More Precise Runtime Analyses of Non-elitist Evolutionary Algorithms in Uncertain Environments

Per Kristian Lehre[1] · Xiaoyu Qin[1]

Abstract

Real-world applications often involve "uncertain" objectives, i.e., where optimisation algorithms observe objective values as a random variables with positive variance. In the past decade, several rigorous analysis results for evolutionary algorithms (EAs) on discrete problems show that EAs can cope with low-level uncertainties, i.e. when the variance of the uncertain objective value is small, and sometimes even benefit from uncertainty. Previous work showed that a large population combined with a non-elitist selection mechanism is a promising approach to handle high levels of uncertainty. However, the population size and the mutation rate can dramatically impact the performance of non-elitist EAs, and the optimal choices of these parameters depend on the level of uncertainty in the objective function. The performance and the required parameter settings for non-elitist EAs in some common objective-uncertainty scenarios are still unknown. We analyse the runtime of non-elitist EAs on two classical benchmark problems ONEMAX and LEADINGONES in in the one-bit, the bitwise, the Gaussian, and the symmetric noise models, and the dynamic binary value problem (DYNBV). Our analyses are more extensive and precise than previous analyses of non-elitist EAs. In several settings, we prove that the non-elitist EAs outperform the current state-of-the-art results. Furthermore, we provide more precise guidance on how to choose the mutation rate, the selective pressure, and the population size as a function of the level of uncertainty.

Per Kristian Lehre and Xiaoyu Qin are listed in alphabetical order.

✉ Xiaoyu Qin
   xxq896@cs.bham.ac.uk

   Per Kristian Lehre
   p.k.lehre@cs.bham.ac.uk

[1]   School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK

1 Introduction

Evolutionary Algorithms (EAs) are widely applied to solve industrial optimisation problems [2–8]. EAs are applicable in black-box optimisation problems in which a fitness function is optimised by only evaluating the fitness of search points [9–11]. The algorithm has only access to the fitness function via a fitness value oracle. Practical examples of black-box optimisation include tuning of hyper-parameters of machine learning systems and optimising air foils in simulation [12]. In these scenarios, it is hard to formulate the relationship between solutions and fitness values, and fitness function evaluations are often expensive. Therefore, the cost (or runtime) of black-box optimisation algorithms is defined as the number of queries to the oracle until an optimal solution is obtained.

The exact objective value of a search point is often impossible to obtain, or a fitness function could change over time in practice. Therefore, a wide range of uncertainties have to be considered in real-world optimisation problems [13]. Most studies of how EAs respond to uncertainties are empirical [13–15]. In the theory community, *runtime analysis* is the main approach to theoretically evaluate the performance of EAs, which mathematically estimates the expected number of evaluations of the objective function until an optimum is found. In uncertain environments, runtime analyses of EAs have considered four abstract uncertainty models which are prior noise [16–22], posterior noise [16, 19, 21–26], dynamic environment [27–31] and partial evaluation [32, 33]. In pseudo-Boolean optimisation, the prior noise randomly flips one or several bits in the search point before each evaluation, e.g. one-bit noise [34], which flips one bit uniformly with probability q, and bit-wise noise [17], which flips each bit independently with probability p, while the posterior noise makes some changes on the fitness value after each evaluation, e.g. Gaussian noise [16], which adds a value sampled from a Gaussian distribution with zero mean and σ^2 variance, i.e., $\mathcal{N}\left(0, \sigma^2\right)$. In dynamic optimisation, the fitness function is fixed in each generation but is varied by time, e.g. the noisy linear function [30] and the DYNBV problem [15].

The simple (1+1) EA is robust to some uncertainties but can also be inefficient under high-level uncertainties. In noisy environments, the runtime of the (1+1) EA on LEADINGONES is exponential if noise levels are $q = \Omega(1)$ and $p = \Omega\left(1/n^2\right)$ under one-bit noise and bit-wise noise, respectively [16–18]. Some algorithms can furthermore improve the robustness, such as estimation of distribution algorithms (EDAs) [23], ant colony optimisation (ACO) [19], the (1+1) EA using a resampling strategy [17, 21], the voting algorithm [26] and population-based algorithms [16, 22, 32, 35]. A common method to cope with high levels of uncertainty is using such a resampling strategy which averages the value of many uncertain evaluations [17, 21]. The (1+1) EA using a resampling strategy solves ONEMAX and LEADINGONES in higher-level one-bit, bit-wise and Gaussian noise models in expected polynomial time [17, 21]. However, the (1+1) EA fails in the symmetric noise model, in which the fitness value of a solution x is $C - \text{OM}(x)$ with probability $1/2$, otherwise $\text{OM}(x)$ [24] (The

definition of the function OM(x) can be found in Sect. 2.1.), where $C \in \mathbb{R}$. A solution to deal with the symmetric noise model can be to use a population. Qian et al. proved that the (μ+1) EA and the (1+λ) EA can solve the ONEMAX problem in the symmetric noise model in expected polynomial time [24]. For the efficiency of non-elitist EAs under uncertainties, Dang and Lehre [22] showed that binary tournament selection with a sufficient population size and a conservative mutation rate has expected runtime $O(n \log(n) \log(\log(n)))$ on ONEMAX under any one-bit noise level. They also proved that the non-elitist EA can handle extremely high-levels Gaussian noise [22] and partial evaluation optimisation [32]. However, the robustness of non-elitist EAs to noise is still unknown in several settings. For the robustness in the dynamic environments, Lengler and Schaller [30] proved that the (1+1) EA can optimise the random weights linear function, in which the positive weights in each generation are randomly sampled independently from some distribution, in expected $O(n \log(n))$ time. The efficiency of population-based algorithms in this setting is currently unknown, though Lengler and Riedi [31] proved that the runtime of the ($\mu + 1$) EA on the DYNBV is $O(n \log(n))$ by assuming that the population is initialised close to the optimum.

Tables 1, 2, 3, 4, 5, 6, 7 and 8 summarise recent theoretical studies (including this paper) of EAs in noisy settings. Tables 1 and 2 show results for the one-bit noise model (q) on ONEMAX and LEADINGONES, respectively. Table 3 and 4 show results for the bit-wise noise model (p) on ONEMAX and LEADINGONES, respectively. Table 5 and 6 show results for the Gaussian noise model (σ) on ONEMAX and LEADINGONES, respectively. Tables 7 and 8 show results for the symmetric noise model (C, q) on ONEMAX and LEADINGONES, respectively. Note that some previous studies do not contain exact runtimes. In these cases, the runtime results are deduced from proofs. For convenience, we call the non-elitist EA with 2-tournament selection as 2-tournament EA.

In this paper, we theoretically analyse the runtime of non-elitist EAs with 2-tournament and (μ, λ) selection in several uncertain settings. For the 2-tournament EA, we first use the level-based theorem [36] to derive a general theorem in uncertain environments. Then we apply this general theorem to obtain upper bounds of runtimes on ONEMAX and LEADINGONES in prior and posterior noise models, i.e., one-bit, bit-wise, Gaussian and symmetric noise models. In noisy settings, our analyses are more extensive and precise than previous analyses of non-elitist EAs [22]. We provide more precise guidance on how to choose the mutation rate and the population size as a function of the level of uncertainty. We also use the negative drift theorem for populations [37] to show that a too high mutation rate relative to the noise level leads to an exponentially low probability to find the optimum within exponential time. For the (μ, λ) EA, we analyse the runtime under symmetric noise for the first time. Similarly, we provide guidance on how to choose the mutation rate, the selective pressure and the population size as a function of the noise level. We also show that too low selective pressure, i.e., λ/μ, and too high mutation rate according to the noise level lead to inefficient optimisation. Overall, in several noisy settings, we prove that non-elitist EAs outperform the current state of the art results (see Tables 1, 2, 3, 4, 5, 6, 7 and 8). Note that the runtime for a certain level of noise can be obtained by plugging the noise level and the appropriate parameters into the general results in these tables. Finally,

Table 1 Theoretical results of evolutionary algorithms on ONEMAX in the one-bit noise model (q)

Algorithm	Parameter settings	Noise level q	Expected runtime
(1+1) EA	$\chi/n = 1/n$	$O(1/n)$	$\Theta(n \log(n))$ [16]
	$\chi/n = 1/n$	$O(\log(n)/n)$	Poly(n) [16]
	$\chi/n = 1/n$	$\omega(\log(n)/n)$	$2^{\omega(\log(n))}$ [16]
(1+1) EA (resampling)	$\chi/n = 1/n$; $m = 4n^3$	[0, 1]	Poly(n) [17]
(μ+1) EA	$\chi/n = 1/n$; $\mu > 12 \log(15n)/q$	(0, 1]	$O(\mu n \log(n))$ [16]
(1+λ) EA	$\chi/n = 1/n$; $\lambda \geq \max\{12/p, 24\}n \log n$	(0, 1]	$O\left(n^2 \lambda\right)$ [16]
ACO-fp	$\rho = o\left(1/\left(n^3 \log n\right)\right)$	[0, 1]	$O\left(n^2 \log(n)/\rho\right)$ [19]
2-Tournament EA	$\chi/n \in (0, \ln(1 + 2\theta)/n)$; $\lambda \in \Omega(\log(n/\chi))$	[0, 1]	$O\left(\lambda n \log(1/\chi) + n \log(n)/\chi\right)$
	$\theta = 1/2 - (q/2)(1 - q/2) - q/2n_0$,		(Theorem 4)
	for any constant $n_0 \geq 3$		

Table 2 Theoretical results of evolutionary algorithms on LEADINGONES in the one-bit noise model (q)

Algorithm	Parameter settings	Noise level q	Expected runtime
(1+1) EA	$\chi/n = 1/n$	$[0, 1/2]$	$\Theta\left(n^2\right) e^{\Theta\left(\min\left\{qn^2, n\right\}\right)}$ [18]
(1+1) EA (resampling)	$\chi/n = 1/n$; $m = 4n^4 \log(n)/15$	$[0, 1]$	$O(mn^8)$ [17]
(1+λ) EA	$\chi/n = 1/n$; $\lambda \geq 3.42 \log(n)$, $\lambda \in O(n)$	$[0, 1/2]$	$O\left(n^2 e^{O(n/\lambda)}\right)$ [18]
UMDA	$\mu \geq c \log(n)$; $\lambda > 4e(1+\delta)\mu$ for constants c, δ;	$[0, 1)$	$O(n\lambda \log(\lambda) + n^2)$ [38]
2-Tournament EA	$\chi/n \in (0, \ln(1 + 2\theta)/n)$; $\lambda \in \Omega(\log(n/\chi))$	$[0, 1)$	$O\left(n\lambda \log(n/\chi) + n^2/\chi\right)$
	$\theta = 1/2 - q(1 - q/2)$		(Theorem 5)

Table 3 Theoretical results of evolutionary algorithms on ONEMAX in the bit-wise noise model (p)

Algorithm	Parameter settings	Noise level p	Expected runtime
(1+1) EA	$\chi/n = 1/n$	$O(1/n^2)$	$\Theta(n\log(n))$ [16]
	$\chi/n = 1/n$	$O(\log(n)/n^2)$	Poly(n) [16]
	$\chi/n = 1/n$	$\omega(\log(n)/n^2)$	$2^{\omega(\log(n))}$ [16]
(1+1) EA (resampling)	$\chi/n = 1/n$; $m = n^{3+2c}/4$	$p = 1/2 - 1/n^c$ for $0 < c = \Theta(1)$	Poly(n) [17]
2-Tournament EA	$\chi/n \in (0, \ln(1+2\theta)/n)$ $\theta = 9(1/2 - p)/(64\sqrt{2pn} + 16)$ $\lambda \in \Omega\left(\frac{1+pn}{(1-2p)^2}\log\left(\frac{n}{(1-2p)\chi}\right)\right)$	$(0, 1/2)$	$O\left(\frac{n(1+pn)}{(1-2p)^2}\left(\lambda\log\left(\frac{1}{\chi}\right) + \frac{\log(n)}{\chi}\right)\right)$ (Theorem 6)

Table 4 Theoretical results of evolutionary algorithms on LEADINGONES in the bit-wise noise model (p)

Algorithm	Parameter settings	Noise level p	Expected runtime
(1+1) EA	$\chi/n = 1/n$	$[0, 1/(2n)]$	$\Theta\left(n^2\right)e^{\Theta\left(\min\left[pn^3, n\right]\right)}$ [18]
(1+1) EA (resampling)	$\chi/n = 1/n; m = 36n^{2c}+4$	$p = c\log(n)/n$ for $0 < c = \Theta(1)$	$12m \cdot n^{30c+1}$ [17]
	$\chi/n = 1/n; m \in O(\text{poly}(n))$	$\omega(\log(n))/n$	$e^{\Omega(n)}$ [17]
2-Tournament EA	$\chi/n \in (0, \ln(1 + 2\theta)/n);$	$[0, 1/3]$	$O\left(\frac{ne^{6np}}{(1-3p)^2}\left(\lambda \log\left(\frac{n}{\chi}\right) + \frac{n}{\chi}\right)\right)$
	$\lambda \in \Omega\left(\frac{e^{6np}}{(1-3p)^2}\log\left(\frac{n}{\chi}\right)\right)$		(Theorem 7)
	$\theta = (1/2 - p)\,e^{-3np}$		

Table 5 Theoretical results of evolutionary algorithms on ONEMAX in the Gaussian noise model (σ^2)

Algorithm	Parameter settings	Noise level σ	Expected runtime
(1+1) EA	$\chi/n = 1/n$	$\sigma^2 \leq 1/(4\log(n))$	$O(n\log(n))$ [16]
	$\chi/n = 1/n$	$\sigma^2 \geq 1$	$e^{\Omega(n)}$ [21]
(1+1) EA (resampling)	$\chi/n = 1/n; m = \lceil \frac{n\sigma^2}{\log(n)} \rceil$	$1 \leq \sigma^2 \in \text{Poly}(n)$	$\text{Poly}(n)$ [21]
$(\mu + 1)$ EA	$\chi/n = 1/n; \mu \in \text{poly}(n)$	$\sigma \geq n^3$	$2^{\omega(\log(n))}$ [25]
cGA	$K = \omega\left(\sigma^2\sqrt{n}\log(n)\right)$	$\sigma^2 > 0$	$O\left(K\sigma^2\sqrt{n}\log(Kn)\right)$ [23]
ACO-fp	$\rho = o\left(1/\left(n(n+\sigma\log n)^2\log n\right)\right)$	$\sigma^2 \geq 0$	$O\left(n^2\log(n)/\rho\right)$ [19]
Voting Algorithm	-	$\sigma^2 \geq 3n/8$	$O\left(\sigma^2\log(n)\right)$ [26]
2-Tournament EA	$\chi/n \in (0, \ln(1+2\theta)/n); \lambda \in \Omega(\sigma^2\log(n/\chi))$ $\theta = 1/(6+48\sigma/\pi)$	$\sigma^2 > 0$	$O(\sigma^2\lambda n\log(1/\chi)+$ $\sigma^2 n\log(n)/\chi)$ (Theorem 8)

