 32, 38, 47].

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.

Table 1 Overview of results on the expected optimisation time on LEADINGONES with prior noise

Setting	Previous work	This work
(1+1) EA, one-bit noise p	$O(n^2)$ if $p \le 1/(6en^2)$ [25, Cor. 18] $2^{\Omega(n)}$ if $p = 1/2$ [25, Thm. 20] polynomial if $p = O((\log n)/n^2)$ [48, Thm. 14] superpolynomial if $p = \omega((\log n)/n) \cap o(1)$ [48, Thm. 14] exponential if $p = \Omega(1)$ [48, Thm. 14]	
(1+1) EA, bit-wise noise $(p, 1/n)$	polynomial if $p = O((\log n)/n^2)$ [48, Thm. 8] superpolynomial if $p = \omega((\log n)/n) \cap o(1)$ [48, Thm. 9] exponential if $p = \Omega(1)$ [48, Thm. 10]	$\Theta(n^2) \cdot e^{\Theta(\min\{pn^2, n\})}$ if $p \le 1/2$
(1+1) EA, bit-wise noise $(1, p/n)$	polynomial if $p = O((\log n)/n^2)$ [48, Thm. 11] superpolynomial if $p = \omega((\log n)/n) \cap o(1)$ [48, Thm. 12] exponential if $p = \Omega(1)$ [48, Thm. 13]	
(1+1) EA, bit-wise noise $(p', q/n)$	polynomial if $p := p' \min\{q, 1\} = O((\log n)/n^2)$ [5, Thm. 5] superpolynomial if $p := p' \min\{q, 1\} = \omega((\log n)/n)$ [5, Thm. 6]	
(1+λ) EA, one-bit noise p	$O(\lambda n + n^2)$ if $p \le 0.028/n$ and $72 \log n \le \lambda = o(n)$ [25, Cor. 24]	$O\left(n^2 \cdot e^{O(pn/\lambda)}\right)$ if $p \le 1/2$ and $3.42 \log n \le \lambda = O(n)$

Results for the $(1 + 1)$ EA also hold for asymmetric one-bit noise, for which no results on LEADINGONES are available, with the caveat that for $p = \omega(1/n)$ we only have an upper bound of $O(n^2) \cdot e^{O(pn^2)}$. The bound $O(\lambda n + n^2)$ from [25] was simplified to $O(n^2)$ using their condition $\lambda = o(n)$

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 $\Theta(n^2) \cdot \exp(\Theta(\min\{pn^2, n\}))$, which is tight up to constant factors in the exponent of the term $\exp(\Theta(\min\{pn^2, n\}))$ that reflects the slowdown resulting from noise. This shows that the time is $\Theta(n^2)$ if $p = O(1/n^2)$, polynomial if $p = O((\log n)/n^2)$, superpolynomial if $p = \omega((\log n)/n^2)$ and exponential ($e^{\Theta(n)}$) if $p = \Omega(1/n)$. This improves previous lower bounds that only showed superpolynomial times for $p = \omega((\log n)/n)$, and exponential times for $p = \Omega(1)$, which are both too large by a factor of n.

The upper bound (Sect. 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. A similar argument was used independently in [11] to derive precise and general results for the $(1 + 1)$ EA on noisy and dynamic ONEMAX; their approach also works for shorter periods of time without noise where the algorithm makes progress towards the optimum. The lower bound (Sect. 4) follows arguments from Rowe and Sudholt [54] who analysed the performance of the non-elitist algorithm $(1, \lambda)$ EA on LEADINGONES.

In Sect. 5 we show an improved upper bound for the $(1 + \lambda)$ EA on LEADINGONES. Finally, in Sect. 6 we show that on the class of HURDLE problems [45], 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.

This manuscript extends a preliminary version [55] that contained parts of the results. In this extension, conditions on bit-wise noise were relaxed in the context of the $(1 + 1)$ EA to allow for larger noise values. An exponential upper bound for the $(1 + 1)$ EA was added to obtain asymptotically tight exponents for all reasonable noise strengths. Several empirical analyses were added to complement the theoretical results for LEADINGONES and HURDLE.

2 Preliminaries

Algorithm 1 shows the $(1 + \lambda)$ 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 $(1 + \lambda)$ 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: $(1+\lambda)$ EA with prior noise

Choose x uniformly at random.
while *termination criterion not met* **do**
    for $i = 1, \ldots, \lambda$ **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 i uniformly at random from $\arg\max\{f_1, \ldots, f_\lambda\}$.
    if $f_i \geq f(\text{noise}(x))$ **then** $x := y_i$;

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 [27].

One-bit noise(p) [18, 25]: with probability $1 - p$, $\text{noise}(x) := x$ and otherwise $\text{noise}(x) := x'$ where in x', compared to x, one bit chosen uniformly at random was flipped.

Bit-wise noise(p, q) [48]: with probability $1 - p$, $\text{noise}(x) := x$ and otherwise $\text{noise}(x) := x'$ where in x', compared to x, each bit was flipped independently with probability q.

Asymmetric one-bit noise(p) [51]: 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 uniformly random 0-bit is flipped, with probability $1/2$ a uniformly random 1-bit is flipped, and if $x \in \{0^n, 1^n\}$ a uniformly random bit is flipped.

The special case $(1, q)$ denotes bit-wise noise as investigated in [25]. We often write $(p, q/n)$ for bit-wise noise instead of (p, q) as then q plays a similar role as p in one-bit prior noise p, which allows for a more unified presentation of results: we obtain identical noise thresholds across both models (thresholds for q in the $(1, q)$ model are by a factor of n smaller than those for p [48]). Note that we do generally allow $q > 1$, while in our preliminary work [55] q was restricted to $q \leq 1$. The conditions from [5] for $(p, q/n)$ bit-wise noise simplify to $p \min\{q, 1\} = O((\log n)/n^2)$ for polynomial expected times and $p \min\{q, 1\} = \omega((\log n)/n)$ for superpolynomial times, respectively.

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 $(p, q/n)$, $\Pr(\text{noise}(x) \neq x) = p(1 - (1 - q/n)^n)$ as noise occurs with probability p and at least one bit is flipped with probability $1 - (1 - q/n)^n$. We simplify the last expression using the following inequalities for all $0 \leq p \leq 1$ and $\ell \in \mathbb{N}$.

$$\frac{1}{2}\min\{p\ell, 1\} \leq \frac{p\ell}{1 + p\ell} \leq 1 - (1 - p)^\ell \leq \min\{p\ell, 1\} \tag{1}$$

The second and third inequality are shown in [2, Lemma 6], and the first one follows from considering the two cases $p\ell \leq 1/2$ and $p\ell > 1/2$. Thus $\Pr(\text{noise}(x) \neq x)$ is tightly bounded as follows:

$$\frac{p}{2}\min\{q,1\} \le p(1-(1-q/n)^n) \le p\min\{q,1\}. \tag{2}$$

We often limit our considerations to $p \le 1/2$ for one-bit noise as otherwise more than half of the time, the optimum will not be recognised as an optimum. This can lead to counterintuitive effects. For instance, [49, Theorem 3.3] for bit-wise noise with $p = 1$ shows that increasing the sample size for the $(1+1)$ EA with resampling can turn a polynomial expected time on LEADINGONES into an exponential time; this is essentially because states close to the optimum become more appealing than the optimum itself. For bit-wise noise $(p, q/n)$ we assume $q/n \le 1/2$ as otherwise noise(x) is more likely return search points that are closer to the bit-wise complement $\bar{x}$ of x than to x itself. With $q/n \le 1/2$ the worst possible noise is $q/n = 1/2$ where noise(x) is chosen uniformly at random from the whole search space, irrespective of x.

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 based on the observation that with a certain probability, a run will complete while not being affected by uncertainty. It is formulated for iterative algorithms that represent Markov chains and maintain a single search point, called *trajectory-based algorithms*. It is easy to extend the definition to population-based algorithms or non-Markovian algorithms as well.[1] The $(1+1)$ EA and the $(1+\lambda)$ EA are both trajectory-based algorithms as they both evolve a single search point. The definition also includes randomised local search (RLS), the Metropolis algorithm, the $(1, \lambda)$ EA [26] or the Strong Selection Weak Mutation (SSWM) algorithm [44].

Definition 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 $M_{\mathcal{A},f}$

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

and the worst-case median optimisation time

---

[1] The following results can be adapted to populations or non-Markovian algorithms when taking the worst case over all possible internal states in the upcoming definitions of worst-case expected optimisation times and worst-case median optimisation times.

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

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 as shown in the following simple lemma.[2]

Lemma 1 *For every $\mathcal{A}$ and every f, $M_{\mathcal{A},f} \leq 2E_{\mathcal{A},f}$.*

Proof Let x be a search point with maximal $M_{\mathcal{A},f}(x)$ value. Then $M_{\mathcal{A},f}(x) \leq 2E_{\mathcal{A},f}(x)$ by Markov's inequality. Noting $2E_{\mathcal{A},f}(x) \leq 2E_{\mathcal{A},f}$ completes the proof. $\square$

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. It uses the notion of a "failure event", which is an event that may occur independently from other iterations and independently from the current state of the algorithm and which may move the algorithm to an arbitrary state. The name "failure event" is used since in typical applications of this framework, the mentioned event may disrupt the progress of the algorithm.

Theorem 2 *Consider a trajectory-based algorithm $\mathcal{A}$ in a setting where in each iteration a failure event occurs independently from other iterations and the state of $\mathcal{A}$ with probability at most $0 \leq p < 1$. Consider any function f on which an iterative algorithm $\mathcal{A}$ has worst-case median optimisation time at most M if $p = 0$. Then the worst-case expected optimisation time of $\mathcal{A}$ with failure probability p is at most*

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

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 $\frac{1}{1-p} = 1 + \frac{p}{1-p} \leq e^{p/(1-p)}$. $\square$

In the setting of prior noise, Theorem 2 implies the following. A failure event may occur if any of the offspring, or the parent, experiences noise. The following theorem is formulated for any v search points being evaluated in one iteration.