Table 6 Theoretical results of evolutionary algorithms on LEADINGONES in the Gaussian noise model (σ^2)

Algorithm	Parameter settings	Noise level σ	Expected runtime
(1+1) EA	$\chi/n = 1/n$	$\sigma^2 \leq 1/\left(12en^2\right)$	$O\left(n^2\right)$ [16]
	$\chi/n = 1/n$	$\sigma^2 \geq n^2$	$\Omega\left(e^n\right)$ [21]
(1+1) EA (resampling)	$\chi/n = 1/n; m = \lceil 12en^2\sigma^2 \rceil$	$n^2 \leq \sigma^2 \in \mathrm{poly}(n)$	$O(mn^2)$ [21]
2-Tournament EA	$\chi/n \in (0, \ln(1+2\theta)/n); \lambda \in \Omega(\sigma^2\log(n/\chi))$	$\sigma^2 > 0$	$O\left(\sigma^2\lambda n\log(n/\chi) + \sigma^2 n^2/\chi\right)$
	$\theta = 1/(6 + 48\sigma/\pi)$		(Theorem 8)

Table 7 Theoretical results of evolutionary algorithms on OneMax in the symmetric noise model (C, q)

Algorithm	Parameter settings	Noise parameters	Expected runtime
(1+1) EA (resampling)	$\chi/n = 1/n$; Any $m \geq 1$	Any $C \in \mathbb{R}$; $q = 1/2$	$e^{\Omega(n)}$ [24]
$(\mu + 1)$ EA	$\chi/n = 1/n$; $\mu = 3 \log(n)$	$C = 2n$; $q = 1/2$	$O\left(n \log^3(n)\right)$ [24]
$(1 + \lambda)$ EA	$\chi/n = 1/n$; $\lambda = 8 \log(n)$	$C = 0$; $q = 1/2$	$O\left(n \log^2(n)\right)$ [24]
	$\chi/n = 1/n$; $\lambda \leq \log(n)/10$	$C = 0$; $q = 1/2$	$e^{\Omega(n)}$ [24]
2-Tournament EA	$\chi/n \in (0, \ln(1 + 2\theta)/n)$; $\lambda \in \Omega(\log(n/\chi))$, $\theta = 1/2 - q$	$C \in \mathbb{R}$; $q \in [0, 1/2)$	$O(\lambda n \log(1/\chi) + n \log(n)/\chi)$ (Theorem 9)
	$\chi/n > \ln(1 + 2\theta + o(1))/n$; $\lambda \in \text{poly}(n)$, $\theta = 1/2 - q$	$C \in \mathbb{R}$; $q \in [0, 1/2)$	$e^{\Omega(n)}$ (Theorem 11)
(μ, λ) EA	$\lambda \in \Omega(\log(n))$; $(1 - q)\zeta\lambda > \mu \in \Omega(\log(n))$, $\chi/n \in \left(0, \ln\left(\frac{(1-q)\lambda}{(1+\delta)\mu}\right)/n\right)$ and $\chi \in \Omega(1)$ for any constant $\zeta, \delta \in (0, 1)$	$C \leq 0$; $q \in [0, 1)$	$O(\lambda n + n \log(n))$ (Theorem 10)
	$\lambda, \mu \in \log(n)$; $(1 - q)\lambda > \mu$, $\chi/n > \ln\left(\frac{(1-q)\lambda}{\mu}\right)/n$	$C \leq 0$; $q \in [0, 1]$	$e^{\Omega(n)}$ (Theorem 12)

Table 8 Theoretical results of evolutionary algorithms on LEADINGONES in the symmetric noise model (C, q)

Algorithm	Parameter settings	Noise parameters	Expected runtime
2-Tournament EA	$\chi/n \in (0, \ln(1 + 2\theta)/n); \lambda \in \Omega(\log(n/\chi))$ $\theta = 1/2 - q$	$C \in \mathbb{R}; q \in [0, 1/2)$	$O\left(n\lambda \log(n/\chi) + n^2/\chi\right)$ (Theorem 9)
	$\chi/n > \ln(1 + 2\theta + o(1))/n; \lambda \in \text{poly}(n)$ $\theta = 1/2 - q$	$C \in \mathbb{R}; q \in [0, 1/2)$	$e^{\Omega(n)}$ (Theorem 11)
(μ, λ) EA	$\lambda \in \Omega(\log(n)); (1 - q)\zeta\lambda > \mu \in \Omega(\log(n))$ $\chi/n \in \left(0, \ln\left(\frac{(1-q)\lambda}{(1+\delta)\mu}\right)/n\right)$ and $\chi \in \Omega(1)$ for any constant $\zeta, \delta \in (0, 1);$	$C \le 0; q \in [0, 1)$	$O\left(n\lambda \log(n) + n^2\right)$ (Theorem 10)
	$\lambda, \mu \in \log(n); (1 - q)\lambda > \mu$ $\chi/n > \ln\left(\frac{(1-q)\lambda}{\mu}\right)/n$	$C \le 0; q \in [0, 1]$	$e^{\Omega(n)}$ (Theorem 12)

we prove for the first time that non-elitist EAs can optimise the DYNBV problem in expected polynomial time.

This paper extends our preliminary work [1]. We extend analyses to the symmetric noise model (C, q) and the (μ, λ) EA. We also add analyses to show what mutation rate makes optimisation inefficient in the symmetric noise model.

The paper is structured as follows. Section 2 introduces the studied algorithms, the analysed uncertainty models and mathematical tools used in the paper. Section 3 provides a general theorem for analysing non-elitist EAs with 2-tournament selection in uncertain environments. In Sect. 4, we consider four noise models. We prove that the expected runtime of the 2-tournament EA on ONEMAX and LEADINGONES under one-bit, bit-wise and Gaussian noise are polynomial for appropriate parameter settings in Sects. 4.1–4.3, respectively. In Sect. 4.4, we show that non-elitist EAs with 2-tournament and (μ, λ) selection can find the optimum of ONEMAX and LEADINGONES under symmetric noise in expected polynomial time if using appropriate parameter settings, otherwise runtimes are exponential. Section 5 then shows the runtime analysis for the 2-tournament EA on the DYNBV function. Finally, Sect. 6 concludes the paper.

2 Preliminaries

We consider a non-elitist EA with binary tournament and (μ, λ) selection optimising four noisy versions and one dynamic version of pseudo-Boolean functions. We first define some notations which are used later. For any integer $n > 0$, we define $[n] := \{1, \ldots, n\}$ and $[0...n] := \{0\} \cup [n]$. We use $\mathrm{H}(\cdot, \cdot)$ to denote the Hamming distance. The natural logarithm is denoted by $\ln(\cdot)$, and the logarithm to the base 2 is denoted by $\log(\cdot)$. Let $f : \mathcal{X} \to \mathbb{R}$ be any pseudo-Boolean function, where $\mathcal{X} = \{0, 1\}^n$ is the set of bitstrings of length n.

2.1 Noise and Dynamic Fitness Models

We consider two well-known pseudo-Boolean functions ONEMAX and LEADINGONES which are defined as $\mathrm{OM}(x) := \sum_{i=1}^{n} x_i$ and $\mathrm{LO}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$, respectively. In this paper, we consider four noise models and the DYNBV problem, which are defined as follows.

For noisy optimisation, we let $f^n(x)$ denote the noisy fitness value, and $f(x)$ the fitness without noise. The one-bit noise model (q) [16–19, 34, 38] is the easiest starting point for theoretical analysis, which can be described as: given a probability $q \in [0, 1]$, i.e., noise level, and a solution $x \in \{0, 1\}^n$, then

$$f^n(x) = \begin{cases} f(x) & \text{with probability } 1 - q \\ f(x') & \text{with probability } q \end{cases}$$

where x' is a uniformly sampled Hamming neighbour of x.

In real-world problems, noise can affect a stochastic number of bits rather than at most one bit. The bit-wise model (p) [16–18] can more closely imitate reality: given

a probability $p \in [0, 1]$ and a solution $x \in \{0, 1\}^n$, then $f^n(x) = f(x')$ where x' obtained by independently flipping each bit of the original solution x with probability p, i.e., noise level.

The Gaussian noise model (σ^2) [14, 16, 21–23, 26] is a type of posterior noise, which adds a value independently sampled from a normal distribution in each evaluation. We can define it as: given a parameter $\sigma \geq 0$, i.e., noise level, and a solution $x \in \{0, 1\}^n$, then $f^n(x) = f(x) + \mathcal{N}(0, \sigma^2)$.

The symmetric noise model was first introduced by Qian et al. [24]. They show that the (1+1) EA with a resampling strategy can fail in the symmetric noise model with fixed noise level $q = 1/2$, while using a population can be helpful. In this paper, we propose a more general symmetric noise model using variable noise level q: given a probability $q \in [0, 1]$, i.e., noise level, and an arbitrary number $C \in \mathbb{R}$ and a solution $x \in \{0, 1\}^n$, then

$$f^n(x) = \begin{cases} f(x) & \text{with probability } 1 - q \\ C - f(x) & \text{with probability } q. \end{cases}$$

For dynamic optimisation, we use $f^t(x)$ instead of $f(x)$, where $t \in \mathbb{N}$ represents the current generation. Unlike noisy optimisation, the uncertainty of dynamic optimisation is reflected in different generations rather than in different evaluations. For example, the weight of each bit position in a linear function is sampled from a distribution in each generation, and each individual is evaluated by computing the weighted sum. This dynamic version of linear functions was first proposed by Lengler and Schaller [30]. The DYNBV problem is a special case of random linear functions, which was first proposed by Lengler and Meier [15]. It can be regarded as a dynamic version of the BINVAL problem. For the DYNBV problem, we uniformly sample a new permutation $\pi_t : [n] \to [n]$ and evaluate the individuals in the t-th generation by $f^t(x) = \sum_{i=1}^{n} 2^{n-i} x_{\pi_t(i)}$.

2.2 Non-elitist EAs

Non-elitism means that the fittest individual in a generation is not always copied to the next generation. There exist many non-elitist selection mechanisms [39–43]. It may seem counterintuitive not to keep the fittest individuals in the population during optimisation. However, in recent studies, non-elitism (or with parameter control mechanisms) shows its benefits for helping algorithms escaping local optima [44–46], but parameters, e.g., mutation rate, should be set carefully. In this paper, we consider two popular selection mechanisms: 2-tournament and (μ, λ) selection [47] (see Sect. 2.3). Algorithm 1 describes a non-elitist EA framework. In the t-th generation, we obtain a new individual via selection and mutation. We first select an individual z by independently sampling from a distribution $P_{sel}(P_t) : \mathcal{X}^\lambda \to \mathcal{X}$ which selects one individual from the population P_t. Then we obtain a new individual y by mutating each bit-position in z independently with probability χ/n, the so-called mutation rate. The distribution $P_{sel}(P_t)$ is determined by the selection mechanism.

Algorithm 1 Non-elitist EA

Require: Fitness function $f : \mathcal{X} \to \mathbb{R}$; Population sizes $\lambda \in \mathbb{N}$ where $\lambda \geq 2$; Mutation parameter
$\chi \in (0, n)$; Initial population $P_0 \in \mathcal{X}^\lambda$.
1: **for** $t = 0, 1, 2, \dots$ until termination condition met **do**
2:     **for** $i = 1$ to λ **do**
3:         Sample $z \sim P_{sel}(P_t)$
4:         $P_{t+1}(i) \leftarrow y$, y is obtained by mutating z with mutation rate χ/n.
5:     **end for**
6: **end for**

In practice, the time spent on performing fitness function evaluation is usually significantly higher than the time spent on the rest of operations in each generation. Therefore, the runtime of an EA can be defined by the number of fitness function evaluations until an optimum is found, i.e., $T := \min\{t\lambda \mid |P_t \cap A^*| > 0\}$ where A^* is a set containing all optimal solutions. We say that optimisation is efficient if the expected runtime $E[T] = \text{poly}(n)$, and it is inefficient if the runtime is in $e^{\Omega(n)}$ with a probability $1 - e^{-\Omega(n)}$.

2.3 Selection Mechanisms

Algorithm 2 describes the binary tournament selection mechanism in which the fittest individual is picked up from two uniformly selected individuals x_1 and x_2. Thus, the 2-tournament EA can be described as Algorithm 1 using the 2-tournament selection mechanism shown in Algorithm 2. For noisy optimisation, we compare the pair of individuals based on the noisy function $f^n(x)$ instead of $f(x)$. In practice, we often evaluate once for each individual in the population P_t before selection and mutation. In this case, each comparison in 2-tournament selection would not be an independent event which can make analysis difficult. Thus, the reevaluation strategy [16, 17, 19, 21, 22, 24] is applied in this study which means the noisy fitness value of an individual will be reevaluated every time the individual enters a tournament. Similarly, for dynamic optimisation, we replace $f(x_1) \geq f(x_2)$ with $f^t(x_1) \geq f^t(x_2)$ in line 3 of Algorithm 2.

Algorithm 2 2-tournament selection mechanism

Require: Fitness function $f : \mathcal{X} \to \mathbb{R}$; Population sizes $\lambda \in \mathbb{N}$ where $\lambda \geq 2$; population $P \in \mathcal{X}^\lambda$.
1: $x_1 \leftarrow P(i_1)$ where $i_1 \backsim \text{Uniform}([\lambda])$
2: $x_2 \leftarrow P(i_2)$ where $i_2 \backsim \text{Uniform}([\lambda])$
3: **if** $f(x_1) \geq f(x_2)$ **then** $z \leftarrow x_1$ **else** $z \leftarrow x_2$
4: Return z

Algorithm 3 describes the (μ, λ) selection mechanism in which an individual is selected uniformly at random among the μ individuals with the highest fitness. Note that we usually sort P once in a generation in practice. In noisy environments, we assume that the noisy fitness value of an individual will not change during a generation, i.e., we also sort P once in a generation. We define the *selection pressure* of the (μ, λ)

selection mechanism as λ/μ, which refers to the expected number of offspring from any given individual ranked μ or better [39].

Algorithm 3 (μ, λ) selection mechanism

Require: Fitness function $f : \mathcal{X} \to \mathbb{R}$; Population sizes $\mu, \lambda \in \mathbb{N}$ where $\lambda > \mu \geq 2$; population $P \in \mathcal{X}^\lambda$.
1: Sort P s.t. $f(P(1)) \geq \cdots \geq f(P(\lambda))$
2: $z \leftarrow P(i)$ where $i \curvearrowleft \text{Uniform}([\mu])$
3: Return z

2.4 Analytical Tools

2.4.1 Level-Based Analysis

The *level-based theorem* [36, 48, 49] is a runtime analysis tool used to obtain upper bounds on the runtime of many EAs on a wide variety of optimisation problems [38, 45, 50]. The theorem applies to algorithms that follow the scheme of Algorithm 4. Assume that the search space $\mathcal{X}$ is partitioned into $m + 1$ disjoint subsets A_0, A_1, ..., A_m, which are called levels. The final level A_m consists of the optimal solution of function f. We denote $A_{\geq j} := \cup_{k=j}^{m} A_k$ the search points in level j and higher. Let D be some mapping from the set of all possible populations $\mathcal{X}^\lambda$ into the space of probability distributions of $\mathcal{X}$. Thus, we can consider that each individual in P_{t+1} is sampled from a distribution $D(P_t)$ which represent all genetic operators, such as selection and mutation. For example, line 3 of Algorithm 4 is corresponding to lines 3 and 4 of Algorithm 1.

Theorem 1 *(Level-based theorem [36]) Given a partition $(A_0, A_1, ..., A_m)$ of a finite state space $\mathcal{X}$, let $T := \min\{t\lambda \mid |P_t \cap A_m| > 0\}$ be the first point in time that the elements of A_m appear in P_t of Algorithm 4. If there exist $z_0, z_1, \ldots, z_{m-1}, \delta \in (0, 1]$, and $\gamma_0 \in (0, 1)$ such that for any population $P \in \mathcal{X}^\lambda$,*

(G1) for all $j \in [0..m - 1]$, if $|P \cap A_{\geq j}| \geq \gamma_0 \lambda$ then

$$\Pr_{y \sim D(P)} \left(y \in A_{\geq j+1} \right) \geq z_j,$$

(G2) for all $j \in [0..m-2]$, and all $\gamma \in (0, \gamma_0]$, if $|P \cap A_{\geq j}| \geq \gamma_0 \lambda$ and $|P \cap A_{\geq j+1}| \geq \gamma \lambda$ then $\Pr_{y \sim D(P)} \left(y \in A_{\geq j+1} \right) \geq (1 + \delta)\gamma$,

(G3) and the population size $\lambda \in \mathbb{N}$ satisfies

$$\lambda \geq 4/(\gamma_0 \delta^2) \ln \left(128(m + 1)/(z_* \delta^2) \right) \ where \ z_* := \min \{z_j\},$$

then $E[T] \leq \frac{8}{\delta^2} \sum_{j=0}^{m-1} \left(\lambda \ln \left(\frac{6\delta\lambda}{4 + z_j \delta\lambda} \right) + \frac{1}{z_j} \right).$

Algorithm 4 Population-based Algorithm

Require: Finite state space $\mathcal{X}$ and population size $\lambda \in \mathbb{N}$; Map D from $\mathcal{X}^\lambda$ to the space of probability distributions over $\mathcal{X}$
Require: Initial population $P_0 \in \mathcal{X}^\lambda$
1: **for** $t = 0, 1, 2, \ldots$ until termination condition met **do**
2:     **for** $i = 1$ to λ **do**
3:         Sample $P_{t+1}(i) \sim D(P_t)$
4:     **end for**
5: **end for**

2.4.2 Negative Drift for Populations

In this paper, we apply the *negative drift theorem for populations* [37] to obtain tail bounds on the runtime of Algorithm 1 in uncertain environments. Let $I_t(j)$ denote the j-th sampled index in generation t. We first define the *reproductive rate* α_0 [37] of the individual $P_t(i)$ in Algorithm 2 as the expected number of times the individual is sampled from $P_{sel}(P_t)$, i.e., $E[R_t(i) \mid P_t]$ where $R_t(i) := \sum_{j=1}^{\lambda} [I_t(j) = i]$ for $t \in \mathbb{N}$ and $i \in [\lambda]$. Informally, the negative drift theorem for populations (Theorem 2) states that if all individuals close to a given search point $x^* \in \mathcal{X}$ have a reproductive rate below $e^{-\chi}$, then the algorithm needs exponential time to reach x^*.

Without uncertainties, it is well-know that $\alpha_0 = 2(1 - 1/\lambda)$ and $\alpha_0 = \lambda/\mu$ for 2-tournament selection and (μ, λ), respectively [51]. By Theorem 2, the runtime of optimising on any function with a polynomially number of global optima, e.g., ONE-MAX and LEADINGONES, with the 2-tournament EA is exponential if the mutation rate is greater than $\ln(\alpha_0 + \delta)/n$ for a constant $\delta > 0$. Section 4.4 shows that uncertainties essentially affect the maximal reproductive rate α_0, where a low mutation rate is required to avoid exponential runtime according to a high-level noise.

Theorem 2 [37] *The probability that Algorithm 1 with population size $\lambda = \text{poly}(n)$, mutation rate χ/n and maximal reproductive rate bounded by $\alpha_0 < e^\chi - \delta$, for a constant $\delta > 0$, optimises any function with a polynomial number of optima within e^{cn} generations is $e^{-\Omega(n)}$, for some constant $c > 0$.*

3 2-Tournament EA in Uncertain Environments

In this section, we introduce a general result (Theorem 3) which is an upper bound of the expected runtime of the 2-tournament EA in uncertain environments. The key step is to estimate the probability of the "real" fittest individual being selected from x_1 and x_2 in line 3 of Algorithm 2. In an uncertainty-free case, the fittest individual is selected with probability 1. In uncertain environments, condition (C2) of Theorem 3 is satisfied if the probability that the truly fitter individual is selected is greater than $1/2$. We call this probability minus $1/2$ *fitness bias*. This property of an uncertain problem decides how small the mutation rate should be set, how large the population size should be used and how fast the algorithm can achieve the optimum. We summarise fitness biases in some noisy scenarios in Lemma 1. Note that the concept of fitness bias only describes a property of an uncertainty model for 2-tournament selection.

The general theorem for the 2-tournament EA is derived from the level-based theorem (Theorem 1). Condition (C1) is to estimate a lower bound of the probability of "level upgrading", i.e., producing an individual in level $j + 1$ after mutation of an individual in level j. Condition (C2) shows the fitness bias in uncertain environments. Condition (C3) states the required population size. Finally, we can get an upper bound for the runtime.

Theorem 3 *Let $(A_0, A_1,..., A_m)$ be a fitness partition of a finite state space $\mathcal{X}$. Let $T := \min\{2t\lambda \mid |P_t \cap A_m| > 0\}$ be the first point in time that the elements of A_m appear in P_t of the 2-tournament EA with noisy function $f^n(x)$ and mutation rate χ/n. If there exist $h_0, h_1,..., h_{m-1}$ and $\theta \in (0, 1/2]$, and where $\chi \in (0, \ln(1 + 2\theta\zeta))$ for an arbitrary constant $\zeta \in (0, 1)$, such that, for an arbitrary constant $\xi \in (0, 1/16)$,*

(C1) *for all $j \in [0..m-1]$, $\Pr(y \in A_{\geq j+1} \mid z \in A_j) \geq h_j$,*
(C2) *for all $j \in [0..m-2]$, and all search points $x_1 \in A_{\geq j+1}$ and $x_2 \in A_{\leq j}$, it follows $\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) \geq \frac{1}{2} + \theta$,*
(C3) *and the population size $\lambda \in \mathbb{N}$ satisfies*

$$\lambda > \frac{4(1 + o(1))}{\theta^2 \xi (1 - \zeta)^4} \ln\left(\frac{128(m + 1)}{\theta^2 \xi (1 - \zeta)^4 \min\{h_j\}}\right),$$

then $E[T] < \frac{16(1 + o(1))}{\theta^2 \xi (1-\zeta)^2} \sum_{j=0}^{m-1}\left(\lambda \ln\left(\frac{6}{\xi(1-\zeta)^2 h_j}\right) + \frac{1}{\xi(1-\zeta)^2 h_j}\right)$.

Proof We use the level-based theorem (Theorem 1) to prove Theorem 3. Firstly, we derive some inequalities which are used later. From $\theta \in (0, 1/2]$, $\zeta \in (0, 1)$ and $0 < \chi < \ln(1 + 2\theta\zeta)$ which are assumptions of Theorem 3, we obtain

$$e^\chi < 1 + 2\theta\zeta \tag{1}$$
$$(1 + 2\theta) - e^\chi > 2\theta(1 - \zeta). \tag{2}$$

Then we define ε and γ_0 which are used later. Let constant $\varepsilon := \left(1 + \sqrt{1 - 4\sqrt{\xi}}\right)/2$, and we know that $\varepsilon \in (1/2, 1)$ by $\xi \in (0, 1/16)$. We define $\gamma_0 := \frac{(1+2\theta) - \exp(\chi)}{2\theta}(1 - \varepsilon)$. By Eq. (2), we know that

$$\gamma_0 = \frac{(1 + 2\theta) - e^\chi}{2\theta}(1 - \varepsilon) > \frac{2\theta(1 - \zeta)(1 - \varepsilon)}{2\theta} = (1 - \zeta)(1 - \varepsilon). \tag{3}$$

We first show that condition (G2) of Theorem 1 holds. We define the "current level" to be the highest level $j \in [0..m-1]$ such that there are at least $\gamma_0\lambda$ individuals in level j or higher, and there are fewer than $\gamma_0\lambda$ individuals in level $j+1$ or higher. Following condition (G2) of Theorem 1, we assume that the current level is $j \leq m - 2$, which means that there are at least $\gamma_0\lambda$ individuals of the population P_t in $A_{\geq j}$, and at least $\gamma\lambda$ but less than $\gamma_0\lambda$ individuals in $A_{\geq j+1}$. Let x_1 and x_2 be the individuals selected from the population P_t in lines 1 and 2 of 2-tournament selection (Algorithm 2), z be the solution after comparison in line 3 of 2-tournament selection (Algorithm 2), and y be the solution after mutating corresponding to line 4 of Algorithm 1.

Now we estimate a lower bound on the probability that the offspring y is still in $A_{\geq j+1}$. By the law of total probability,

$$\Pr(y \in A_{\geq j+1}) \geq \Pr(z \in A_{\geq j+1}) \cdot \Pr(y \in A_{\geq j+1} | z \in A_{\geq j+1}).$$

The probability of selecting an individual z which is in $A_{\geq j+1}$ via binary tournament is composed of two cases. The first case is both x_1 and x_2 which are selected in lines 1 and 2 of Algorithm 2 are in $A_{\geq j+1}$ whose probability is at least γ^2. The second case is that x_1 or x_2 is evaluated to be in $A_{\geq j+1}$, whereas the other is evaluated to be in $A_{\leq j}$. In this case, noise leads to incorrect comparison in line 3 of Algorithm 2 with some probability. Let S be the event of a successful comparison, i.e. the better individual of x_1 and x_2 is exactly selected from line 3 of Algorithm 2. Hence, the second case occurs with probability $2(1-\gamma)\gamma \Pr(S)$. Then,

$$\Pr(z \in A_{\geq j+1}) \geq \gamma^2 + 2(1-\gamma)\gamma \Pr(S).$$

To estimate a lower bound for $\Pr(S)$, we assume without loss of generality $x_1 \in A_{\geq j+1}$ and $x_2 \in A_{\leq j}$. Then, by condition (C2),

$$\Pr(S) = \Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2} \Pr(f^n(x_1) = f^n(x_2)) \geq \frac{1}{2} + \theta.$$

To estimate a lower bound for $\Pr(y \in A_{\geq j+1} | z \in A_{\geq j+1})$, we only consider the case that the mutation operator does not flip any bits, then by Lemmas 3 and 2,

$$\Pr(y \in A_{\geq j+1} | z \in A_{\geq j+1})$$
$$\geq \left(1 - \frac{\chi}{n}\right)^n \geq e^{-\chi}\left(1 - \frac{\chi^2}{n}\right) \geq e^{-\chi}\left(1 - \frac{2\theta}{n}\right) \text{ for all } n > 1.$$

Overall, we can get a lower bound for $\Pr(y \in A_{\geq j+1})$ by plugging in $\Pr(z \in A_{\geq j+1})$, $\Pr(y \in A_{\geq j+1} | z \in A_{\geq j+1})$ and $\Pr(S)$,

$$\Pr(y \in A_{\geq j+1})$$
$$> \left(\gamma^2 + 2(1-\gamma)\gamma \Pr(S)\right) e^{-\chi}\left(1 - \frac{2\theta}{n}\right)$$
$$\geq \left(\gamma^2 + 2(1-\gamma)\gamma \left(\frac{1}{2} + \theta\right)\right) e^{-\chi}\left(1 - \frac{2\theta}{n}\right)$$
$$\geq \gamma \left(1 + 2\theta - 2\theta\gamma_0\right) e^{-\chi}\left(1 - \frac{2\theta}{n}\right)$$

by the definition of $\gamma_0 = \frac{(1+2\theta) - \exp(\chi)}{2\theta}(1 - \varepsilon)$,

$$= \gamma \left(1 + 2\theta - \left(1 + 2\theta - e^\chi\right) + \left(1 + 2\theta - e^\chi\right)\varepsilon\right) e^{-\chi}\left(1 - \frac{2\theta}{n}\right)$$

$$= \gamma \left(1 + \left(1 + 2\theta - e^{\chi}\right) \varepsilon e^{-\chi}\right) \left(1 - \frac{2\theta}{n}\right)$$

letting $\delta := \left(1 + (1 + 2\theta - e^{\chi}) \varepsilon e^{-\chi}\right)(1 - 2\theta/n) - 1$,

$$= \gamma(1 + \delta). \tag{4}$$

Now we prove that $\delta > 0$, where

$$\delta = \left(1 + \left(1 + 2\theta - e^{\chi}\right) \varepsilon e^{-\chi}\right) \left(1 - \frac{2\theta}{n}\right) - 1$$

by Eq. (1),

$$> \left(1 + 2\theta e^{-\chi} \varepsilon(1 - \zeta)\right) \left(1 - \frac{2\theta}{n}\right) - 1$$
$$= 2\theta e^{-\chi} \varepsilon(1 - \zeta) - \frac{2\theta}{n} \left(1 + 2\theta e^{-\chi} \varepsilon\right) + \frac{4\theta^2 e^{-\chi} \varepsilon \zeta}{n}$$
$$> 2\theta e^{-\chi} \varepsilon(1 - \zeta) - \frac{6\theta}{n} = \theta \left(2e^{-\chi} \varepsilon(1 - \zeta) - \frac{6}{n}\right)$$

by Eq. (1), we have $e^{\chi} < 1 + 2\theta\zeta < 1 + 2\theta < 2$,

$$> \theta \left(\varepsilon(1 - \zeta) - \frac{6}{n}\right) = \theta \varepsilon(1 - \zeta)(1 - o(1)). \tag{5}$$

Thus, we get $\delta > 0$ so condition (G2) of Theorem 1 holds from Eq. 4.