Theorem 3 *Consider a trajectory-based algorithm $\mathcal{A}$ that evaluates up to v search points in each iteration. For every function f on which $\mathcal{A}$ has worst-case median*

---

[2] Much stronger results can be shown, but Lemma 1 is sufficient for our purposes.

optimisation time M without prior noise, its worst-case expected optimisation time is at most

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

for all prior noise models where, for all x, $\Pr(\text{noise}(x) \neq x) \leq p$, including:

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

Proof In all mentioned noise models, the probability of noise occurring in one search point is at most p; this is immediate for one-bit noise and it is $p'(1 - (1 - q/n)^n) \leq p' \min\{q, 1\}$ for bit-wise noise by (2). Since noise is applied to all search points independently, noise occurs in one iteration with probability at most $p^* := 1 - (1 - p)^v$. Invoking Theorem 2 with parameter p^* and the occurrence of noise as failure event yields the first claimed bound. The inequality follows as in the Proof of Theorem 2. □

We remark that Theorem 2 (and straightforward extensions to populations) also applies in many other settings, for example in

– restart strategies that restart the algorithm in each iteration with probability p,
– non-elitist algorithms like the $(1, \lambda)$ EA, where the failure event could be defined as the best fitness decreasing,
– stochastic ageing [8, 41], an approach from artificial immune systems, where individuals are suddenly killed off with a fixed probability and the failure event is that the minimum fitness in the population decreases during an appropriately defined time period, see Lemma 2 in [41],
– dynamic optimisation where p is the probability of the fitness function changing, if M is taken as (an upper bound for) the worst-case median optimisation time *for all possible fitness functions* that can be attained in the considered dynamic setting.

For LEADINGONES, Theorem 3 implies the following.

Theorem 4 *The expected optimisation time of the $(1 + 1)$ EA with prior noise probability $p \leq 1/2$ for each of the settings from Theorem 3 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 The upper bound follows directly from Theorem 3 with $v = 2$ (as the $(1 + 1)$ EA evaluates parent and offspring in each generation), $2p/(1 - p) = O(p)$,

and the fact that the worst-case expected optimisation time of the $(1 + 1)$ EA on LEADINGONES is $O(n^2)$ [19], hence by Lemma 1 the worst-case median optimisation time is $M = O(n^2)$. $\square$

Despite the simplicity of the above proofs, Theorem 4 matches, unifies and generalises the best known results [5, 48] which only classify the expected optimisation time on LEADINGONES as being either polynomial, superpolynomial, or exponential (see Table 1). It also gives results for asymmetric one-bit noise, for which no results on LEADINGONES are available.

3.1 An Exponential Upper Bound for Large Noise

For very large noise levels p, Theorem 4 gives an upper bound of essentially $e^{O(pn^2)}$, which can be as bad as $e^{O(n^2)}$ for $p = \Omega(1)$. This is clearly too pessimistic as the expected time to create the optimum by mutation is at most $n^n = e^{n \ln n}$ for *every* fitness function and every initial search point.

We therefore provide a new, tailored upper bound for large noise levels, showing that the expected optimisation time is at most $e^{O(n)}$. To this end, we will prove that the $(1 + 1)$ EA converges to a stationary distribution π in which the optimum 1^n has stationary mass $\pi(1^n) \geq 2^{-n}$. We then bound the mixing time, that is, the time until the algorithm has approached the stationary distribution such that the optimum is found with a probability close to $\pi(1^n)$. Throughout this section we assume that the reader is familiar with the foundations of Markov chain theory and mixing times as described in relevant text books like [35].

The following lemma shows that transitions to higher fitness values are at least as likely as transitions to lower values.

Lemma 5 *Let* $\Pr(x \to y)$ *denote the probability that the* $(1 + 1)$ EA *with prior noise transitions from* x *to* y *in one generation. Then for all* x, y *with* LEADINGONES$(x) <$ LEADINGONES(y) *we have* $\Pr(x \to y) \geq \Pr(y \to x)$ *in each of the following settings:*

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

Proof A transition from x to y is made if and only if mutation of x results in y and y is accepted. Since the probability of mutation of x creating y is equal to that of mutation of y creating x, we just need to show that the probability of accepting y as offspring of x is no smaller than the probability of accepting x as offspring of y.

Let i denote the smallest index of any bit flipped in the parent's noise, and $i := \infty$ if there is no such bit. Define j in the same way for the offspring's noise. Abbreviate $\ell := $ LEADINGONES(x).

Now, if $i \leq \ell$ and $i \leq j$ then the offspring will be accepted regardless of whether the parent is x or y. If $j \leq \ell$ and $j < i$ the offspring will be rejected in both scenarios.

Hence, if $\min(i,j) \leq \ell$ selection is determined by noise on the first ℓ bits and we only need to show the claimed inequality for conditional probabilities assuming $\min(i,j) \geq \ell + 1$.

If the parent is x then a transition from x to y will be made if $i > \ell + 1$ and $j \geq \ell + 1$ since then the noisy fitness of the parent is ℓ and the noisy fitness of the offspring is at least ℓ.

If the parent is y then a transition from y to x will be made if $i = \ell + 1$ and $j \geq \ell + 1$ as then the noisy fitness of the parent is ℓ and the noisy fitness of the offspring is at least ℓ.

The above two scenarios cover all cases where $\min(i,j) \geq \ell + 1$. Thus the claim follows if we can show that

$$\Pr(i > \ell + 1 \wedge j \geq \ell + 1) \geq \Pr(i = \ell + 1 \wedge j \geq \ell + 1).$$

Since noise is determined independently for parent and offspring, this is equivalent to

$$\Pr(i > \ell + 1) \cdot \Pr(j \geq \ell + 1) \geq \Pr(i = \ell + 1) \cdot \Pr(j \geq \ell + 1)$$
$$\Leftrightarrow \Pr(i > \ell + 1) \geq \Pr(i = \ell + 1).$$

In the symmetric and asymmetric one-bit noise settings, the left-hand side is at least $1 - p \geq 1/2$ and the right-hand side is at most $p \leq 1/2$. For the bit-wise noise setting, the left-hand side is at least $p(1 - q/n)^{\ell+1} \geq pq/n \cdot (1 - q/n)^{\ell} = \Pr(i = \ell + 1)$. $\square$

The exponential upper bound is stated as follows.

Theorem 6 *The expected optimisation time of the* $(1 + 1)$ EA *with prior noise probability* $p \leq 1/2$ *for each of the specific settings from Theorem* 3, *except for asymmetric one-bit noise, on* LEADINGONES *is at most* $2^{O(n)}$.

Proof If $p = 0$ then the expected optimisation time of the $(1 + 1)$ EA on LEADINGONES is $O(n^2) \leq 2^{O(n)}$, hence we assume $p > 0$ in the following.

We first show that the $(1 + 1)$ EA on LEADINGONES is an ergodic Markov chain, which implies the existence of a stationary distribution π. Ergodicity simply follows from the fact that every search point x can be turned into any other search point y in one generation if mutation of x creates y (probability at least n^{-n}) and LEADINGONES(noise(x)) $= 0$, which happens with probability at least $p/n > 0$ for one-bit noise and probability at least $p'q/n > 0$ for bit-wise noise with $p = p' \min\{q, 1\} > 0$.

To prove the claimed inequality $1/\pi(1^n) \leq 2^n$ we will use the following property of stationary distributions (cf. Proposition 1.19 in [35]):

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

Since by Lemma 5 $\Pr(x \to 1^n) \geq \Pr(1^n \to x)$ for every search point x, $\pi(1^n) \geq \pi(x)$ for all 2^n possible x and thus $\pi(1^n) \geq 2^{-n}$.

It remains to bound the mixing time, that is, the time until the algorithm has gotten close to the stationary distribution (as will be made precise soon). Let p_t be the distribution of the current search point at time t. The difference to the stationary distribution π is described by the *total variation distance* that describes the maximum difference between probabilities for any event A:

$$||p_t - \pi|| := \max_{A \subset \Omega} |p_t(A) - \pi(A)|.$$

In particular, we have $\Pr(x_t = 1^n) \geq \pi(1^n) - ||p_t - \pi|| \geq 2^{-n} - ||p_t - \pi||$.

We now show that $||p_t - \pi|| \leq 2^{-n-1}$ for a suitable $t = \text{poly}(n) \cdot 2^{O(n)}$. This will be achieved by using a coupling (X^t, Y^t). In a nutshell, a coupling is a pair process where, viewed individually, X^t and Y^t are both faithful copies of the original process, the $(1 + 1)$ EA on LEADINGONES. But they may not be independent: they can follow a joint distribution and the coupling ensures that, once they have reached the same state, their states will always be equal. More formally, if $X^t = Y^t$ then $X^{t+1} = Y^{t+1}$. The first point in time where their states become equal, when starting in states $X^0 = x$ and $Y^0 = y$ is called the coupling time T_{xy}.

It is known that the tail of the coupling time, or more precisely the tail of the worst-case coupling time for any initial states x, y, yields a bound on the total variation distance. Using [35, Theorem 5.2] we get

$$||p_t - \pi|| \leq \Pr(\max_{x,y} T_{x,y} > t).$$

We will show the right-hand side becomes less than 2^{-n-1} within $2^{O(n)}$ generations.[3]

We use the following coupling between two copies X^t, Y^t of the $(1 + 1)$ EA, where we identify X^t and Y^t with the $(1 + 1)$ EA 's current search points in the respective chains. During mutation, for bits where X^t and Y^t agree we make the same decisions in both Markov chains. Otherwise, with probability $1/n$ we flip the bit in X^t but not in Y^t, with probability $1/n$ we flip the bit in Y^t but not in X^t, and with the remaining probability $1 - 2/n$ the bit is not flipped at all. We further assume that the same noise is applied in both chains. It is easy to verify that both chains, viewed in isolation, represent faithful copies of the $(1 + 1)$ EA on LEADINGONES, and that after both chains have reached the same state, their states will always be equal as they experience the same mutations and the same noise.

Let Eq_t denote the size of the largest prefix that is identical in X^t and Y^t, i.e., $Eq_t = \max\{i \mid X_1^t \ldots X_i^t = Y_1^t \ldots Y_i^t\}$. Note that if both chains decide to reject their offspring, $Eq_{t+1} = Eq_t$ and if both chains decide to accept then $Eq_{t+1} \geq Eq_t$ due to the way mutations are coupled. Once Eq_t has reached a value of n, both chains will always have the same state.

Let $i := Eq_t < n$ then $X_{i+1}^t \neq Y_{i+1}^t$ by definition of Eq_t. Assume without loss of generality that $X_{i+1}^t = 0$. We first show that $\Pr(Eq_{t+1} > Eq_t \mid Eq_t, Eq_t < n) \geq 1/(3en)$. A sufficient event is that mutation makes bit $i + 1$ equal in X^t and Y^t and the outcome

---

[3] The author conjectures that this mixing time is, in fact, polynomial, but was unable to prove this. This is left as an open problem for future work.

is accepted in both chains. Mutation flips X_{i+1}^t while not flipping $X_1^t, \ldots, X_i^t$ and $Y_1^t, \ldots, Y_{i+1}^t$ with probability $1/n \cdot (1 - 1/n)^i \geq 1/(en)$ as per definition of the coupling mutation flips X_{i+1}^t and does not flip Y_{i+1}^t with probability $1/n$ and every bit $j \leq i$ is not flipped in X^t and Y^t with probability $1 - 1/n$ since $X_j^t = Y_j^t$. The outcome of such a mutation then needs to be accepted in X despite noise. Let α_{i+1} denote the probability of noise flipping any of the first $i + 1$ bits. The offspring will be accepted in X if noise leaves the first $i + 1$ bits intact in both parent and offspring, or if noise does flip at least one bit amongst the first $i + 1$ bits in both parent and offspring, but still the offspring's noisy fitness is at least as good as that of its parent. Noting the symmetry in the latter case, the probability of accepting said mutation is at least $(1 - \alpha_{i+1})^2 + \alpha_{i+1}^2/2 \geq 1/3$ for every possible value α_{i+1}. Together, this shows $\Pr(Eq_{t+1} > Eq_t \mid Eq_t, Eq_t < n) \geq 1/(3en)$.

Note that the first i bits are identical in the noisy parent evaluation of both X^t and Y^t, and they are also identical in the noisy evaluation of both offspring x', y' in X^t and Y^t, respectively. If either of these noisy evaluations is less than i, the decision whether to accept or reject is only based on the first i bits and X^t and Y^t make the same decision. The only problematic case is when noise(X^t), noise(Y^t), noise(x'), and noise(y') all have at least i leading ones as then one Markov chain might accept their offspring while the other might reject theirs. If LeadingOnes(x') and LeadingOnes(y') are both at least i, $Eq_{t+1} \geq Eq_t$ and no harm is done.

However, we might have LeadingOnes$(x') < i$ or LeadingOnes$(y') < i$ in case mutation destroys the prefix of i leading ones (probability at most i/n), but noise flips the same bits, covering up all detrimental mutations. The probability of the latter event is at most p/n for one-bit noise (or 0 in case mutation flipped more than one bit). We call step t a *relevant step* if $Eq_{t+1} \neq Eq_t$. In a relevant step, the conditional probability of increasing Eq_t is $\Omega(1)$ and the probability of increasing Eq_t in at most n subsequent relevant steps, until $Eq_t = n$ is reached, is at least $(\Omega(1))^n = 2^{-\Omega(n)}$.

In the case of bit-wise noise, the probability of decreasing Eq_t is at most $q/n \cdot (1 - q/n)^{i-1}$ as (since $q/n \leq 1/2$) the best case is that mutation has only flipped one bit, which needs to be covered up by noise. The conditional probability of Eq_t increasing in a relevant step is thus at least

$$\frac{1/(3en)}{q/n \cdot (1 - q/n)^{i-1} + 1/(3en)} = \frac{1}{1 + 3eq(1 - q/n)^{i-1}}.$$

The probability of increasing Eq_t in at most n subsequent relevant steps until a value of n is reached is thus at least

$$\prod_{i=1}^{n} \frac{1}{1 + 3eq(1 - q/n)^{i-1}} = \prod_{i=0}^{n-1} \frac{1}{1 + 3eq(1 - q/n)^i}.$$

The reciprocal of this expression is upper bounded by

$$\prod_{i=0}^{n-1} \left(1 + 3eq(1 - q/n)^i\right) \leq \prod_{i=0}^{n-1} \exp\left(3eq(1 - q/n)^i\right)$$

$$= \exp\left(\sum_{i=0}^{n-1} 3eq(1 - q/n)^i\right)$$

$$\leq \exp\left(3eq \sum_{i=0}^{\infty}(1 - q/n)^i\right) = \exp(3en).$$

For both one-bit and bit-wise noise, a relevant step occurs with probability at least $1/(3en)$ (unless the chains have already coupled). Hence the expected waiting time for n relevant steps is at most $3en^2$. Thus, from any initial configuration of X^t and Y^t, the expected time for a sequence of up to n relevant steps all increasing Eq_t until the maximum value n is reached and the chains are coupled is bounded by $\mathrm{E}(\max_{xy} T_{xy}) \leq 3en^2 \cdot e^{O(n)} := t^*$. By Markov's inequality, $\Pr(\max_{xy} T_{xy} \geq 2t^*) \leq 1/2$ and the probability that the process has not coupled within $n + 1$ subsequent phases of length $2t^*$ each is at most 2^{-n-1}.

This shows that the time until the total variation distance to π has decreased to a value of at most 2^{-n-1} is $O(n^3) \cdot 2^{O(n)} = 2^{O(n)}$. Then the probability of sampling the optimum in the next generation is at least $\pi(1^n) - 2^{-n-1} \geq 2^{-n-1}$. If the optimum is not found then, we repeat the above arguments. This establishes an upper bound of $O(n^3) \cdot 2^{O(n)} \cdot 2^{n+1} = 2^{O(n)}$. $\square$

4 A Matching Lower Bound for the $(1 + 1)$ EA on LeadingOnes

The arguments from Sect. 3 and Theorem 2 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 [31], separable functions [17], and for the $(1, \lambda)$ EA on LeadingOnes [54]; parts of this section closely follow the Proof of Theorem 12 in [54] (but had to be adapted to noisy settings).

The main result of this section is the following.

Theorem 7 *The expected optimisation time of the* $(1 + 1)$ EA *with prior noise probability* $p \leq 1/2$ *for each of the settings from Theorem* 3 *on* LeadingOnes *is* $\Omega\left(n^2 \cdot e^{\Omega(pn^2)}\right)$ *if* $p = O(1/n)$ *and* $e^{\Omega(n)}$ *if* $p = \omega(1/n)$. *This is superpolynomial for* $p = \omega((\log n)/n^2)$.

Along with Theorems 4 and 6 and the fact that polynomial factors only account for a $\pm O(\log n)$ term in the exponent, yielding $e^{\Omega(n)} = \Theta(n^2) \cdot e^{\Omega(n)}$, we get the following result.

Theorem 8 *The expected optimisation time of the* $(1+1)$ EA *on* LEADINGONES *is*

$$\Theta(n^2) \cdot e^{\Theta(\min\{pn^2, n\})}$$

for each of the following settings:

1. *one-bit prior noise with probability* $p \le 1/2$ *and*
2. *bit-wise prior noise* $(p', q/n)$ *with* $q/n \le 1/2$ *and* $p := p' \min\{q, 1\}$.

The result is tight up to constants in exponent of the term $\exp(\Theta(\min\{pn^2, n\}))$ that reflects the impact of noise.

Theorem 7 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 [5, 48] and the noise parameter regime $p = O((\log n)/n^2)$ that led to polynomial upper bounds in [5, 48] and in Theorem 4.

Theorem 7 closes this gap by showing that superpolynomial times already occur for noise parameters $p = \omega((\log n)/n^2)$, which is by a factor of $1/n$ smaller than previous results [5, 48]. This shows that the $(1+1)$ EA on LEADINGONES is highly sensitive to noise, especially since the corresponding threshold for ONE-MAX is at $p = \Theta((\log n)/n)$ [18, 25]. Theorem 7 also unifies and generalises all known results for LEADINGONES under prior noise by giving bounds that hold for the whole range of noise parameters p, and for different prior noise models.

In order to prove Theorem 7, we first analyse the probability of the fitness dropping significantly.

Lemma 9 *Consider the setting of Theorem 7 with a current* LEADINGONES *value of* $i \ge 2$. *Then the probability that the* LEADINGONES *value decreases to a value in* $[i/4, i/2]$ *in one generation is* $\Omega(pi^2/n^2)$. *This is* $\Omega(p)$ *if* $i = \Omega(n)$.

Proof Mutation flips a bit at position $\{\lceil i/4 \rceil, \ldots, \lfloor i/2 \rfloor\}$ and leaves the other bits unflipped with probability $\Omega(i/n)$ (note that the set of positions is non-empty since $i \ge 2$). Let $i/4 \le i^* \le i/2$ denote the position of the bit flipped during mutation. Let i_x denote the smallest index of any bit flipped during the parent's noise and $i_x := \infty$ if no such bit exists. Define i_y in the same way for the offspring. We claim that after a mutation as described above, the probability that the offspring is accepted regardless is $\Omega(pi/n)$. A sufficient condition for this to happen is that $i_x \le i/4 \le i^*$ and $i_y \ge i_x$.

For one-bit noise, we have $\Pr(i_x \le i/4) \ge pi/(4n)$. For asymmetric one-bit noise we get $\Pr(i_x \le i/4) \ge pi/(8n)$ as with probability $p/2$, one of at most n 1-bits is flipped. For bit-wise noise $(p', q/n)$ with $p := p' \min\{q, 1\}$ we have

$\Pr(i_x \leq i/4) \geq p'(1 - (1 - q/n)^{i/4}) \geq p'/2 \cdot \min\{iq/(4n), 1\}$ by (1). Since $1 \geq i/(4n)$, this is at least $p'/2 \cdot \min\{iq/(4n), i/(4n)\} = p'i/(8n) \cdot \min\{q, 1\} = pi/(8n)$.