To verify condition (G1), we estimate the probability of sampling an individual beyond the current level of the population. We assume there are at least $\gamma_0 \lambda$ individuals in $A_{\geq j}$ where $j \in [0..m - 1]$. We only consider the case that the selected individuals are both in A_j in lines 1 and 2 of Algorithm 2, and the individual increases its level after mutation,

$$\Pr(y \in A_{\geq j+1}) \geq \gamma_0^2 \Pr(y \in A_{\geq j+1} \mid z \in A_{\geq j}) \geq \gamma_0^2 h_j =: z_j.$$

Condition (G3) requires the population size to satisfy

$$\frac{4}{\gamma_0 \delta^2} \ln \left(\frac{128(m + 1)}{\min\{z_j\}\delta^2}\right) = \frac{4}{\gamma_0 \delta^2} \ln \left(\frac{128(m + 1)}{\gamma_0^2 \min\{h_j\}\delta^2}\right)$$

by Eq. (3) and (5),

$$
< \frac{4\,(1 + o(1))}{(1 - \zeta)(1 - \varepsilon)\,(\theta\varepsilon(1 - \zeta))^2} \cdot \ln\left(\frac{128(m + 1)\,(1 + o(1))}{(1 - \zeta)^2(1 - \varepsilon)^2\,(\theta\varepsilon(1 - \zeta))^2 \min\{h_j\}} \right)
$$

$$
= \frac{4\,(1 + o(1))}{\theta^2\varepsilon^2(1 - \varepsilon)(1 - \zeta)^3} \ln\left(\frac{128(m + 1)}{\theta^2\varepsilon^2(1 - \varepsilon)^2(1 - \zeta)^4 \min\{h_j\}} \right)
$$

$$
< \frac{4\,(1 + o(1))}{\theta^2\varepsilon^2(1 - \varepsilon)^2(1 - \zeta)^4} \ln\left(\frac{128(m + 1)}{\theta^2\varepsilon^2(1 - \varepsilon)^2(1 - \zeta)^4 \min\{h_j\}} \right)
$$

because $\varepsilon^2(1 - \varepsilon)^2 = \xi$ by the definition of $\varepsilon = \left(1 + \sqrt{1 - 4\sqrt{\xi}}\right)/2$,

$$
< \frac{4\,(1 + o(1))}{\theta^2\xi(1 - \zeta)^4} \ln\left(\frac{128(m + 1)}{\theta^2\xi(1 - \zeta)^4 \min\{h_j\}} \right) < \lambda.
$$

Therefore, condition (C3) of Theorem 3 guarantees that the population size satisfies condition (G3) of Theorem 1.

Finally, all conditions of Theorem 1 hold and the expected time (the reevaluation strategy is taken into account) to reach the optimum is no more than

$$
E[T] \le 2 \cdot \frac{8}{\delta^2} \sum_{j=0}^{m-1} \left(\lambda \ln\left(\frac{6\delta\lambda}{4 + z_j\delta\lambda} \right) + \frac{1}{z_j} \right) < \frac{16}{\delta^2} \sum_{j=0}^{m-1} \left(\lambda \ln\left(\frac{6}{z_j} \right) + \frac{1}{z_j} \right)
$$

$$
\le \frac{16}{\delta^2} \sum_{j=0}^{m-1} \left(\lambda \ln\left(\frac{6}{\gamma_0^2 h_j} \right) + \frac{1}{\gamma_0^2 h_j} \right)
$$

by Eq. (3) and (5),

$$
< \frac{16\,(1 + o(1))}{\theta^2\varepsilon^2(1 - \zeta)^2} \sum_{j=0}^{m-1} \left(\lambda \ln\left(\frac{6}{(1 - \varepsilon)^2(1 - \zeta)^2 h_j} \right) + \frac{1/h_j}{(1 - \varepsilon)^2(1 - \zeta)^2} \right)
$$

$$
< \frac{16\,(1 + o(1))}{\theta^2\varepsilon^2(1 - \varepsilon)^2(1 - \zeta)^2} \sum_{j=0}^{m-1} \left(\lambda \ln\left(\frac{6}{\varepsilon^2(1 - \varepsilon)^2(1 - \zeta)^2 h_j} \right) \right.
$$

$$
\left. + \frac{1/h_j}{\varepsilon^2(1 - \varepsilon)^2(1 - \zeta)^2} \right)
$$

$$
= \frac{16\,(1 + o(1))}{\theta^2\xi(1 - \zeta)^2} \sum_{j=0}^{m-1} \left(\lambda \ln\left(\frac{6}{\xi(1 - \zeta)^2 h_j} \right) + \frac{1}{\xi(1 - \zeta)^2 h_j} \right).
$$

$\square$

In Lemma 1, we show fitness biases of ONEMAX and LEADINGONES functions in the one-bit, the bit-wise, the Gaussian and the symmetric noise models.

Lemma 1 *Let* $A_j := \{x \in \{0, 1\}^n | f(x) = j\}$ *for all* $j \in [0...n]$ *be a partition of* $\{0, 1\}^n$. *Let* x_1 *and* x_2 *be two individuals in* $A_{\geq j+1}$ *and* $A_{\leq j}$ *respectively, where* $j \in [0...n-2]$, *then* $\theta_1 \leq \Pr\left(f^n(x_1) > f^n(x_2)\right) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) - 1/2 \leq \theta_2$ *where*

(a) $\theta_1 = 1/2 - q/2(1-q/2) - q/(2n_0)$ *for* $q \in [0, 1)$ *and* $n_0 \in [3, \infty)$ *on* ONEMAX *in the one-bit noise model* (q),

(b) $\theta_1 = 1/2 - q(1-q/2)$ *for* $q \in [0, 1)$ *on* LEADINGONES *in the one-bit noise model* (q),

(c) $\theta_1 = \frac{9(1/2-p)}{64\sqrt{2pn}+16}$ *for* $p \in (0, 1/2)$ *on* ONEMAX *in the bit-wise noise model* (p),

(d) $\theta_1 = (1/2 - 3p/2) e^{-3np}$ *for* $p \in [0, 1/3)$ *on* LEADINGONES *in the bit-wise noise model* (p), *and*

(e) $\theta_1 = 1/(6+48\sigma/\pi)$ *for* $\sigma > 0$ *on* ONEMAX *and* LEADINGONES *in the Gaussian noise model* (σ^2).

(f) $\theta_1 = \theta_2 = 1/2 - q$ *for any* $C \in \mathbb{R}$ *and* $q \in [0, 1/2)$ *on* ONEMAX *and* LEADINGONES *in the symmetric noise model* (C, q).

Proof Let E be the event that $f^n(x_1) > f^n(x_2)$ or individual x_1 is selected uniformly from $\{x_1, x_2\}$ if $f^n(x_1) = f^n(x_2)$, then $\Pr(E) = \Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2))$. Now we derive the fitness bias in different cases.

(a) To estimate a lower bound for $\Pr(E)$ on the ONEMAX problem in the one-bit noise model (q), we pessimistically assume that $x_1 \in A_{j+1}$ and $x_2 \in A_j$. Then we say that x_1 "wins" if the event E happens, and we distinguish between four cases:

- Let E_{00} be the event that there is no noise, and x_1 wins, then $\Pr(E_{00}) = (1-q)^2$.
- Let E_{01} be the event that there is no noise in x_1 and noise in x_2, and x_1 wins, then
$$\Pr(E_{01}) \geq (1-q)q\left(\left(1 - \frac{j}{n}\right)\frac{1}{2} + \frac{j}{n}\right) = q(1-q)\left(\frac{j}{2n} + \frac{1}{2}\right).$$
- Let E_{10} be the event that there is noise in x_1 and no noise in x_2, and x_1 wins, then
$$\Pr(E_{10}) \geq q(1-q)\left(\frac{j+1}{n} \cdot \frac{1}{2} + \left(1 - \frac{j+1}{n}\right)\right) = q(1-q)\left(-\frac{j}{2n} - \frac{1}{2n} + 1\right).$$
- Let E_{11} be the event that there is noise in x_1 and x_2, and x_1 wins. There are two situations leading x_1 to win:

  1. The noise flips one of the $j+1$ 1-bits of x_1 and one of the j 1-bit of x_2.
  2. The noise flips one of $n-(j+1)$ 0-bits of x_1.

Thus,

$$\Pr(E_{11}) \geq q^2\left(\frac{j+1}{n} \cdot \frac{j}{n} + \left(1 - \frac{j+1}{n}\right)\right) = \left(\frac{(j+1)(j-n)}{n^2} + 1\right)q^2$$

since $(j+1)(j-n)$ achieves the minimum when $j = (n-1)/2$,

$$\geq \left(\frac{3}{4} - \frac{1}{2n}\right)q^2.$$

By combining all four cases above, we obtain

$$\Pr(E) \geq \Pr(E_{00}) + \Pr(E_{01}) + \Pr(E_{10}) + \Pr(E_{11})$$

$$\geq (1-q)^2 + q(1-q)\left(\frac{j}{2n} + \frac{1}{2}\right)$$

$$+ q(1-q)\left(-\frac{j}{2n} - \frac{1}{2n} + 1\right) + \left(\frac{3}{4} - \frac{1}{2n}\right)q^2$$

$$= 1 - \frac{q}{2} + \frac{q^2}{4} - \frac{q}{2n} = \frac{1}{2} + \frac{1}{2} - \frac{q}{2}\left(1 - \frac{q}{2}\right) - \frac{q}{2n}$$

$$\geq \frac{1}{2} + \frac{1}{2} - \frac{q}{2}\left(1 - \frac{q}{2}\right) - \frac{q}{2n_0} = \frac{1}{2} + \theta$$

(b) To estimate a lower bound for $\Pr(E)$ on the LEADINGONES problem in the one-bit noise model (q), we pessimistically assume that $x_1 \in A_{j+1}$ and $x_2 \in A_j$. We also pessimistically assume that the suffix of x_1, i.e. the bits after the $(j+2)$-th position, are all 0-bits, and the suffix of x_2, i.e. the bits after the $(j+1)$-th position, are all 1-bits, which is the worst case because if the noise flips the $(j+1)$-th bit in x_2, then x_2 will have the maximal noisy fitness n. We say that x_1 "wins" if the event E happens, then we distinguish between four cases to estimate $\Pr(E)$:

- Let E_{00} be the event that there is no noise, and x_1 wins, then $\Pr(E_{00}) = (1-q)^2 = q^2 - 2q + 1$.
- Let E_{01} be the event that there is no noise in x_1 and noise in x_2, and x_1 wins. By the assumption of x_2, x_1 only fails if noise flips the only 0-bit in x_2. Thus, $\Pr(E_{01}) \geq (1-q) \cdot q \cdot (1 - 1/n) = -(1 - 1/n)q^2 + (1 - 1/n)q$.
- Let E_{10} be the event that there is noise in x_1 and no noise in x_2, and x_1 wins. Unless any of the first $j+1$ 1-bits of x_1 is flipped, x_1 wins. Therefore, $\Pr(E_{10}) \geq q \cdot (1-q) \cdot (1 - (j+1)/n) = -(1 - (j+1)/n) \cdot q^2 + (1 - (j+1)/n)q$.
- Let E_{11} be the event that there is noise in x_1 and x_2, and x_1 wins. Because $j = n-2$ is a special case, we first estimate the probability $\Pr(E_{11})$ when $j \leq n-3$. There are four situations leading x_1 to win:

  1. The noise does not flip the first $j+1$ 1-bits of x_1, and does not flip the 0-bit of x_2.
  2. The noise flips the i-th 1-bits of x_1 where $i \leq j+1$, and flips one of the first $i-1$ 1-bits of x_2,
  3. The noise flips the same bit-position in the first j 1-bits of x_1 and x_2 (tie and half chance to win).
  4. The noise flips the $(j+1)$-th 1-bit of x_1, and does not flip the first j 1-bits of x_2 (tie and half chance to win).

Thus,

$$\Pr(E_{11}) \geq q^2 \left(\left(1 - \frac{j+1}{n}\right)\left(1 - \frac{1}{n}\right) + \sum_{i=2}^{j+1}\left(\frac{i-1}{n} \cdot \frac{1}{n}\right) + \frac{j}{2n^2} \right)$$

$$+ \frac{1}{2n}\left(1 - \frac{j+1}{n}\right)\Bigg)$$

$$= \left(j^2/2 - (n - 3/2)\,j + (n-1)(2n-1)/2\right)\frac{q^2}{n^2}$$

since $j^2/2 - (n-3/2)\,j + (n-1)(2n-1)/2$ is monotone decreasing as j increases when $j \le n - 3/2$, $\Pr(E_{11})$ achieves the minimum if $j = n - 3$,

$$\ge \frac{1}{2}\left(1 + \frac{1}{n^2}\right)q^2 > \frac{q^2}{2}.$$

Then we estimate $\Pr(E_{11})$ in the special case $j = n - 2$. Since both x_1 and x_2 have one 0-bit to the optimum, i.e. x_1 has only one 0-bit in the last position and x_2 has only one 0-bit in the penultimate position, there are five situations to leading x_1 to win:

1. The noise flips the i-th 1-bits of x_1 where $i \le n - 1$, and flips one of the first $i - 1$ 1-bits of x_2.
2. The noise flips the same bit-position in the first $n - 2$ 1-bits of x_1 and x_2 (tie and half chance to win).
3. The noise flips the last 0-bits of x_1, and does not flip the 0-bit of x_2.
4. The noise flips both the 0-bits of x_1 and x_2 (tie and half chance to win).
5. The noise flips the $(n-1)$-th 1-bit of x_1, and flips the last 0-bits of x_2 (tie and half chance to win).