For all noise models, we claim that $\Pr(i_y \geq i_x \mid i_x \leq i^*) \geq 1/2$. If $i_y > i^*$ then $i_y \geq i_x$ with probability 1; otherwise we argue that $\Pr(i_y \geq i_x \mid i_x \leq i^*, i_y \leq i^*) \geq \Pr(i_x \geq i_y \mid i_x \leq i^*, i_y \leq i^*)$ as parent and offspring are subject to the same independent noise under identical conditions.

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 in $[i/4, i/2]$. Since all events are independent (or conditionally independent), multiplying these probabilities implies the claim. □

As argued in [54] for the $(1, \lambda)$ EA, such a fallback is not too detrimental per se as the $(1 + 1)$ EA might recover from this easily. Assume the fitness has dropped to $i^* \in [i/4, i/2]$. If the bits between $i^* + 1$ 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 $i^* + 1$ 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 observed in [12, Proof of Theorem 10] and formalised in [31, Lemma 3] stated below. The lemma states that the probability distribution of a bit subjected to random mutations rapidly approaches a uniform distribution.

Lemma 10 (Adapted from Lässig and Sudholt [31]) *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 to a fitness of i^* for some $n/6 \leq i^* \leq n/2$. We speak of a *lasting fallback* if in the $2n/(1 - p)$ generations directly following a fallback the following holds:

1. all acceptance decisions are made independently from bit values at positions $i^* + 2, \ldots, n$,
2. bit $i^* + 1$ is never flipped during mutation and
3. in at least $n/2$ generations the offspring is accepted.

A lasting fallback implies that the fitness remains at most i^* during at least $n/2$ accepted steps. In these accepted steps, the bits at positions $i^* + 2, \ldots, n$ are mutated independently from acceptance decisions and hence take on a near-random state.

We remark that in a noise-free setting, so long as bit $i^* + 1$ is never flipped, the acceptance decisions would trivially be independent from bit positions $i^* + 2, \ldots, n$.

In a setting with noise, however, these bits might play a role as bit $i^* + 1$ might be flipped by noise, and then the acceptance decision might depend on further bits. Hence more careful arguments are needed.

We also say that the initial search point is a lasting fallback if its fitness is at most $n/2$. If i^* is the initial fitness, the bits at positions $i^* + 2, \dots, n$ take on a uniformly random state.

The following lemma estimates probabilities for fallbacks and lasting fallbacks. For asymmetric noise we assume that there is a linear number of zeros in the current search point. We will show later that this assumption is met with an overwhelming probability.

Lemma 11 *Consider any of the specific settings from Theorem* 3. *For asymmetric one-bit noise, assume that the number of zeros in the current search point is $\Omega(n)$. If $p \leq 1/2$ 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 directly from Lemma 9 as $f^* = \Omega(n)$ and the fitness after a fallback is at least $n/6$ and at most $n/2$.

It remains to estimate the probability of a fallback becoming a lasting fallback.

Let i^* be the fitness obtained during a fallback and let j_t be the index of the first bit flipped by the parent's noise in generation t. We call a generation t *good* if

– bit i^* is not flipped during mutation and
– $j_t \neq i^* + 1$.

In a good generation, the LEADINGONES value cannot increase beyond i^*. The second condition implies that the parent's noisy fitness is at most i^*. The offspring is accepted if and only if its noisy fitness is better than the parent's noisy fitness. As the latter is at most i^*, the decision whether to accept the offspring only depends on bits at positions $1, \dots, i^* + 1$ and is independent from bits at positions $i^* + 2, \dots, n$.

If all generations since the fallback have been good then the LEADINGONES value remains at most i^* and decisions are independent from bits $i^* + 2, \dots, n$ as claimed.

We estimate the probability of all $2n/(1 - p)$ generations being good. For any generation t, the probability of the first event is $1 - 1/n$. The probability of the second event is at least $1 - p/n \geq 1 - 1/n$ for one-bit noise. For bit-wise noise $(p', q/n)$, it is at least $1 - p'q/n \cdot (1 - q/n)^{i^*} \geq 1 - p'q/n \cdot (1 - q/n)^{n/6} \geq 1 - q/n \cdot e^{-q/6} \geq 1 - 6/(en)$ as the function $q/n \cdot e^{-q/6}$ is maximised for $q = 6/n$. For asymmetric one-bit noise, the probability of the second event is at least $1 - O(1/n)$ by assumption on the number of zeros in the current search point.

Hence, in all settings, the probability of a generation t being good is at least $1 - O(1/n)$ by a union bound and the probability that all $2n/(1 - p)$ generations are good is $(1 - O(1/n))^{2n/(1-p)} = \Omega(1)$.

Assuming that these generations are all good, we finally estimate the number of accepted generations under this condition. Using $\Pr(A \mid B) = \Pr(A \cap B)/P(B) \geq \Pr(A \cap B)$, we lower-bound the probability of

a generation t being accepted and good. This happens if bits $1, \ldots, i^* + 1$ are not flipped during mutation (probability at least $(1 - 1/n)^n$), bit $i^* + 1$ is set to 0 in the noisy parent (probability at least $1 - O(1/n)$ as estimated above) and the offspring does not suffer from noise (probability at least $1 - p$). Together, the probability of an accepted generation conditional on it being good is at least $(1 - 1/n)^n \cdot (1 - O(1/n)) \cdot (1 - p) \geq (1 - p)/3$ if n is large enough. The expected number of accepted generations in $2n/(1 - p)$ good generations is at least $2n/3$ and by Chernoff bounds, the probability of having at least $n/2$ accepted generations is $1 - 2^{-\Omega(n)}$.

Together, all three criteria in the definition of lasting fallbacks hold with probability $\Omega(1)$. $\square$

The following lemma shows that the assumption for asymmetric one-bit noise from Lemma 14 is met with overwhelming probability. If the LEADINGONES value does not exceed a given threshold, the suffix of bits past this threshold evolves almost uniformly at random.

Lemma 12 *Consider the* $(1 + 1)$ EA *on* LEADINGONES *with asymmetric one-bit noise and parameter p. For every constant* $0 < \gamma < 1$, *as long as the* LEADINGONES *value is strictly less than* $n - \gamma n$, *the number of zeros on the* γn *last bit positions is at least* $\gamma n/3$ *throughout the first* $2^{\kappa n}$ *generations, for a constant* $\kappa > 0$, *with probability* $1 - 2^{-\Omega(n)}$.

Proof With probability $1 - 2^{-\Omega(n)}$, the $(1 + 1)$ EA starts with at least $5/12 \cdot \gamma n$ zeros on the last γn positions (hereinafter called the suffix). We aim to apply the negative drift theorem [42, 43] to the number of zeros in the suffix and the interval $[\gamma n/3, 5/12 \cdot \gamma n]$. Let $Z_0, Z_1, \ldots$ denote this value over time.

Let A_t be the event that at time t, the offspring is accepted and the noisy LEADINGONES values of both parent and offspring are less than $n - \gamma n$. This implies that mutations of the suffix are independent from the acceptance decision. In expectation, Z_t/n bits will flip from 0 to 1 and $(\gamma n - Z_t)/n$ bits will flip from 1 to 0. Thus,

$$\mathrm{E}\left(Z_{t+1} - Z_t \mid Z_t, A_t, Z_t \leq 5/12 \cdot \gamma n\right) = \frac{\gamma n - Z_t}{n} - \frac{Z_t}{n} = \frac{\gamma n - 2Z_t}{n} \geq \frac{\gamma}{6}.$$

Let B_t be the event that the offspring is accepted and the noisy LEADINGONES value of parent or offspring is at least $n - \gamma n$. In order for the noisy LEADINGONES value of any search point to exceed the real LEADINGONES value, the first 0-bit has to flip. While $Z_t \geq \gamma n/3$, this has probability at most $p/(2Z_t) \leq 3p/(2\gamma n) \leq 3/(2\gamma n)$. By the union bound, the probability that this happens for the parent or the offspring is at most $3/(\gamma n)$, hence $\Pr(B_t) \leq 3/(\gamma n)$. We pessimistically assume that under event B_t, at most $\sqrt{n}$ bits in the suffix flip from 0 to 1, hence decreasing Z_t by $\sqrt{n}$. This assumption is justified as the probability of flipping any $\sqrt{n}$ bits is exponentially small. Thus

$$\mathrm{E}\left(Z_{t+1} - Z_t \mid Z_t, B_t\right) \geq -\sqrt{n}.$$

Note that $A_t \cup B_t$ denotes the event that a step is accepted. Moreover, $\Pr(A_t \cup B_t) \geq 1/(2e)$ since a sufficient event for acceptance is that the first $n-1$ bits are not flipped and the noisy offspring is no worse than the noisy parent, which by symmetry has probability at least $1/2$. Thus, $\Pr(B_t \mid A_t \cup B_t) \leq \Pr(B_t)/\Pr(A_t \cup B_t) \leq 6e/(\gamma n)$. Together,

$$E(Z_{t+1} - Z_t \mid Z_t, A_t \cup B_t) \geq \Pr(A_t \mid A_t \cup B_t) \cdot \frac{\gamma}{6} - \Pr(B_t \mid A_t \cup B_t) \cdot \sqrt{n}$$

$$\geq \left(1 - \frac{6e}{\gamma n}\right) \cdot \frac{\gamma}{6} - \frac{6e}{\gamma n} \cdot \sqrt{n} = \Omega(1).$$

This establishes a constant drift when $\gamma n/3 \leq Z_t \leq 5/12 \cdot \gamma n$. By standard arguments, the second condition of the negative drift theorem is met since the transitions of Z_t are bounded by the number of flipping bits, which has an exponential decay [42, Proof of Theorem 5]. Then the negative drift theorem [42, 43] implies the claim. □

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, for a constant $c > 0$, are needed to increase the best fitness since the latest lasting fallback by at least $n/6$.

Lemma 13 *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 We pessimistically overestimate the probability of a fitness improvement due to the effects of noise in generations from t to $t + cn^2$: we assume that noise never leads to a decrease in the number of leading ones. Secondly, we call a step *successful* if the first 0-bit is flipped during mutation or if it is flipped during the parent's or offspring's noise. In this case we assume that this bit becomes part of the leading ones for the next generation and the next parent's fitness is determined by the position of the first 0-bit amongst the following bits. The probability of a successful step is still bounded from above by $3/n$.

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 $t_{\mathrm{mix}} = n/2$ times and these mutations were independent of the acceptance decision (by definition of a lasting fallback). 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* [37, p. 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 10 we know that then the probability such a bit is set to 1 is at most

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

Note that due to our pessimistic assumptions concerning successful steps, the bits following the first 0-bit will always be irrelevant for the decision whether or not to accept the offspring. Hence the above probability bound also holds after generation t.

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 $6cn$ steps are successful or
2. during at most $6cn$ successful steps the total fitness gain is at least $n/6$.

The probability of a successful step is always at most $3/n$ as mentioned earlier. 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 $6cn$—plus a sum of up to $6cn$ geometric random variables to account for additional bits gained (these additional bits are often called "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 14 *Let $c > 0$ be any constant. Within cn^2 generations where the current fitness is larger than $f^* = 2n/3$, 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

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

□

Now we prove Theorem 7.

Proof of Theorem 7 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 13, reaching a fitness of at least $f^* = 2n/3$ from there takes time at least cn^2 with overwhelming probability, for a suitably small constant $c > 0$. Applying Lemma 13 every time the fitness increases to at least f^*, the $(1 + 1)$ EA does not find a search point with fitness at least $3n/4$ (let alone an optimum) within the next cn^2 generations where the fitness is at least f^*, with overwhelming probability. This implies that, for asymmetric one-bit noise, Lemma 12 is in force, with respect to a prefix of the last $n/4$ bits. Then by Lemma 14 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. $\square$

5 Improved Results for Offspring Populations

The general Theorem 2 can also be used in the context of offspring populations in the $(1 + \lambda)$ 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. The current fitness can decrease 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 15 *Consider the $(1+\lambda)$ EA in a prior noise model where $\Pr(\text{noise}(y) \neq y) \leq p$ 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 For every offspring, the probability that a copy of x is created is $(1 - 1/n)^n$, and the probability that a copy of x is created and affected by noise is at most $p(1 - 1/n)^n$. Hence, the probability that for all offspring either no copy of x is created or a copy of x is created and affected by noise is at most

$$\left(1 - \left(1 - \frac{1}{n}\right)^n + p\left(1 - \frac{1}{n}\right)^n\right)^\lambda = \left(1 - \left(1 - \frac{1}{n}\right)^n (1 - p)\right)^\lambda.$$

In addition, the probability that the parent x itself is affected by noise is at most p. Hence the sought probability is at most

$$p\left(1 - \left(1 - \frac{1}{n}\right)^n (1 - p)\right)^\lambda.$$

For the second bound we use $(1 - 1/n)^n = (1 - 1/n)(1 - 1/n)^{n-1} \geq (1 - 1/n) \cdot 1/e,$

$$\left(1 - \left(1 - \frac{1}{n}\right)^n (1-p)\right)^\lambda \leq \left(1 - \frac{1}{e}\left(1 - \frac{1}{n}\right)(1-p)\right)^\lambda$$

$$= \left(1 - \frac{1}{e}(1-p)\right)^\lambda \left(\frac{1 - \frac{1}{e}\left(1 - \frac{1}{n}\right)(1-p)}{1 - \frac{1}{e}(1-p)}\right)^\lambda$$

$$= \left(1 - \frac{1}{e}(1-p)\right)^\lambda \left(1 + \frac{\frac{1}{en}(1-p)}{1 - \frac{1}{e}(1-p)}\right)^\lambda$$

$$= \left(\frac{e - (1-p)}{e}\right)^\lambda \left(1 + \frac{\frac{1}{n}(1-p)}{e - (1-p)}\right)^\lambda$$

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

□

Our aim is to apply Theorem 2 where the failure event is the union of the event described in Lemma 15 and other events described later. However, we still need a bound on the worst-case median optimisation time, or (by Lemma 1) the worst-case expected optimisation time, assuming that the algorithm always retains at least one copy of the current search point.

Note that we cannot simply use a runtime result for the $(1 + \lambda)$ EA without noise as noise can still affect the generated offspring; the only condition we can rely on is that we cannot lose all copies of the current search point. If noise is disruptive, the $(1 + \lambda)$ EA may behave like having a smaller effective offspring population, the size of which is random. Note that we cannot pessimistically use a bound on the $(1 + 1)$ EA to upper bound the time of the $(1 + \lambda)$ EA in this setting as different offspring population sizes can affect search dynamics in unforeseen ways. Jansen et al. [28] presented a problem class where different offspring population sizes lead to very different performance.

The following theorem gives improved upper bounds for one-bit noise and bit-wise noise.[4]

Theorem 16 *The expected number of function evaluations for the $(1 + \lambda)$ EA with prior noise parameter $p \leq 1/2$ on* LEADINGONES *with $\log_{\frac{e}{e-1/2}}(n) \leq \lambda = O(n)$ is*

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

---

[4] 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. Arguments similar to those in Lemma 12 could address this, however for simplicity we do not consider asymmetric noise. We also restrict bit-wise noise to $q/n \leq 1/n$ for reasons explained after the Proof of Theorem 16.

in each of the following settings:

1. *one-bit prior noise with probability $p < 1$ and*
2. *bit-wise prior noise $(p', q/n)$ with $q/n \leq 1/n$ and $p := p' \min\{q, 1\}$.*

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 16 improves and generalises the best known result for the $(1 + \lambda)$ EA [25, Corollary 24] which requires $p = O(1/n)$ and $\lambda \geq 72 \log n$ 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 16 We estimate the probability of the following failure events in order to apply a union bound later on.

Failure event E_1: all copies of the current search point are affected by noise. By Lemma 15, 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. LEADINGONES$(y) <$ LEADINGONES$(\text{noise}(y)) =$ 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. LEADINGONES$(y) <$ LEADINGONES(x) and LEADINGONES$(\text{noise}(y)) >$ 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 $(1 + \lambda)$ EA cannot decrease. We now show that, conditional on no failure occurring, the expected worst-case number of generations of the $(1 + \lambda)$ 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 Lemma 1.

Now the result follows from applying Theorem 2 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. □

We remark that the condition $q/n \leq 1/n$ for bit-wise noise in Theorem 16 is necessary to bound the probability of failure event E_3. If, say, $p = 1/2$, $q/n = 1/2$, $\lambda = n$ and LEADINGONES$(x) = 1$, for every offspring y, with probability at least $1/(en)$ the first leading one is flipped and then bit-wise noise flips the first two bits in y with probability $p(q/n)^2 = 1/8$. This results in LEADINGONES$(y) <$ LEADINGONES(x) and LEADINGONES$(\text{noise}(y)) >$ LEADINGONES(x). Since there are $\lambda = n$ possible offspring y, $\Pr(E_3) \geq 1 - (1 - 1/(8en))^{\lambda} = \Omega(1)$. Then the Proof of Theorem 16 breaks down as a probability bound of $\Pr(E_3) = O(p/n)$ is required.

5.1 Experiments for LeadingOnes

We also performed experiments to see the threshold behaviour more clearly and to get further insights into the search dynamics in the presence of noise. For instance, our asymptotic results do not reveal implicit constants, including those in exponents, and therefore the exact location of the thresholds is not clear. It is not clear whether different noise models with the same noise parameter p show a similar performance or not. Experiments show the average performance for reasonable problem sizes and to a degree of precision that cannot be obtained from the asymptotic theoretical results in this work (albeit for fixed values of n).

Figure 1 shows the average optimisation times over 1000 runs of the $(1 + \lambda)$ EA on 100-bit LEADINGONES with $\lambda \in \{1, 2, 4, 8, 16\}$ for both one-bit prior noise with probability p and bit-wise prior noise $(1, q/n)$. For both noise models the parameter was varied exponentially: $p \in \{2^{-20}, 2^{-19}, \ldots, 2^{-1}\}$ and $q \in \{2^{-20}, 2^{-19}, \ldots, 2^0\}$. Runs were stopped after $10n^2 = 10^5$ generations or when the optimum was found. For the $(1 + 1)$ EA with one-bit noise we can see that for small noise values like
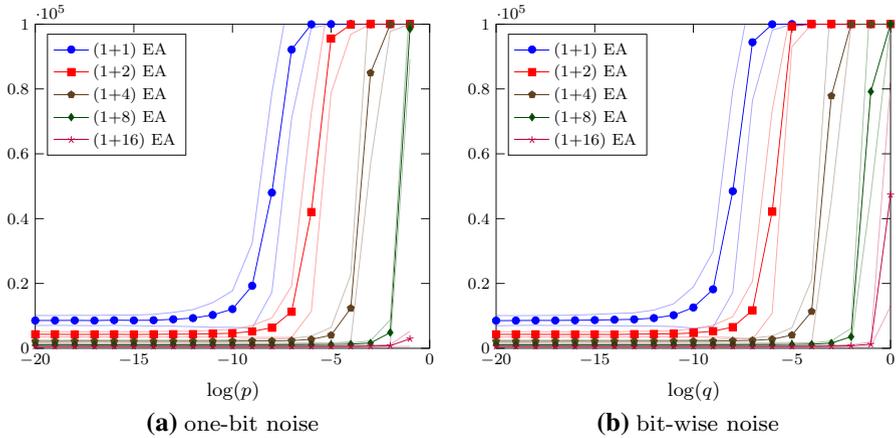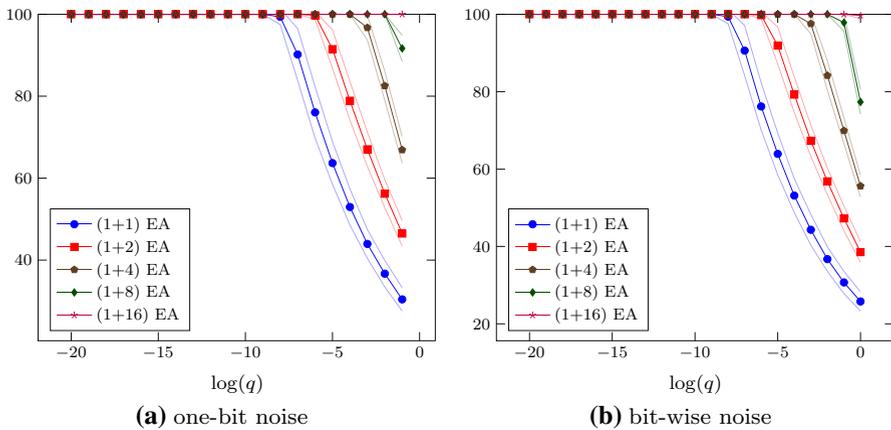
Fig. 1 Average number of generations over 1000 runs for the $(1 + \lambda)$ EA with $\lambda \in \{1, 2, 4, 8, 16\}$ on LEADINGONES ($n = 100$) with one-bit prior noise with probability $p \in \{2^{-20}, 2^{-19}, \ldots, 2^{-1}\}$ and bit-wise prior noise $(1, q/n)$ with $q \in \{2^{-20}, 2^{-19}, \ldots, 2^0\}$. Runs were stopped after $10n^2$ generations. Transparent lines show means $\pm$ standard deviation