Thus,

$$\Pr(E_{11}) \ge q^2 \left(\sum_{i=2}^{n-1}\left(\frac{i-1}{n}\cdot\frac{1}{n}\right) + \frac{n-2}{2n^2} + \frac{1}{n}\left(1 - \frac{1}{n}\right) + \frac{1}{2n^2} + \frac{1}{2n^2}\right) = \frac{q^2}{2}.$$

Therefore, we obtain $\Pr(E_{11}) \ge q^2/2$ for all $j \le n - 2$.

By combining all four cases above and $j \le n - 2$, we obtain

$$\Pr(E) \ge \Pr(E_{00}) + \Pr(E_{01}) + \Pr(E_{10}) + \Pr(E_{11})$$

$$\ge 1 - \frac{j+2}{n}q + \left(\frac{j+2}{n} - 1\right)q^2 + \frac{q^2}{2}$$

$$\ge 1 - \frac{j+2}{n}(1-q)q - q^2 + \frac{q^2}{2} \ge 1 - (1-q)q - q^2 + \frac{q^2}{2}$$

$$= 1 - q + \frac{q^2}{2} = \frac{1}{2} + \frac{1}{2} - q(1 - \frac{q}{2}) = \frac{1}{2} + \theta.$$

(c) To estimate a lower bound for $\Pr(E)$ on the ONEMAX problem in the bit-wise noise model (p), we pessimistically assume that $x_1 \in A_{j+1}$ and $x_2 \in A_j$, such that $f(x_1) = f(x_2) + 1$. There exists at least one bit-position i, such that x_1 has a 1-bit

in position i and x_2 has a 0-bit in position i. The remaining bits of x_1 and x_2 have the same number of 1-bits. Therefore, the bits after noise of x_1 and x_2 in position i decide the outcome of the fitness comparison. Let x_1' and x_2' be two substrings obtained by excluding position i from x_1 and x_2 respectively. Since each bit is flipped independently, we know that $f^n(x_1'), f^n(x_2') \sim \text{Bin}(n-1-j, p) + \text{Bin}(j, (1-p))$ which are Poisson-binomially distributed random variables with variance $\sigma^2 = (1-p)p(n-1)$. Then we apply Lemma 8 with $\sigma = \sqrt{(1-p)p(n-1)} \le \sqrt{pn}$ and $d = 2$ to obtain a lower bound for

$$\Pr(f^n(x_1') = f^n(x_2')) \ge \frac{(1 - 1/2^2)^2}{2 \cdot 2\sqrt{2pn} + 1} \ge \frac{9}{64\sqrt{2pn} + 16}. \tag{6}$$

By symmetry, we know that $\Pr(f^n(x_1') > f^n(x_2')) = \Pr(f^n(x_1') < f^n(x_2'))$. Let $a = \Pr(f^n(x_1') = f^n(x_2'))$ and $b = Pr(f^n(x_1') > f^n(x_2'))$, then we obtain $a = 1 - 2b$. Thus,

$$\Pr(E) = b + a\left((1-p)^2 + 2 \cdot \frac{1}{2} \cdot p(1-p)\right)$$
$$= \frac{1}{2}(1-a) + a \cdot (1-p) = \frac{1}{2} + \left(\frac{1}{2} - p\right)a$$

by Eq. (6),

$$\ge \frac{1}{2} + \frac{9(1/2 - p)}{64\sqrt{2pn} + 16} = \frac{1}{2} + \theta.$$

(d) To estimate a lower bound for $\Pr(E)$ on the LEADINGONES problem in the bit-wise noise model (p), we pessimistically assume that $x_1 \in A_{j+1}$ and $x_2 \in A_j$. We also pessimistically assume that the suffix of x_1, i.e. the bits after the $(j+2)$-th position, are all 0-bits, and the suffix of x_2, i.e. the bits after the $(j+1)$-th position, are all 1-bits, which is the worst case since the noise flipping $(j+1)$-th bit of x_2 to achieve the optimum which leads to incorrect comparison while the noise only can at most increase 1 fitness for x_1. We distinguish between three cases to estimate $\Pr(E)$:

- Let E_0 be the event that $f^n(x_1) \ge j+1$ and $f^n(x_2) \le j$, then $\Pr(E_0) = (1-p)^{j+1}(1-p) + (1-p)^{j+1}p\left(1 - (1-p)^j\right) = \left(1 - p(1-p)^j\right)(1-p)^{j+1}$.
- Let E_{1i} be the event that $f^n(x_1) = i$ and $f^n(x_2) \le i-1$ for any $i \in [1, j]$, then $\Pr(E_1) = \sum_{i=1}^{j} \Pr(E_{1i}) = \sum_{i=1}^{j}\left(p(1-p)^i \cdot \left(1 - (1-p)^i\right)\right) = p\left(\sum_{i=1}^{j}(1-p)^i - \sum_{i=1}^{j}(1-p)^{2i}\right)$, by the sum of a geometric series,

$$\Pr(E_1) = p\left(\frac{1 - (1-p)^{j+1}}{1 - (1-p)} - \frac{1 - (1-p)^{2(j+1)}}{1 - (1-p)^2}\right)$$
$$= \frac{1}{2} - \frac{p}{2(2-p)} - (1-p)^{j+1} + \frac{1}{2-p}(1-p)^{2(j+1)}.$$

- Let E_{2i} be the event that $f^n(x_1) = i$ and $f^n(x_2) = i$ for any $i \in [0, j]$ then x_1 is selected uniformly, then

$$\Pr(E_2) = \frac{1}{2} \sum_{i=0}^{j} \Pr(E_{2i})$$

$$= \frac{1}{2} \left(\sum_{i=0}^{j-1} p^2 (1-p)^{2i} + p(1-p)(1-p)^{2j} \right)$$

since $p < 1/3$,

$$\geq \frac{1}{2} \sum_{i=0}^{j} p^2 (1-p)^{2i}$$

by the sum of a geometric series,

$$= \frac{1}{2} p^2 \left(\frac{1 - (1-p)^{2(j+1)}}{1 - (1-p)^2} \right)$$

$$= \frac{p}{2(2-p)} - \frac{p}{2(2-p)} (1-p)^{2(j+1)}.$$

By combining all three cases above, we obtain

$$\Pr(E) \geq \Pr(E_0) + \Pr(E_1) + \Pr(E_2)$$

$$= \left(1 - p(1-p)^j \right) (1-p)^{j+1} + \frac{1}{2} - \frac{p}{2(2-p)} - (1-p)^{j+1}$$

$$+ \frac{1}{2-p} (1-p)^{2(j+1)} + \frac{p}{2(2-p)} - \frac{p}{2(2-p)} (1-p)^{2(j+1)}$$

$$= \frac{1}{2} + \frac{(1-p)^{2(j+1)}}{2-p} - \frac{p}{2(2-p)} (1-p)^{2(j+1)} - p(1-p)^{2j+1}$$

$$= \frac{1}{2} + \frac{1}{2}(1-p)^{2j+2} - p(1-p)^{2j+1} = \frac{1}{2} + \left(\frac{1}{2} - \frac{3}{2} p \right) (1-p)^{2j+1}$$

by $\left((1-x)^{1/x-1} \right)^y \geq e^{-y}$ (see Lemma 4),

$$\geq \frac{1}{2} + \left(\frac{1}{2} - \frac{3}{2} p \right) e^{-(2j+1)\frac{p}{1-p}} > \frac{1}{2} + \left(\frac{1}{2} - \frac{3}{2} p \right) e^{-2(j+2)\frac{p}{1-p}}$$

by $0 \leq j \leq n - 2$ and $p \in [0, 1/3)$,

$$\geq \frac{1}{2} + \left(\frac{1}{2} - \frac{3}{2} p \right) e^{-3np} \geq \frac{1}{2} + \theta.$$

(e) To estimate a lower bound for $\Pr(E)$ on the ONEMAX and LEADINGONES problem in the Gaussian noise model (σ^2), we pessimistically assume that $x_1 \in A_{j+1}$ and $x_2 \in A_j$. Let $X \sim \mathcal{N}(0, 2\sigma^2)$ be a random variable, then

$$\Pr(E) \geq \Pr(f^n(x_1) - f^n(x_2) > 0) = \Pr(X > -1) = \Pr(X < 1)$$

by Lemma 6 with $x = 1$ and standard deviation is $\sqrt{2}\sigma$,

$$
\begin{aligned}
&> 1 - 1/\sqrt{\pi/(\sqrt{2}\sqrt{2}\sigma) + 4} \\
&= \left(1 - \left(1/\sqrt{\pi/(2\sigma) + 4}\right)^2\right) / \left(1 + 1/\sqrt{\pi/(2\sigma) + 4}\right) \\
&> \left(\frac{\pi/(2\sigma) + 3}{\pi/(2\sigma) + 4}\right) / \frac{3}{2} = \frac{1 + 6\sigma/\pi}{1 + 8\sigma/\pi} \cdot \frac{4}{3} \cdot \frac{1}{2} = \frac{1}{2} + \frac{1}{6 + 48\sigma/\pi} \\
&= \frac{1}{2} + \theta.
\end{aligned}
$$

(f) To estimate a lower bound for $\Pr(E)$ on ONEMAX and LEADINGONES in the symmetric noise model (C, q), we assume that $x_1 \in A_a$ and $x_2 \in A_b$ where $a > b$. Then we say that x_1 "wins" if event E happens, and we distinguish between three cases:

- If $a + b > C$, then x_1 wins if and only if there is noise in x_1, i.e, $\Pr(E) = (1-q)^2 + (1-q)q$.
- If $a + b = C$, then x_1 wins if and only if there is no noise in both x_1 and x_2, or there is noise in either x_1 and x_2 (same fitness values, so with half chance), i.e., $\Pr(E) = (1-q)^2 + (1-q)q/2 + q(1-q)/2$.
- If $a + b < C$, then x_1 wins if and only if there is no noise in x_2, i.e., $\Pr(E) = (1-q)^2 + q(1-q)$.

Therefore, we obtain

$$\Pr(E) = (1-q)^2 + (1-q)q = \frac{1}{2} + \frac{1}{2} - q = \frac{1}{2} + \theta.$$

$\square$

4 Noisy Optimisation

This section provides runtimes bounds for non-elitist EAs with 2-tournament and (μ, λ) selection on two classical functions in four noise models. We give the upper bound of runtime and the appropriate parameter settings, e.g., mutation rate, which leads to efficient optimisation for each noise model. In particular, we show for which parameter settings the non-elitist EA is inefficient in the symmetric noise model.

4.1 One-Bit Noise Model

Theorems 4 and 5 imply that the one-bit noise does not impact the asymptotical runtime of the 2-tournament EA if we choose a constant mutation parameter χ, which satisfies the assumption. However, we have fewer choices of the mutation rate as the level of is noise growing. In contrast, the (1+1) EA becomes inefficient if the noise level is a constant (see Tables 1 and 2). Compared to other EAs, e.g., ACO-fp, UMDA and (1+1) EA (resampling), the 2-tournament EA can outperform the current state of the art results in these two settings (see Tables 1 and 2).

Theorem 4 *For any constant $q \in [0, 1]$, any constant $n_0 \in [3, \infty)$ and any $\chi \in (0, \ln(1 + 2\theta))$, where $\theta := 1/2 - (q/2)(1 - q/2) - q/(2n_0)$, the 2-tournament EA with mutation rate χ/n and population size $\lambda > c \log(n/\chi)$ for a sufficiently large constant c achieves the optimum on* ONEMAX *in the one-bit noise model (q) in expected time $O\left(\lambda n \log(1/\chi) + n \log(n)/\chi\right)$.*

Proof We apply Theorem 3 to prove Theorem 4. If $\chi \in (0, \ln(1 + 2\theta))$, there exists a constant $\zeta \in (0, 1)$ such that $\chi \in (0, \ln(1 + 2\theta\zeta))$, which satisfies the condition in Theorem 3. We first partition the search space into levels. We use the partition $A_j := \{x \in \{0, 1\}^n | f(x) = j\}$ for all $j \in [0...n]$. By constants $q \in [0, 1]$ and $n_0 \in [3, \infty)$, we obtain $1/12 < \theta \le 1/2$ which satisfies the assumption in Theorem 3.

By case (a) of Lemma 1, we get $\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) > 1/2 + \theta$, then condition (C2) of Theorem 3 holds.

To verify condition (C1), we need to estimate the probability of sampling individuals beyond the current level of the population. We assume that there is an individual $z \in A_j$ where $j \in [0...n-1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. We only consider the case that no 1-bits is flipped and one of 0-bits is flipped after mutation, then by Lemma 5,

$$
\Pr(y \in A_{\ge j+1} \mid z \in A_j) > \left(1 - \frac{\chi}{n}\right)^j \frac{\chi}{n}(n - j)
$$
$$
\ge e^{-\chi}(1 - o(1))(n - j)\chi/n =: h_j \in \Omega\left((n - j)\chi/n\right).
$$

Then we compute the population size required by condition (C3). Let $\xi \in (0, 1/16)$ be a constant, then

$$
\lambda > \frac{4(1 + o(1))}{\theta^2\xi(1 - \zeta)^4} \ln\left(\frac{128(m + 1)}{\theta^2\xi(1 - \zeta)^4 \min\{h_j\}}\right) = O(\log(n/\chi)).
$$

Condition (C3) is satisfied by $\lambda \ge c \log(n/\chi)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 3 hold and the expected time to reach the optimum is no more than

$$
E[T] \le \frac{16(1 + o(1))}{\theta^2\xi(1 - \zeta)^2}\left(\lambda \sum_{j=0}^{m-1} \ln\left(\frac{6}{\xi(1 - \zeta)^2 h_j}\right) + \frac{1}{\xi(1 - \zeta)^2}\sum_{j=0}^{m-1}\frac{1}{h_j}\right)
$$

$$= O\left(\lambda \sum_{j=0}^{m-1} \ln\left(\frac{n}{(n-j)\chi}\right) + \sum_{j=0}^{m-1} \frac{n}{(n-j)\chi}\right)$$

$$= O\left(\lambda \ln\left(\prod_{j=0}^{m-1} \frac{n}{(n-j)\chi}\right) + n \sum_{j=0}^{m-1} \frac{1}{(n-j)\chi}\right)$$

$$= O\left(\lambda \ln\left(\frac{n^n}{n!\chi^n}\right) + n\log(n)/\chi\right)$$

using the lower bound $n! > (n/e)^n$,

$$= O\left(\lambda n \log(1/\chi) + n\log(n)/\chi\right).$$

$\square$

Theorem 5 *For any constant $q \in [0, 1)$ and any $\chi \in (0, \ln(1 + 2\theta))$, where $\theta := 1/2 - q(1-q/2)$, the 2-tournament EA with mutation rate χ/n and population size $\lambda > c\log(n/\chi)$ for a sufficiently large constant c achieves the optimum on* LEADINGONES *in the one-bit noise model (q) in expected time $O\left(n\lambda \log(n/\chi) + n^2/\chi\right)$.*

Proof We apply Theorem 3 to prove Theorem 5. If $\chi \in (0, \ln(1 + 2\theta))$, there exists a constant $\zeta \in (0, 1)$ such that $\chi \in (0, \ln(1 + 2\theta\zeta))$, which satisfies the condition in Theorem 3. We first partition the search space into levels. We use the partition $A_j := \{x \in \{0, 1\}^n | f(x) = j\}$ for all $j \in [0...n]$. By $q \in [0, 1)$, we know that $0 < \theta \le \frac{1}{2}$ which satisfies the assumption in Theorem 3.

By case (b) of Lemma 1, we get $\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) > 1/2 + \theta$, then condition (C2) of Theorem 3 holds.

To verify condition (C1), we need to estimate the probability of sampling individuals beyond the current level of the population. We assume that there is an individual $z \in A_j$ where $j \in [0...n-1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. We only consider the case that the first 0-bit is flipped and other bits are not flipped. By Lemma 5 it follows,

$$\Pr(y \in A_{\ge j+1} \mid z \in A_j) \ge \left(1 - \frac{\chi}{n}\right)^{n-1}\frac{\chi}{n} \ge e^{-\chi}(1 - o(1))\frac{\chi}{n}$$
$$=: h_j = \Omega\left(\chi/n\right). \tag{7}$$

Then we compute the population size required by condition (C3). Let $\xi \in (0, 1/16)$ be a constant, then

$$\lambda > \frac{4(1 + o(1))}{\theta^2\xi(1-\zeta)^4}\ln\left(\frac{128(m+1)}{\theta^2\xi(1-\zeta)^4\min\{h_j\}}\right) = O\left(\log(n/\chi)\right)$$

Condition (C3) is satisfied by $\lambda \ge c\log(n/\chi)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 3 hold and the expected time to reach the optimum is no more than

$$E[T] < \frac{16(1+o(1))}{\theta^2 \xi(1-\zeta)^2} \left(\lambda \sum_{j=0}^{m-1} \ln \left(\frac{6}{\xi(1-\zeta)^2 h_j} \right) + \frac{1}{\xi(1-\zeta)^2} \sum_{j=0}^{m-1} \frac{1}{h_j} \right)$$

$$= O \left(\lambda \sum_{j=0}^{m-1} \ln(n/\chi) + \sum_{j=0}^{m-1} n/\chi \right) = O \left(n\lambda \log(n/\chi) + n^2/\chi \right).$$

$\square$

4.2 Bit-Wise Noise Model

The best-known result on ONEMAX in the bit-wise noise model is that the (1+1) EA using a resampling strategy can achieve the optimum in expected polynomial time even if the noise level is extremely high, i.e. $p = 1/2 - 1/n^b$ for any constant $b > 0$ (see Table 3). By Theorem 6, we can compute that for extremely high-levels of bit-wise noise, the 2-tournament EA with mutation rate $\chi/n = \theta\zeta/n$ which is less than $\ln(1 + 2\theta\zeta)/n$ by Lemma 7, i.e., $\chi = d/n^{b+1/2}$ for some constant $d > 0$, and a sufficiently large population size $\lambda \in \Omega\left(n^{2b+1}\log(n)\right)$ has polynomial expected runtime $O\left(n^{2b+2}\lambda\log(n)\right)$ on ONEMAX. In contrast, the (1+1) EA cannot find the optimum in expected polynomial time if noise level $q \in \omega\left(\log(n)/n^2\right)$ (see Table 3).

Theorem 6 *For any $p \in (0, 1/2)$ and any $\chi \in (0, \ln(1 + 2\theta))$, where $\theta := 9(1/2 - p)/\left(64\sqrt{2pn} + 16\right)$, the 2-tournament EA with mutation rate χ/n and population size $\lambda > \frac{c(1+pn)}{(1-2p)^2} \log\left(\frac{n}{(1-2p)\chi}\right)$ for a sufficiently large constant c achieves the optimum on ONEMAX in the bit-wise noise model (p) in expected time $O\left(\frac{n(1+pn)}{(1-2p)^2} \left(\lambda \log\left(\frac{1}{\chi}\right) + \frac{\log(n)}{\chi}\right)\right)$.*

Proof We apply Theorem 3 to prove Theorem 6. If $\chi \in (0, \ln(1 + 2\theta))$, there exists a constant $\zeta \in (0, 1)$ such that $\chi \in (0, \ln(1 + 2\theta\zeta))$, which satisfies the condition in Theorem 3. We first partition the search space into levels. We use the partition $A_j := \{x \in \{0, 1\}^n | f(x) = j\}$ for all $j \in [0...n]$. Since $p \in (0, 1/2)$, we know that $0 < \theta < 9/32$ which satisfies the assumption in Theorem 3.

By case (c) of Lemma 1, we get $\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) > 1/2 + \theta$, then condition (C2) of Theorem 3 holds.

To verify condition (C1), we need to estimate the probability of sampling individuals beyond the current level of the population. We assume that there is an individual $z \in A_j$ where $j \in [0....n - 1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. We only consider the case that no 1-bits are flipped and one of the 0-bits is flipped after mutation and by Lemma 5,

$$\Pr(y \in A_{\geq j+1} \mid z \in A_j) > \left(1 - \frac{\chi}{n}\right)^j \frac{\chi}{n}(n - j)$$

$$\geq \left(1 - \frac{\chi}{n}\right)^n \frac{\chi}{n}(n - j) \geq e^{-\chi}\left(1 - \frac{\chi^2}{n}\right)\frac{\chi}{n}(n - j)$$

$$\geq e^{-\chi}(1 - o(1))(n - j)\chi/n =: h_j = \Omega\left((n - j)\chi/n\right)$$

since $e^{-\chi} \in \Omega(1)$ for any $\chi \in O(1)$.

Then we compute the population size required by condition (C3). Let $\xi \in (0, 1/16)$ be a constant, then

$$\lambda > \frac{4(1 + o(1))}{\theta^2\xi(1 - \zeta)^4}\ln\left(\frac{128(m + 1)}{\theta^2\xi(1 - \zeta)^4\min\{h_j\}}\right) = O\left(\frac{1}{\theta^2}\ln\left(\frac{n^2}{\chi\theta^2}\right)\right)$$

$$= O\left(\frac{\log(n/(\chi\theta))}{\theta^2}\right) = O\left(\frac{1 + pn}{(1 - 2p)^2}\log\left(\frac{n}{(1 - 2p)\chi}\right)\right).$$

Condition (C3) is satisfied by $\lambda \geq c\frac{1 + pn}{(1 - 2p)^2}\log\left(\frac{n}{(1 - 2p)\chi}\right)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 3 hold and the expected time to reach the optimum is no more than

$$E[T] \leq \frac{16(1 + o(1))}{\theta^2\xi(1 - \zeta)^2}\left(\lambda\sum_{j=0}^{m-1}\ln\left(\frac{6}{\xi(1 - \zeta)^2 h_j}\right) + \frac{1}{\xi(1 - \zeta)^2}\sum_{j=0}^{m-1}\frac{1}{h_j}\right)$$

$$= O\left(\frac{1}{\theta^2}\left(\lambda\sum_{j=0}^{m-1}\ln\left(\frac{n}{(n - j)\chi}\right) + \sum_{j=0}^{m-1}\frac{n}{(n - j)\chi}\right)\right)$$

$$= O\left(\frac{1}{\theta^2}\left(\lambda\ln\left(\frac{n^n}{\chi^n \cdot n!}\right) + \frac{n}{\chi}\log(n)\right)\right)$$

using the lower bound $n! > (n/e)^n$,

$$= O\left(\frac{1}{\theta^2}\left(n\lambda\log\left(\frac{1}{\chi}\right) + \frac{n}{\chi}\log(n)\right)\right)$$

$$= O\left(\frac{n(1 + pn)}{(1 - 2p)^2}\left(\lambda\log\left(\frac{1}{\chi}\right) + \frac{\log(n)}{\chi}\right)\right).$$

$\square$

For the LEADINGONES problem, we consider the case of the high bit-wise noise $p = b\log(n)/n$ for any constant $b > 0$. By Theorem 7, we can get that the 2-tournament EA with mutation rate $\chi/n = \theta\zeta/n$ which satisfies the condition, i.e., $\chi = d/n^{3b}$ for some constant $d > 0$, and a sufficiently large population size $\lambda \in \Omega\left(n^{6b}\log(n)\right)$ achieves the optimum on LEADINGONES in expected time $O\left(n^{6b+1}\lambda\log(n) + n^{9b+2}\right)$. In contrast, the expected runtime of the (1+1) EA with a resampling strategy is $12mn^{30b+1}$ under high bit-wise noise (see Table 4).

Theorem 7 *For any $p \in [0, 1/3)$ and any $\chi \in (0, \ln(1 + 2\theta))$, where $\theta :=$ $(1/2 - 3p/2)\, e^{-3np}$, the 2-tournament EA with mutation rate χ/n and population size $\lambda \geq c\frac{e^{6np}}{(1-3p)^2} \log\left(\frac{n}{\chi}\right)$ for a sufficiently large constant c achieves the optimum on* LEADINGONES *in the bit-wise noise model (p) in expected time* $O\left(\frac{ne^{6np}}{(1-3p)^2}\left(\lambda \log\left(\frac{n}{\chi}\right) + \frac{n}{\chi}\right)\right)$.

Proof We apply Theorem 3 to prove Theorem 7. If $\chi \in (0, \ln(1 + 2\theta))$, there exists a constant $\zeta \in (0, 1)$ such that $\chi \in (0, \ln(1 + 2\theta\zeta))$, which satisfies the condition in Theorem 3. We first partition the search space into levels. We use the partition $A_j := \{x \in \{0, 1\}^n \mid f(x) = j\}$ for all $j \in [0...n]$. Since $p \in [0, 1/3)$, we know that $0 < \theta \leq 1/2$ satisfies the assumption in Theorem 3.

By case (d) of Lemma 1, we get $\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) > 1/2 + \theta$, then condition (C2) of Theorem 3 holds.

To verify condition (C1), we need to estimate the probability of sampling individuals beyond the current level of the population. We assume that there is an individual $z \in A_j$ where $j \in [0....n - 1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. We only consider the case that the first 0-bit is flipped and other bits are not flipped, then by Lemma 5,

$$\Pr(y \in A_{\geq j+1} \mid z \in A_j) \geq \left(1 - \frac{\chi}{n}\right)^{n-1}\frac{\chi}{n} \geq e^{-\chi}(1 - o(1))\frac{\chi}{n} =: h_j$$
$$= \Omega\left(\chi/n\right).$$

Then we compute the population size required by condition (C3). Let $\xi \in (0, 1/16)$ be a constant, then

$$\lambda > \frac{4(1 + o(1))}{\theta^2\xi(1 - \zeta)^4}\ln\left(\frac{128(m + 1)}{\theta^2\xi(1 - \zeta)^4\min\{h_j\}}\right)$$
$$= O\left(\log(n/\chi)/\theta^2\right) = O\left(\frac{e^{6np}}{(1 - 3p)^2}\log\left(\frac{n}{\chi}\right)\right).$$

Condition (C3) is satisfied by $\lambda \geq c\frac{e^{6np}}{(1-3p)^2}\log\left(\frac{n}{\chi}\right)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 3 hold and the expected time to reach the optimum is no more than

$$E[T] < \frac{16(1 + o(1))}{\theta^2\xi(1 - \zeta)^2}\left(\lambda\sum_{j=0}^{m-1}\ln\left(\frac{6}{\xi(1 - \zeta)^2 h_j}\right) + \frac{1}{\xi(1 - \zeta)^2}\sum_{j=0}^{m-1}\frac{1}{h_j}\right)$$
$$= O\left(\frac{1}{\theta^2}\left(\lambda\sum_{j=0}^{m-1}\ln\left(\frac{n}{\chi}\right) + \sum_{j=0}^{m-1}\frac{n}{\chi}\right)\right)$$
$$= O\left(\frac{1}{\theta^2}\left(n\lambda\log\left(\frac{n}{\chi}\right) + \frac{n^2}{\chi}\right)\right)$$

$$= O\left(\frac{1}{\theta^2}\left(n\lambda \log(n/\chi) + \frac{n^2}{\chi}\right)\right)$$
$$= O\left(\frac{ne^{6np}}{(1-3p)^2}\left(\lambda \log\left(\frac{n}{\chi}\right) + \frac{n}{\chi}\right)\right).$$

$\square$

4.3 Gaussian Noise Model

Theorem 8 implies that the 2-tournament EA with mutation rate $\chi/n = \theta\zeta/n$, i.e., $\chi = d/\sigma$ for some constant $d > 0$, and a sufficiently large population size $\lambda \in \Omega\left(\sigma^2 \log(\sigma n)\right)$ can optimise ONEMAX and LEADINGONES in expected polynomial time, i.e., $O\left(\sigma^4 \log^2(n) + \sigma^3 n \log(n)\right)$ and $O\left(\sigma^4 \log^2(n) + \sigma^4 n^2\right)$ respectively, even if $\sigma^2 \in \text{poly}(n)$. Similarly to optimisation in the bit-wise noise model, the mutation rate should be fairly conservative, and the population size should be large enough if the noise level is extremely high, e.g., $\sigma = n^b$ for any constant $b > 0$. However, the (1+1) EA using a resampling strategy and EDAs can outperform the 2-tournament EA in these scenarios. It may be possible to increase the tournament size to achieve a better result.

Theorem 8 *For any $\sigma > 0$ and any $\chi \in (0, \ln(1 + 2\theta))$, where $\theta := 1/(6 + 48\sigma/\pi)$, the 2-tournament EA with mutation rate χ/n and population size $\lambda > c\sigma^2 \log(n/\chi)$ for a sufficiently large constant c achieves the optimum on ONEMAX and LEADINGONES in the Gaussian noise model (σ^2) in expected time $O\left(\sigma^2\lambda n \log(1/\chi) + \sigma^2 n \log(n)/\chi\right)$ and $O\left(\sigma^2\lambda n \log(n/\chi) + \sigma^2 n^2/\chi\right)$ respectively.*

Proof We apply Theorem 3 to prove Theorem 8. If $\chi \in (0, \ln(1 + 2\theta))$, there exists a constant $\zeta \in (0, 1)$ such that $\chi \in (0, \ln(1 + 2\theta\zeta))$, which satisfies the condition in Theorem 3. We first partition the search space into levels. We use the partition $A_j := \{x \in \{0, 1\}^n | f(x) = j\}$ for all $j \in [0...n]$. By $\sigma \in \text{poly}(n)$, we obtain $0 < \theta < \frac{1}{6}$ which satisfies the assumption in Theorem 3.