$p \in \{2^{-20}, \ldots, 2^{-15}\}$ the averages seem unaffected by the noise parameter, as noise occurs too rarely to have a noticeable effect. When increasing p, the average time increases slightly before shooting up around $p = 2^{-8}$ and hitting the generation limit at $p = 2^{-6}$ in nearly all runs. This clearly shows that and how the expected optimisation time grows exponentially in pn^2 in this regime.

Figure 1 further shows how offspring populations can shift the threshold between efficient and inefficient times towards higher values of p. Even very small offspring population sizes λ have a significant effect. For instance, the $(1 + 8)$ EA is still efficient for $p = 1/4$ and only becomes inefficient for $p = 1/2$. The $(1 + 16)$ EA is efficient even for $p = 1/2$. Note that the curves for all $(1 + \lambda)$ EA s have a very similar shape, independent of λ; they just appear to be shifted towards different values of p. This matches our theoretical results as the exponential term $e^{O(pn/\lambda)}$ contains the ratio p/λ, indicating that the noise strength can be compensated by the offspring population size in a linear fashion.

Comparing plots for one-bit noise and bit-wise noise, the curves look almost identical.

Another interesting performance measure not covered by our theoretical results is to inspect the best fitness found during a run before either finding an optimum or being stopped at $10n^2$ generations. Figure 2 shows averages over these values. For the $(1 + 1)$ EA the best fitness steadily decreases when increasing the noise parameter beyond the threshold for inefficient running times, reaching values of 30.414 for one-bit noise with $p = 1/2$ and 25.781 for bit-wise noise with $q = 1$. For comparison, the average best fitness found during $10n^2 = 10^5$ uniformly random samples was 16.926. Again, we see that offspring populations help by shifting the curves towards higher noise strengths.

Fig. 2 Average best fitness during 1000 runs for the $(1 + \lambda)$ EA with $\lambda \in \{1, 2, 4, 8, 16\}$ on LEADINGONES ($n = 100$) with one-bit prior noise with probability $p \in \{2^{-20}, 2^{-19}, \ldots, 2^{-1}\}$ and bit-wise prior noise $(1, q/n)$ with $q \in \{2^{-20}, 2^{-19}, \ldots, 2^{0}\}$. Runs were stopped after $10n^2$ generations. Transparent lines show means $\pm$ standard deviation

6 An Example Where Noise Helps

The results so far show that on LEADINGONES, noise is disruptive and larger noise values lead to higher expected optimisation times.

The final contribution of this paper is to look at noise from a very different angle. We will 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 [40, 53].

Prügel-Bennett defined such a class of problems known as HURDLE problems [45] 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

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

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 Fig. 3.

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 [39] proved that the $(1 + 1)$ EA has expected time $\Theta(n^w)$ if $2 \leq w \leq n/2$.

Fig. 3 Sketch of a HURDLE function with hurdle width $w = 4$ and problem size $n = 20$
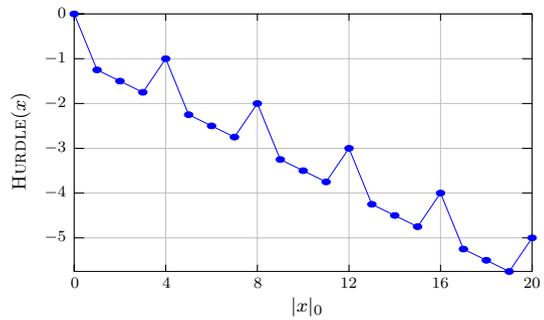
In the following, we consider the well-known algorithm *Randomised Local Search (RLS)*, which works like the $(1 + 1)$ EA, but only flips exactly one bit in each mutation (chosen uniformly at random). We choose RLS instead of the $(1 + 1)$ EA to keep the analyses simple and to make the point that even a very badly performing algorithm can be turned into a highly efficient algorithm through beneficial effects of noise. We will in particular show that RLS under noise is drastically faster than the $(1 + 1)$ EA without noise. Sect. 6.2 will further discuss whether results for RLS under noise can be transferred to the $(1 + 1)$ EA under noise.

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 [39] 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 17 *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 $p = 1/(6n)$ 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 17 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 $p(n - i + 1)/n$) or
2. the parent is misevaluated as having $i - 1$ zeros (probability pi/n).

The probability of the union of these events is

$$\frac{p(n-i+1)}{n} + \frac{pi}{n} - \frac{p^2 i(n-i+1)}{n^2} = p\left(1 + \frac{1}{n} - \frac{pi(n-i+1)}{n^2}\right) \geq p\left(1 + \frac{1}{n} - p\right) \geq p$$

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 pi/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

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

The term $a_i \frac{n^2}{i^2 p(1-p)^2}$ is necessary since on a slope towards a local optimum there is a chance to increase the number of zeros and to possibly return to a worse, previously visited local optimum. The term is largest, $\frac{n^2}{i^2 p(1-p)^2}$, for $i = w - 1 \bmod w$ as from there returning to a local optimum with $i + 1$ zeros is very likely. This needs to be accounted for in our choice of potential function. The term decreases exponentially for decreasing $i \bmod w$ since this risk is reduced as the algorithm moves away from a local optimum.

Note that $g(0) \leq g(1) \leq \cdots \leq g(n)$, 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 $\frac{n}{ip}$ as well as terms

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

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

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

$$= O\left(n \log n + \frac{n}{wp} \sum_{j=1}^{n/w} \frac{1}{j} + \frac{n^2}{w^2 p(1-p)^2} \sum_{j=1}^{n/w} \frac{1}{j^2}\right)$$

$$= O\left(n \log n + \frac{n \log(n/w)}{wp} + \frac{n^2}{w^2 p}\right)$$

$$= O\left(n \log n + \frac{n^2}{w^2 p}\right)$$

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 $a_{i+1} = 2a_i$,

$$E(g(X_t) - g(X_{t+1}) \mid X_t = i, i \bmod w \notin \{0, w-1\})$$

$$\geq \frac{i}{n}(1-p)(g(i) - g(i-1)) - 2p(g(i+1) - g(i))$$

$$= \frac{i}{n}(1-p)\left(\frac{n}{i} + \frac{a_i n^2}{i^2 p(1-p)^2}\right) - 2p\left(\frac{n}{i+1} + \frac{a_{i+1} n^2}{(i+1)^2 p(1-p)^2}\right)$$

$$\geq 1 - p + (1-p)\frac{a_i n}{ip(1-p)^2} - 2p\left(\frac{n}{i} + \frac{2a_i n^2}{i^2 p(1-p)^2}\right)$$

$$= 1 - p - \frac{2pn}{i} + \frac{a_i n}{ip(1-p)^2}\left(1 - p - \frac{4pn}{i}\right).$$

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

$$E(g(X_t) - g(X_{t+1}) \mid X_t = i, i \bmod w \notin \{0, w-1\})$$

$$\geq 1 - p - \frac{2pn}{i} \geq 1 - \frac{1}{6n} - \frac{1}{3} \geq \frac{1}{2}.$$

For $i \bmod w = 0$, the potential is decreased by $g(i) - g(i-1) = \frac{n}{ip}$ with probability at least pi/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

$$E(g(X_t) - g(X_{t+1}) \mid X_t = i, i \bmod w = 0)$$

$$\geq \frac{n}{ip} \cdot \frac{ip}{n} - 2p(g(i+1) - g(i))$$

$$= 1 - 2p(g(i+1) - g(i))$$

$$= 1 - 2p \cdot \left(\frac{n}{i+1} + 2^{-w+2} \cdot \frac{n^2}{(i+1)^2 p(1-p)^2}\right)$$

$$\geq 1 - 2pn - 2^{-w+3} \cdot \frac{n^2}{i^2(1-p)^2}$$

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

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

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

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

For all states $i > 0$, the expected decrease in $g(X_t)$ is at least c for a suitable constant $c > 0$. Once $g(X_t) = 0$ is reached, an optimum is found. Standard additive drift analysis (see, e.g. [33, Theorem 1] for a self-contained statement and proof) then implies that the expected time until $g(X_t) = 0$ is reached is at most $g(n)/c = O(g(n)) = O(n \log n + n^2/(w^2 p))$. $\square$

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 [40, 53]. This effect has been observed in continuous spaces before [52] 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 [51].