By case (e) of Lemma 1, we get $\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) > 1/2 + \theta$, then condition (C2) of Theorem 3 holds.

To verify condition (C1), we need to estimate the probability of sampling individuals beyond the current level of the population. We assume that there is an individual $z \in A_j$ where $j \in [0....n - 1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. We only consider the case that no 1-bits is flipped and one of the 0-bits is flipped after mutation for ONEMAX, then by Lemma 5,

$$\Pr(y \in A_{\geq j+1} \mid z \in A_j) > \left(1 - \frac{\chi}{n}\right)^j \frac{\chi}{n}(n - j)$$
$$\geq e^{-\chi}(1 - o(1))(n - j)\chi/n =: h_j$$

For LEADINGONES, we only consider the case that the first 0-bit is flipped and other bits are not flipped, then by Lemma 5,

$$\Pr(y \in A_{\geq j+1} \mid z \in A_j) \geq \left(1 - \frac{\chi}{n}\right)^{n-1} \frac{\chi}{n} \geq e^{-\chi}(1 - o(1)) \frac{\chi}{n} =: h_j.$$

Then, we get $h_j \in \Omega((n-j)\chi/n)$ and $h_j \in \Omega(\chi/n)$ for ONEMAX and LEADINGONES respectively.

Then we compute the population size required by condition (C3). Let $\xi \in (0, 1/16)$ be a constant and we know that $\min\{h_j\} \in \Omega(\chi/n)$ for both problems, then

$$\lambda > \frac{4(1 + o(1))}{\theta^2 \xi (1 - \zeta)^4} \ln \left(\frac{128(m+1)}{\theta^2 \xi (1 - \zeta)^4 \min\{h_j\}} \right) = O\left(\sigma^2 \log(n/\chi)\right)$$

Condition (C3) is satisfied by $\lambda \geq c\sigma^2 \log(n/\chi)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 3 hold and the expected time on ONEMAX to reach the optimum is no more than

$$E[T] \leq \frac{16(1 + o(1))}{\theta^2 \xi (1 - \zeta)^2} \left(\lambda \sum_{j=0}^{m-1} \ln \left(\frac{6}{\xi (1 - \zeta)^2 h_j} \right) + \frac{1}{\xi (1 - \zeta)^2} \sum_{j=0}^{m-1} \frac{1}{h_j} \right)$$

$$= O\left(\sigma^2 \left(\lambda \sum_{j=0}^{m-1} \ln \left(\frac{n}{(n-j)\chi} \right) + \sum_{j=0}^{m-1} \frac{n}{(n-j)\chi} \right) \right)$$

$$= O\left(\sigma^2 \left(\lambda \ln \left(\frac{n^n}{\chi^n n!} \right) + n \log(n)/\chi \right) \right)$$

using the lower bound $n! > (n/e)^n$,

$$= O\left(\sigma^2 \left(\lambda n \log(1/\chi) + n \log(n)/\chi \right) \right),$$

and on LEADINGONES,

$$E[T] = O\left(\sigma^2 \left(\lambda \sum_{j=0}^{m-1} \ln(\sigma n) + \sum_{j=0}^{m-1} \sigma n \right) \right)$$

$$= O\left(\sigma^2 \lambda n \log(n/\chi) + \sigma^2 n^2/\chi \right).$$

$\square$

4.4 Symmetric Noise Model

Resampling is a common method to cope with uncertainties [17, 21]. It dramatically improves the robustness of the (1+1) EA on ONEMAX and LEADINGONES in the one-

bit, the bit-wise and the Gaussian noise (see Tables 1-6). However, from Table 7, we know that the symmetric noise model (C, q) for any $C \in \mathbb{R}$ and $q = 1/2$ makes the resampling strategy inefficient, but using an elitist population can help. This section shows that non-elitist EAs also find the optimum in expected polynomial time if using an appropriate parameter setting. We also demonstrate a mutation rate *error threshold* as a function of the noise level in the symmetric noise model. Optimisation is efficient if the mutation rate is above the error threshold; otherwise inefficient.

4.4.1 Efficient Optimisation

Theorem 9 states that the 2-tournament EA with any mutation rate can optimise on noisy ONEMAX and LEADINGONES functions, for the noise level $q < 1/2$. Theorem 10 gives that the (μ, λ) EA can optimise in $O(n\lambda)$ time under all level noise if we set a sufficiently large population size, i.e., $\lambda \in \Omega(\log(n))$, a sufficiently large selective pressure, i.e., $\lambda/\mu > 1/((1-q)\zeta)$, and a low mutation rate $\chi/n \in \left(0, \ln\left(\frac{(1-q)\lambda}{(1+\delta)\mu}\right)/n\right)$ for any constant $\zeta, \delta \in (0, 1)$ and $\chi \in \Omega(1)$. This is due to insufficient selective pressure in 2-tournament selection in the high-level symmetric noise model. If we increase the tournament size, i.e., select $k > 2$ candidate individuals from P_t in Algorithm 2, it could be possible to optimise efficiently in such high level noisy functions, because the selective pressure of the non-elitist EA with k tournament selection is approximately k [51].

Theorem 9 *For any constant $q \in [0, 1/2)$, and $C \in \mathbb{R}$ and any $\chi \in (0, \ln(1 + 2\theta))$, where $\theta := 1/2 - q$, the 2-tournament EA with mutation rate χ/n and population size $\lambda > c\log(n)$ for a sufficiently large constant c achieves the optimum on ONEMAX and LEADINGONES in the symmetric noise model (C, q) in expected time $O(\lambda n \log(1/\chi) + n \log(n)/\chi)$ and $O\left(n\lambda \log(n/\chi) + n^2/\chi\right)$ respectively.*

Proof We apply Theorem 3 to prove Theorem 9. If $\chi \in (0, \ln(1 + 2\theta))$, there exists a constant $\zeta \in (0, 1)$ such that $\chi \in (0, \ln(1 + 2\theta\zeta))$, which satisfies the condition in Theorem 3. We first partition the search space into levels. We use the partition $A_j := \{x \in \{0, 1\}^n | f(x) = j\}$ for all $j \in [0...n]$. By $q \in [0, 1/2)$, we obtain $0 < \theta \leq \frac{1}{2}$ which satisfies the assumption in Theorem 3.

By case (f) of Lemma 1, we get $\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2)) = 1/2 + \theta$, then condition (C2) of Theorem 3 holds.

To verify condition (C1), we need to estimate the probability of sampling individuals beyond the current level of the population. We assume that there is an individual $z \in A_j$ where $j \in [0...n-1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. We only consider the case that no 1-bits is flipped and one of the 0-bits is flipped after mutation for ONEMAX, then by Lemma 5,

$$\Pr(y \in A_{\geq j+1} \mid z \in A_j) > \left(1 - \frac{\chi}{n}\right)^j \frac{\chi}{n}(n - j)$$
$$\geq e^{-\chi}(1 - o(1))(n - j)\chi/n =: h_j \in \Omega\left((n - j)\chi/n\right).$$

For LEADINGONES, we only consider the case that the first 0-bit is flipped and other bits are not flipped, then by Lemma 5,

$$\Pr(y \in A_{\geq j+1} \mid z \in A_j) \geq \left(1 - \frac{\chi}{n}\right)^{n-1} \frac{\chi}{n} \geq e^{-\chi}(1 - o(1))\frac{\chi}{n} =: h_j = \Omega(\chi/n).$$

Then, we get $h_j \in \Omega((n-j)\chi/n)$ and $h_j \in \Omega(\chi/n)$ for ONEMAX and LEADINGONES respectively.

Then we compute the population size required by condition (C3). Let $\xi \in (0, 1/16)$ be a constant and we know that $\min\{h_j\} \in \Omega(\chi/n)$ for both problems, then

$$\lambda > \frac{4(1+o(1))}{\theta^2 \xi(1-\zeta)^4} \ln\left(\frac{128(m+1)}{\theta^2 \xi(1-\zeta)^4 \min\{h_j\}}\right) = O(\log(n/\chi)).$$

Condition (C3) is satisfied by $\lambda \geq c \log(n/\chi)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 3 hold and the expected time on ONEMAX to reach the optimum is no more than

$$E[T] \leq \frac{16(1+o(1))}{\theta^2 \xi(1-\zeta)^2}\left(\lambda \sum_{j=0}^{m-1} \ln\left(\frac{6}{\xi(1-\zeta)^2 h_j}\right) + \frac{1}{\xi(1-\zeta)^2} \sum_{j=0}^{m-1} \frac{1}{h_j}\right)$$

$$= O\left(\lambda \sum_{j=0}^{m-1} \ln\left(\frac{n}{(n-j)\chi}\right) + \sum_{j=0}^{m-1} \frac{n}{(n-j)\chi}\right)$$

$$= O\left(\lambda \ln\left(\prod_{j=0}^{m-1} \frac{n}{(n-j)\chi}\right) + n \sum_{j=0}^{m-1} \frac{1}{(n-j)\chi}\right)$$

$$= O\left(\lambda \ln\left(\frac{n^n}{n!\chi^n}\right) + n \log(n)/\chi\right)$$

using the lower bound $n! > (n/e)^n$,

$$= O\left(\lambda n \log(1/\chi) + n \log(n)/\chi\right).$$

and on LEADINGONES,

$$E[T] \leq \frac{16(1+o(1))}{\theta^2 \xi(1-\zeta)^2}\left(\lambda \sum_{j=0}^{m-1} \ln\left(\frac{6}{\xi(1-\zeta)^2 h_j}\right) + \frac{1}{\xi(1-\zeta)^2} \sum_{j=0}^{m-1} \frac{1}{h_j}\right)$$

$$= O\left(\lambda \sum_{j=0}^{m-1} \ln\left(n/\chi\right) + \sum_{j=0}^{m-1} n/\chi\right)$$

$$= O\left(n\lambda \log\left(n/\chi\right) + n^2/\chi\right).$$

$\square$

Theorem 10 *For any constant $q \in [0, 1)$, and $C \leq 0$, any constant $\delta \in (0, 1)$ and any $\chi \in \left(0, \ln\left(\frac{(1-q)\lambda}{(1+\delta)\mu}\right)\right)$ and $\chi \in \Omega(1)$, the (μ, λ) EA with mutation rate χ/n, population size $\lambda > c \log(n)$ for a sufficiently large constant c and $(1-q)\lambda/(1+\delta) > \mu \in \Omega(\log(n))$ achieves the optimum on ONEMAX and LEADINGONES in the symmetric noise model (C, q) in expected time $O(\lambda n + n \log(n))$ and $O\left(n\lambda \log(n) + n^2\right)$ respectively.*

Proof We use the level-based theorem (Theorem 1) to prove Theorem 10. We use the partition $A_j := \{x \in \{0, 1\}^n | f(x) = j\}$ for all $j \in [0...n]$ and we define $\gamma_0 := \mu/\lambda$.

We first show that condition (G2) of Theorem 1 holds. If current level is $j \leq n - 2$, then there are at least $\gamma\lambda$ individuals in level $j + 1$. Let $\varepsilon := \delta^2/(1+\delta)$ be a constant. We assume that there are at least $\gamma(1-q)(1-\varepsilon)\lambda$ individuals which are in level $j+1$ and there is no noise when evaluating these individuals. We will verify this assumption later. Let z be the selected individual and y be the individual after mutating z, then

$$\Pr(y \in A_{\geq j+1}) \geq \Pr(z \in A_{\geq j+1}) \cdot \Pr(y \in A_{\geq j+1} | z \in A_{\geq j+1})$$
$$\geq \frac{\gamma\lambda(1-\varepsilon)(1-q)}{\mu}\left(1 - \frac{\chi}{n}\right)^n$$

by Lemma 3,

$$\geq \frac{\gamma\lambda(1-\varepsilon)(1-q)}{\mu}e^{-\chi}\left(1 - \frac{\chi^2}{n}\right)$$

by $\chi < \ln\left(\frac{(1-q)\lambda}{(1+\delta)\mu}\right)$,

$$> \gamma(1-\varepsilon)(1+\delta)(1-o(1))$$

by $\varepsilon = \delta^2/(1+\delta)$,

$$= \gamma(1+\delta-\delta^2)(1-o(1))$$
$$= \gamma\left(1 + \left(\delta - \delta^2\right)(1-o(1))\right)$$

To verify condition (G1), we need to estimate the probability of sampling individuals beyond the current level of the population if there are at least $\gamma_0\lambda$ individuals in level

$j \in [0....n-1]$. We assume that there is an individual $z \in A_j$ where $j \in [0....n-1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. We only consider the case that no 1-bit is flipped and one 0-bit is flipped after mutation for ONEMAX, then by Lemma 5,

$$
\begin{aligned}
\Pr(y \in A_{\geq j+1}) = \Pr(z \in A_{\geq j}) \cdot \Pr(y \in A_{\geq j+1} \mid z \in A_j) &> \left(1 - \frac{\chi}{n}\right)^j \frac{\chi}{n}(n-j) \\
&\geq e^{-\chi}(1 - o(1))(n-j)\chi/n =: z_j \in \Omega\left((n-j)\chi/n\right).
\end{aligned}
$$

For LEADINGONES, we only consider the case that the first 0-bit is flipped and no other bit is flipped, then by Lemma 5,

$$
\begin{aligned}
\Pr(y \in A_{\geq j+1}) = \Pr(z \in A_{\geq j}) \cdot \Pr(y \in A_{\geq j+1} \mid z \in A_j) &\geq \left(1 - \frac{\chi}{n}\right)^{n-1} \frac{\chi}{n} \\
&\geq e^{-\chi}(1 - o(1))\frac{\chi}{n} =: z_j = \Omega\left(1/n\right).
\end{aligned}
$$

Then, we get $z_j \in \Omega\left((n-j)/n\right)$ and $z_j \in \Omega\left(1/n\right)$ for ONEMAX and LEADINGONES respectively.