To put our result in perspective, we have shown that noise can mitigate a poor choice of algorithm. In our case, an elitist algorithm became a non-elitist algorithm because of noise. This is helpful for HURDLE as here non-elitism is advantageous, while even a small amount of non-elitism is clearly detrimental for LEADINGONES. Note that, as argued in [1, Sect. 4], noise can never improve an *optimal* algorithm for a particular problem. If noise was able to improve the performance of an optimal algorithm, we could simply simulate the effect of noise in the algorithm and obtain a better performing algorithm.

6.1 Experiments

We also provide experiments for HURDLE to see how well the theory predicts the average optimisation time, and to answer questions not covered by Theorem 17.

Figure 4 shows the expected optimisation time of RLS and the $(1+1)$ EA, for HURDLE with $n = 100$ bits and a hurdle width of $w = \lceil 2 \log n \rceil = 14$. Runs were stopped after $n^3 = 10^6$ generations or when the optimum was found. For one-bit noise with noise strength p, the plots show that the algorithm is very efficient in the region $p \in \{2^{-10}, \ldots, 2^{-4}\} \approx \{1/(10n), \ldots, 6.4/n\}$ as predicted by Theorem 17. The
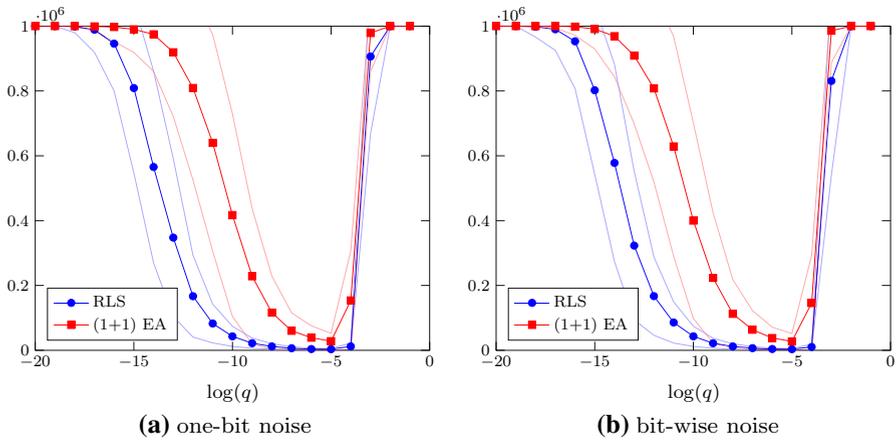
Fig. 4 Average optimisation times during 1000 runs for RLS and the $(1+1)$ EA on HURDLE with $n = 100$ and hurdle width $w = 14$ with one-bit prior noise with probability $p \in \{2^{-20}, 2^{-19}, \dots, 2^{-1}\}$ and bit-wise prior noise $(1, q/n)$ with $q \in \{2^{-20}, 2^{-19}, \dots, 2^{0}\}$. Runs were stopped after 10^6 generations. Transparent lines show means $\pm$ standard deviation

time further seems to increase with $1/p$ as p is decreased, which matches the term $n^2/(pw^2)$ in the running time bound.

We can further see that as p becomes too large, i.e., for $p \geq 2^{-3}$, the average time increases sharply. This matches known results for ONEMAX where $p = \omega((\log n)/n)$ leads to superpolynomial expected times [18].

Figure 4 further shows that the choice of the noise model is insignificant: the results are nearly identical for one-bit prior noise p and bit-wise prior noise $(1, q/n)$ across all values of $p = q$.

6.2 On the Performance of the $(1+1)$ EA

The $(1+1)$ EA shows a similar behaviour to RLS, except that there is a smaller window of efficient parameter ranges. The reader may think that Theorem 17 could also be proven for the $(1+1)$ EA with a more complicated proof that considers all transition probabilities.

However, this is not the case. The problem for the $(1+1)$ EA is that, compared to RLS, it is much more prone to climbing back up into the previous local optimum after making a fitness-decreasing jump towards the optimum. For instance, if $w = O(1)$ then there is always a constant probability of jumping to a local optimum with w zeros from any search point with $1 \leq i < w$ zeros. And the probability of moving close to the optimum is only of order $O(1/n)$, thus the conditional probability of moving closer to the global optimum in a generation where the $(1+1)$ EA either moves closer or jumps to a state with w zeros is still only $O(1/n)$. The algorithm may need to make several such steps in order to arrive at the optimum, and it loses all progress made if a jump back to state w occurs. This problem becomes less and less important as w increases.

Fig. 5 Average optimisation times during 1000 runs for RLS and the (1 + 1) EA on HURDLE with $n = 100$ and hurdle width $w = 6$ with one-bit prior noise with probability $p \in \{2^{-20}, 2^{-19}, \ldots, 2^{-1}\}$. Runs were stopped after 10^6 generations. Transparent lines show means $\pm$ standard deviation. The (1 + 1) EA failed in all runs, except for a single run at $\log(p) = -7$ that succeeded after 519377 generations
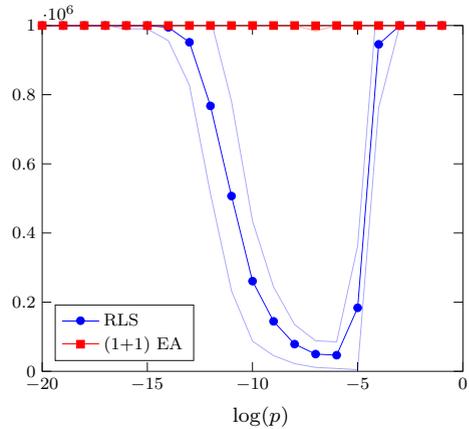
Note that the same fundamental challenge also exists for RLS as it can also move back to the previous local optimum. However, it can only increase the number of zeros by 1 in any step, and if the number of zeros is less than $w - 1 \bmod w$, such a move will decrease the fitness and thus will only be accepted if noise makes the offspring appear competitive to the parent. In Theorem 17 the noise probability p is chosen low enough such that the latter is unlikely.

In the experiments from Fig. 4, the hurdle width $w = 14$ is quite large in relation to the problem size $n = 100$, so that the above issue does not affect performance too much. Decreasing the hurdle width shows a different picture: Fig. 5 shows the performance of both algorithms for a smaller hurdle width of $w = 6$ under one-bit noise.

While RLS is still effective in the regime $p \in \{2^{-10}, \ldots, 2^{-5}\}$ (even though the hurdle width is lower than required by Theorem 17), the (1 + 1) EA failed in all runs, except for a single run at $\log(p) = -7$ that succeeded after 519,377 generations.

This indicates why Theorem 17 had to be limited to RLS. As an aside, we have obtained a rare case where the performance of the (1 + 1) EA is drastically worse than that of RLS. So far, only very artificial examples were known [13] and some of them, examples of *monotone functions*, needed a significantly higher mutation rate [15, 34].

6.3 Offspring Populations are Harmful for Hurdle

Finally, we consider the role of offspring populations on HURDLE, defining the $(1 + \lambda)$ RLS as a variant of the $(1 + \lambda)$ EA where mutation flips exactly one bit. For consistency we refer to RLS as $(1 + 1)$ RLS.

The proof of Theorem 17 relies on the fact that a fitness-decreasing step leaving a local optimum towards the global optimum is accepted because of noise. While this effect was helpful on LEADINGONES, it is detrimental for HURDLE. This is shown empirically in Fig. 6.
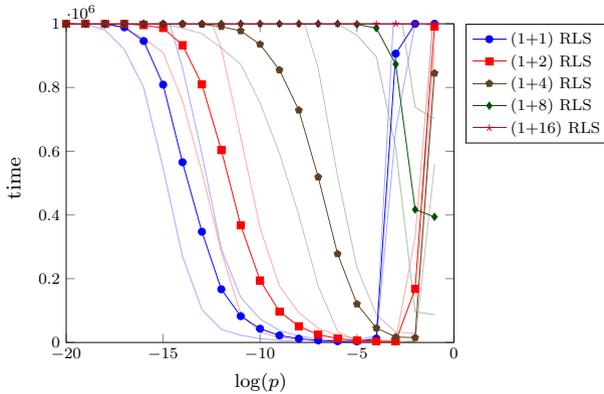
Fig. 6 Average optimisation times during 1000 runs for $(1 + \lambda)$ RLS on HURDLE with $n = 100$ and hurdle width $w = 14$ with one-bit prior noise with probability $p \in \{2^{-20}, 2^{-19}, \dots, 2^{-1}\}$. Runs were stopped after 10^6 generations. Transparent lines show means $\pm$ standard deviation

An increased offspring population shifts the curves towards higher noise parameters, while maintaining the unimodal shape of the curve, with steep increases for too large values. This shift is very similar to the one observed for the $(1 + \lambda)$ EA on LEADINGONES.

For instance, for $p \in \{2^{-10}, \dots, 2^{-4}\}$ where $(1 + 1)$ RLS is efficient, the $(1 + 8)$ RLS fails to find the optimum before time runs out in almost all runs, and the $(1 + 16)$ RLS only found the optimum in a single run at $p = 0.5$, with a time of 795,151 generations. We conclude that, in this context, offspring populations can be harmful.

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 [5, 48] 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 $\Theta(n^2) \cdot \exp(\Theta(\min\{pn^2, n\}))$ for one-bit noise $p \leq 1/2$, asymmetric one-bit noise with $p = O(1/n)$, and bit-wise noise $(p', q/n)$ where $q/n \leq 1/2$ and $p = p' \min\{q, 1\}$. This confirms that the threshold between polynomial and super-polynomial expected times is $p = \Theta((\log n)/n^2)$ and $p = \Omega(1/n)$ leads to exponential expected times.

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 $O\left(n^2 \cdot e^{O(pn/\lambda)}\right)$, 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. Experiments complemented our theoretical results and also showed that RLS under noise outperforms the $(1 + 1)$ EA both with and without noise. Experiments further showed that on Hurdle, in stark contrast to LeadingOnes, offspring populations in RLS can be harmful as here they reduce the beneficial effects of noise.

Open problems for future work include showing a lower bound for the expected optimisation time of the $(1 + \lambda)$ EA on LeadingOnes, and obtaining tighter results on the performance of evolutionary algorithms with parent populations, i.e., the $(\mu + 1)$ EA, on LeadingOnes and other problems.

References

1. Akimoto, Y., Astete-Morales, S., Teytaud, O.: Analysis of runtime of optimization algorithms for noisy functions over discrete codomains. Theor. Comput. Sci. **605**, 42–50 (2015)
2. Badkobeh, G., Lehre, P.K., Sudholt, D.: Black-box complexity of parallel search with distributed populations. In: Proceedings of Foundations of Genetic Algorithms (FOGA'15), pp. 3–15. ACM Press, New York (2015)
3. Baswana, S., Biswas, S., Doerr, B., Friedrich, T., Kurur, P.P., Neumann, F.: Computing single source shortest paths using single-objective fitness functions. In: Proceedings of FOGA'09, pp. 59–66. ACM Press, New York (2009)
4. Beyer, H.-G.: Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. Comput. Methods Appl. Mech. Eng. **186**(2), 239–267 (2000)
5. Bian, C., Qian, C., Tang, K.: Towards a running time analysis of the $(1 + 1)$-EA for OneMax and LeadingOnes under general bit-wise noise. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) Parallel Problem Solving from Nature—PPSN XV, pp. 165–177. Springer, Cham (2018)
6. Bianchi, L., Dorigo, M., Gambardella, L.M., Gutjahr, W.J.: A survey on metaheuristics for stochastic combinatorial optimization. Nat. Comput. **8**(2), 239–287 (2009)
7. Covantes Osuna, E., Gao, W., Neumann, F., Sudholt, D.: Speeding up evolutionary multi-objective optimisation through diversity-based parent selection. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'17), pp. 553–560. ACM, New York (2017)
8. Cutello, V., Nicosia, G., Pavone, M.: An immune algorithm with stochastic aging and kullback entropy for the chromatic number problem. J. Comb. Optim. **14**, 9–33 (2007)

9. Dang, D.-C., Lehre, P.K.: Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In: Proceedings of Foundations of Genetic Algorithms (FOGA'15), pp. 62–68. ACM, New York (2015)

10. Dang, D.-C., Lehre, P.K.: Runtime analysis of non-elitist populations: from classical optimisation to partial information. Algorithmica **75**(3), 428–461 (2016)

11. Dang-Nhu, R., Dardinier, T., Doerr, B., Izacard, G., Nogneng, D.: A new analysis method for evolutionary optimization of dynamic and noisy objective functions. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18), pp. 1467–1474. ACM, New York (2018)

12. Doerr, B., Gnewuch, M., Hebbinghaus, N., Neumann, F.: A rigorous view on neutrality. In: Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC'07), pp. 2591–2597 (2007)

13. Doerr, B., Jansen, T., Klein, C.: Comparing global and local mutations on bit strings. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08), pp. 929–936. ACM, New York (2008)

14. Doerr, B., Hota, A., Kötzing, T.: Ants easily solve stochastic shortest path problems. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'12), pp. 17–24 (2012)

15. Doerr, B., Jansen, T., Sudholt, D., Winzen, C., Zarges, C.: Mutation rate matters even when optimizing monotonic functions. Evol. Comput. **21**(1), 1–21 (2013)

16. Doerr, B., Kodric, B., Voigt, M.: Lower bounds for the runtime of a global multi-objective evolutionary algorithm. In: Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC'13), pp. 432–439 (2013)

17. Doerr, B., Sudholt, D., Witt, C.: When do evolutionary algorithms optimize separable functions in parallel? In: Proceedings of Foundations of Genetic Algorithms (FOGA'13), pp. 51–64. ACM, New York (2013)

18. Droste, S.: Analysis of the $(1+1)$ EA for a noisy OneMax. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), pp. 1088–1099. Springer, Berlin (2004)

19. Droste, S., Jansen, T., Wegener, I.: On the analysis of the $(1+1)$ evolutionary algorithm. Theor. Comput. Sci. **276**(1–2), 51–81 (2002)

20. Feldmann, M., Kötzing, T.: Optimizing expected path lengths with ant colony optimization using fitness proportional update. In: Proceedings of Foundations of Genetic Algorithms (FOGA'13), pp. 65–74. ACM, New York (2013)

21. Friedrich, T., Kötzing, T., Krejca, M.S., Sutton, A.M.: Robustness of ant colony optimization to noise. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'15), pp. 17–24. ACM, New York (2015)

22. Friedrich, T., Kötzing, T., Krejca, M.S., Sutton, A.M.: The compact genetic algorithm is efficient under extreme Gaussian noise. IEEE Trans. Evol. Comput. **21**(3), 477–490 (2017)

23. Giel, O.: Expected runtimes of a simple multi-objective evolutionary algorithm. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03), vol. 3, pp. 1918–1925 (2003)

24. Giel, O., Lehre, P.K.: On the effect of populations in evolutionary multi-objective optimisation. Evol. Comput. **18**(3), 335–356 (2010)

25. Gießen, C., Kötzing, T.: Robustness of populations in stochastic environments. Algorithmica **75**(3), 462–489 (2016)

26. Jägersküpper, J., Storch, T.: When the plus strategy outperforms the comma strategy and when not. In: Proceedings of the IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007, pp. 25–32. IEEE, New York (2007)

27. Jansen, T., Sudholt, D.: Analysis of an asymmetric mutation operator. Evol. Comput. **18**(1), 1–26 (2010)

28. Jansen, T., De Jong, K.A., Wegener, I.: On the choice of the offspring population size in evolutionary algorithms. Evol. Comput. **13**, 413–440 (2005)

29. Jebalia, M., Auger, A., Hansen, N.: Log-linear convergence and divergence of the scale-invariant $(1+1)$-ES in noisy environments. Algorithmica **59**(3), 425–460 (2011)

30. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments—a survey. IEEE Trans. Evol. Comput. **9**(3), 303–317 (2005)

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

32. Laumanns, M., Thiele, L., Zitzler, E.: Running time analysis of multiobjective evolutionary algorithms on pseudo-boolean functions. IEEE Trans. Evol. Comput. **8**(2), 170–182 (2004)

33. Lehre, P.K., Witt, C.: General drift analysis with tail bounds (2013). CoRR. arXiv:1307.2559

34. Lengler, J., Steger, A.: Drift analysis and evolutionary algorithms revisited. Comb. Probab. Comput. **27**(4), 643–666 (2018)

35. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society, New York (2008)

36. Meyer-Nieberg, S., Beyer, H.-G.: Why noise may be good: additive noise on the sharp ridge. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08), pp. 511–518. ACM, New York (2008)

37. Mitzenmacher, M., Upfal, E.: Probability and Computing. Cambridge University Press, Cambridge (2005)

38. Nguyen, A.Q., Sutton, A.M., Neumann, F.: Population size matters: rigorous runtime results for maximizing the hypervolume indicator. Theor. Comput. Sci. **561**, 24–36 (2015)

39. Nguyen, P.T.H., Sudholt, D.: Memetic algorithms beat evolutionary algorithms on the class of hurdle problems. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18), pp. 1071–1078. ACM, New York (2018)

40. Ochoa, G., Veerapen, N.: Deconstructing the big valley search space hypothesis. In: Proceedings of Evolutionary Computation in Combinatorial Optimization (EvoCOP 2016), pp. 58–73. Springer, New York (2016)

41. Oliveto, P.S., Sudholt, D.: On the runtime analysis of stochastic ageing mechanisms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014), pp. 113–120. ACM Press, New York (2014)

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

43. Oliveto, P.S., Witt, C.: Erratum: simplified drift analysis for proving lower bounds in evolutionary computation. ArXiv e-prints (2012)

44. Paixão, T., Pérez Heredia, J., Sudholt, D., Trubenová, B.: Towards a runtime comparison of natural and artificial evolution. Algorithmica **78**(2), 681–713 (2017)

45. Prügel-Bennett, A.: When a genetic algorithm outperforms hill-climbing. Theor. Comput. Sci. **320**(1), 135–153 (2004)

46. Prugel-Bennett, A., Rowe, J., Shapiro, J.: Run-time analysis of population-based evolutionary algorithm in noisy environments. In: Proceedings of Foundations of Genetic Algorithms (FOGA'15), pp. 69–75. ACM, New York (2015)

47. Qian, C., Yu, Y., Zhou, Z.-H.: An analysis on recombination in multi-objective evolutionary optimization. Artif. Intell. **204**, 99–119 (2013)

48. Qian, C., Bian, C., Jiang, W., Tang, K.: Running time analysis of the $(1 + 1)$-EA for OneMax and LeadingOnes under bit-wise noise. Algorithmica **81**, 749–795 (2018)

49. Qian, C., Bian, C., Yu, Y., Tang, K., Yao, X.: Analysis of noisy evolutionary optimization when sampling fails. In: Proceedings of the 20th ACM Conference on Genetic and Evolutionary Computation (GECCO'18), pp. 1507–1514 (2018)

50. Qian, C., Yu, Y., Tang, K., Jin, Y., Yao, X., Zhou, Z.-H.: On the effectiveness of sampling for evolutionary optimization in noisy environments. Evol. Comput. **26**(2), 237–267 (2018)

51. Qian, C., Yu, Y., Zhou, Z.-H.: Analyzing evolutionary optimization in noisy environments. Evol. Comput. **26**(1), 1–41 (2018)

52. Rana, S., Whitley, L.D., Cogswell, R.: Searching in the presence of noise. In: Proceedings of PPSN IV, pp. 198–207. Springer, Berlin (1996)

53. Reeves, C.R.: Landscapes, operators and heuristic search. Ann. Oper. Res. **86**, 473–490 (1999)

54. Rowe, J.E., Sudholt, D.: The choice of the offspring population size in the $(1, \lambda)$ evolutionary algorithm. Theor. Comput. Sci. **545**, 20–38 (2014)

55. Sudholt, D.: 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), pp. 1523–1530. ACM, New York (2018)

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