Then we compute the population size required by condition (G3). We know that $\min\{z_j\} \in \Omega\left(\chi/n\right)$ for both problems, then

$$
\lambda > \frac{4\lambda}{\left(\left(\delta - \delta^2\right)(1 - o(1))\right)^2 \mu} \ln\left(\frac{128(n+1)}{\left(\left(\delta - \delta^2\right)(1 - o(1))\right)^2 \min\{z_j\}}\right) = O\left(\log(n)\right).
$$

Condition (G3) is satisfied by $\lambda \geq c \log(n)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 1 hold and the expected time on ONEMAX to reach the optimum is no more than

$$
\begin{aligned}
&t_0(n) \\
&\leq \frac{8}{\left(\delta - \delta^2\right)(1 - o(1))^2}\left(\lambda \sum_{j=0}^{m-1} \ln\left(\frac{6\left(\left(\delta - \delta^2\right)(1 - o(1))\right)^2 \lambda}{4 + \left(\left(\delta - \delta^2\right)(1 - o(1))\right)^2 \lambda z_j}\right) + \sum_{j=0}^{m-1}\frac{1}{z_j}\right) \\
&= O\left(\lambda n + n\log(n)\right),
\end{aligned}
$$

and on LEADINGONES,

$$
= O\left(n\lambda \log(n) + n^2\right).
$$

Now we verify the assumption that there are at least $\gamma(1-q)(1-\varepsilon)$ individuals in level $j+1$ and the noise does not affect the ranking if the current level is $j+1$ for any $j \in [n-2]$. We refer to a sequence of $2t_0(n)/\lambda$ generations as a *phase*, and call a phase *good* if for $2t_0(n)/\lambda$ consecutive generations the assumption holds. Let $Z \sim$

$\text{Bin}\,(\gamma\lambda,\,(1-q))$ be a random variable, which represent the number of individuals not affected by noise in any generation $t \in \mathbb{N}$. By a Chernoff bound, the probability that the assumption holds in a generation is $\Pr\,(Z \leq \gamma\lambda(1-\varepsilon)(1-q)) \leq e^{-\Omega(\lambda)}$. By a union bound, a phase is good with probability $1 - (2t_0(n)/\lambda)\,e^{-\Omega(\lambda)} = \Omega(1)$. By Markov's inequality, the probability of reaching a global optimum in a good phase is at least $1 - \Pr\,(T \geq 2t_0(n)) \geq 1 - \frac{t_0(n)}{2t_0(n)} \geq 1 - \frac{1}{2} = \frac{1}{2}$. Hence, the expected number of phases required, each costing $2t_0(n)$ evolutions, is $O(1)$, and the theorem follows. $\qquad\square$

4.4.2 Inefficient Optimisation

Non-elitist EAs fail when the mutation rate becomes too high [37]. In this section, we investigate what mutation rate is too high for non-elitist EAs in uncertain settings. We use the negative drift theorem for populations to derive what mutation rate make non-elitist EAs inefficient on ONEMAX and LEADINGONES (shown in Theorems 11 and 12). For 2-tournament selection, there exists a mutation rate error threshold $\ln\,(2(1-q))\,/n$. Similarly, we can find a mutation rate error threshold in (μ, λ) selection, which is $\ln\,((1-q)\lambda/\mu)\,/n$. Without uncertainty, it is well-known that error thresholds of mutation rate is $\ln(2)/n$ and $\ln(\lambda/\mu)/n$ for the 2-tournament EA and the (μ, λ) EA, respectively [51]. As we can see from the proofs of Theorems 11 and 12, the presence of uncertainties can reduce the maximal reproductive rate of algorithms. Consequentially, the error threshold of the mutation rate would be reduced. We should reduce the mutation rate or increase the selection pressure, e.g., reduce μ in the (μ, λ) EA, as the uncertainty level is increased to ensure efficient optimisation.

Theorem 11 *For any $C \in \mathbb{R}$ and any constant $q \in [0, 1/2)$, the probability that the 2-tournament EA with any population size $\lambda = \mathrm{poly}(n)$, mutation rate $\chi/n > \ln\,(2(1-q)+o(1))\,/n$, optimises ONEMAX or LEADINGONES in the symmetric noise model (C, q) within e^{cn} generations is $e^{-\Omega(n)}$, for some constant $c > 0$.*

Proof We estimate the maximal reproductive rate under noise. In each generation of the 2-tournament EA, two individuals x_1 and x_2 are selected uniformly at random from the population by lines 1 and 2 of Algorithm 2, respectively. Then the fittest individual of x_1 and x_2 is chosen by fitness comparison (line 3). However, the presence of noise can lead to a failed comparison, i.e., the worse individual is selected in line 3 of Algorithm 2. we assume without loss of generality $f(x_1) > f(x_2)$. Let E be the event that $f^n(x_1) > f^n(x_2)$ or individual x_1 is selected uniformly from $\{x_1, x_2\}$ if $f^n(x_1) = f^n(x_2)$, then $\Pr(E) = \Pr\,(f^n(x_1) > f^n(x_2)) + \frac{1}{2}\Pr(f^n(x_1) = f^n(x_2))$. Then the maximal reproductive rate is the reproductive rate of the best individual, which is

$$\alpha_0 \leq E[R_t(i)] = \lambda\left(\frac{1}{\lambda^2} + \frac{2\Pr(E)}{\lambda}\left(1 - \frac{1}{\lambda}\right)\right) = \frac{1}{\lambda} + 2\Pr(E)\left(1 - \frac{1}{\lambda}\right)$$

$$< 2\Pr(E) + \frac{1}{\lambda} = 2\Pr(E) + o(1)$$

by Lemma 1 (f), $\Pr(E) = 1 - q$,

$$= 2(1 - q) + o(1).$$

Then Theorem 11 is proved by applying Theorem 2.   □

Theorem 12 *For any constant $q \in [0, 1]$, the probability that the (μ, λ) EA with any population size $\lambda = \mathrm{poly}(n)$, mutation rate $\chi/n > (\ln(1 - q)\lambda/\mu)/n$, optimises* ONEMAX *or* LEADINGONES *in the symmetric noise model (C, q) within e^{cn} generations is $e^{-\Omega(n)}$, for some constant $c > 0$.*

Proof We first compute the maximal reproductive rate. Let x be the fittest individual in P_t evaluated by $f(x)$. Only if there is no noise in x, x has a chance to be selected with probability $1/\mu$. Then we can compute the maximal reproductive rate $\alpha_0 \leq \frac{\lambda}{\mu}(1 - q) + 0 = \frac{\lambda}{\mu}(1 - q)$. Finally, Theorem 12 is proved by applying Theorem 2. □

5 Dynamic Optimisation

Now we consider dynamic optimisation. First, we apply the general theorem for the 2-tournament EA (Theorem 3) on the DYNBV problem and derive the runtime and the parameters required. The proof idea is similar to noisy optimisation in which the critical step is estimating a lower bound of the fitness bias.

Lengler and Schaller [30] proved that the (1+1) EA can achieve the optimum in $O(n \log(n))$ with standard mutation rate $\chi/n = 1/n$ on the noisy linear function which is a general case of the DYNBV problem. However, there only exists a partial result for population-based EAs, i.e., runtime when the population is initiated close to the optimum [15]. Theorem 13 gives for the first time the runtime from any start point on DYNBV for a population-based EA. It implies that if choosing a low mutation rate, e.g., $\chi/n = \zeta/(2n^2)$ and a population size $\lambda > cn^2 \log(n)$ for a sufficiently large constant c, the 2-tournament EA can optimise the DYNBV problem in $O(n^3\lambda \log(n))$ time. The analysis could be further improved by estimating the maximal Hamming-distance in all pairs of individuals more precisely.

Theorem 13 *For any $\chi \in (0, \ln(1 + 2\theta))$, where $\theta := 1/(2n)$, the 2-tournament EA with mutation rate χ/n and population size $\lambda > cn^2 \log(n)$ for a large enough constant c achieves the optimum on* DYNBV *in expected time $O\left(n^3\lambda \log(1/\chi) + n^3 \log(n)/\chi\right)$.*

Proof We apply Theorem 3 to prove Theorem 13. If $\chi \in (0, \ln(1 + 2\theta))$, there exists a constant $\zeta \in (0, 1)$ such that $\chi \in (0, \ln(1 + 2\theta\zeta))$, which satisfies the condition in Theorem 3. We first partition the search space into levels. We use the partition $A_j := \{x \in \{0, 1\}^n | \mathrm{OM}(x) = j\}$ for $j \in [0...n]$. It is easy to see that θ satisfies the assumption in Theorem 3.

We first show that condition (C2) of Theorem 3 holds. Let x_1 and x_2 be two individuals in $A_{\geq j+1}$ and $A_{\leq j}$ respectively, where $j \in [0...n-2]$. Let E be the event that $f^t(x_1) > f^t(x_2)$ or individual x_1 is selected uniformly from $\{x_1, x_2\}$ if $f^t(x_1) = f^t(x_2)$. The probability of this event is $\Pr(f^t(x_1) > f^t(x_2)) + \frac{1}{2}\Pr(f^t(x_1) = f^t(x_2)) = \Pr(E)$.

To estimate a lower bound for $\Pr(E)$ on DYNBV, we pessimistically assume that $x_1 \in A_{j+1}$ and $x_2 \in A_{\leq j}$, such that $\mathrm{OM}(x_1) = \mathrm{OM}(x_2) + h$ where $h \in [1, j]$. We assume $\mathrm{H}(x_1, x_2) \leq l + l + h = s$, where $s \leq n$ and there exist l bit-positions that x_1 has a 1-bit and x_2 has a 0-bit, and there exist another l bit-positions that x_1 is with 0-bit and x_2 has a 1-bit, such that x_1 and x_2 have the same bit in the rest of $n - 2l - h$ positions. For the DYNBV problem, the coefficients vary exponentially, thus the largest coefficient is the deciding factor for the fitness value. We first compare the fitness in the $n - 2l - h$ positions of x_1 and x_2, which are the same in the same generation. The next largest coefficient decides the final fitness value. Then we say that x_1 "wins" if the event E happens. If the next largest coefficient is in the $l + h$ positions, x_1 wins, else in another l positions, x_2 wins. Therefore,

$$\begin{aligned}
\Pr(E) &\geq \frac{l+h}{2l+h} = \frac{l + h/2 + h/2}{2(l + h/2)} = \frac{1}{2} + \frac{h}{2(2l+h)} \\
&\geq \frac{1}{2} + \frac{1}{2(2l+h)} = \frac{1}{2} + \frac{1}{2s}
\end{aligned}$$

since the Hamming-distance s between any pair of individuals is at most n, then

$$\geq \frac{1}{2} + \theta.$$

Condition (C2) of Theorem 3 holds. Since $s \leq n$, we get the fitness bias $\theta \geq 1/(2n)$.

To verify condition (C1), we need to estimate the probability of sampling individuals beyond the current level of the population. We assume that there is an individual $z \in A_j$ where $j \in [0...n-1]$, and let y be obtained from z by the mutation operator with mutation rate χ/n. For a lower bound, it suffices to only consider the case that none of the 1-bits are flipped and one of 0-bits is flipped after mutation. Then, by Lemma 5 it follows,

$$\begin{aligned}
\Pr(y \in A_{\geq j+1} \mid z \in A_j) &> \left(1 - \frac{\chi}{n}\right)^j \frac{\chi}{n}(n-j) \\
&\geq e^{-\chi}(1 - o(1))(n-j)\chi/n =: h_j \\
&= \Omega\left(\frac{(n-j)\chi}{n}\right).
\end{aligned}$$

Then we compute the population size required by condition (C3). Let $\xi \in (0, 1/16)$ be a constant, then

$$\lambda > \frac{4(1 + o(1))}{\theta^2 \xi (1 - \zeta)^4} \ln\left(\frac{128(m+1)}{\theta^2 \xi (1 - \zeta)^4 \min\{h_j\}}\right) = O\left(n^2 \log(n/\chi)\right).$$

Condition (C3) is satisfied by $\lambda > cn^2 \log(n/\chi)$ for a sufficiently large constant c.

Finally, all conditions of Theorem 3 hold and the expected time $E[T]$ to reach the optimum is no more than

$$E[T] \leq \frac{16\,(1+o(1))}{\theta^2 \xi (1-\zeta)^2} \left(\lambda \sum_{j=0}^{m-1} \ln \left(\frac{6}{\xi (1-\zeta)^2 h_j} \right) + \frac{1}{\xi (1-\zeta)^2} \sum_{j=0}^{m-1} \frac{1}{h_j} \right)$$

$$= O\left(s^2 \left(\lambda \sum_{j=0}^{m-1} \ln \left(\frac{n}{(n-j)\chi} \right) + \sum_{j=0}^{m-1} \frac{n}{(n-j)\chi} \right) \right)$$

$$= O\left(s^2 \left(\lambda \ln \left(\frac{n^n}{\chi^n n!} \right) + n \log(n)/\chi \right) \right)$$

using the bounds $n! > (n/e)^n$, and $s \leq n$,

$$= O\left(s^2 n \lambda \log(1/\chi) + s^2 n \log(n)/\chi \right)$$

$$= O\left(n^3 \lambda \log(1/\chi) + n^3 \log(n)/\chi \right).$$

$\square$

6 Conclusion

This paper has derived runtime results for non-elitists EAs with 2-tournament and (μ, λ) selection on two well-known benchmark functions, i.e., ONEMAX and LEADINGONES in uncertain environments. For the one-bit, the bit-wise and the Gaussian noise models, we improve and extend results of the non-elitist EA with 2-tournament from Dang and Lehre [22]. We introduce the notion of fitness bias which indicates the probability that the truly fitter individual is selected. We summarise fitness biases for 2-tournament selection in some noisy scenarios in Lemma 1. Then we get more precise upper bounds for the expected runtimes and also provide more precise guidance on how to choose the mutation rate and the population size as a function of the level of noise. From Tables 1, 2, 3, 4, 5, 6, 7 and 8, we conclude that by using an appropriate mutation parameter, i.e., $\chi \in [\theta, \ln(1+2\theta))$ where θ is a function of the level of noise, and a sufficiently large population size, the 2-tournament EA optimises ONEMAX and LEADINGONES in less time in expectation under one-bit and extremely high-level bit-wise noise, than the (1+1) EA using a resampling strategy [17, 21]. In some settings, such as in the Gaussian noise model, we obtain a lower upper bound of runtimes than the ACO-fp [19] and a comparable upper bound with EDAs [23, 26, 38].

We then, for the first time, studied the performance of non-elitist EAs with two selection mechanisms, i.e., 2-tournament and (μ, λ), on ONEMAX and LEADINGONES in the symmetric noise model. We also provide for the first time mutation rate error thresholds under the symmetric noise model, which are $\ln(2(1-q))/n$ and $\ln((1-q)\lambda/\mu)/n$ for the 2-tournament and the (μ, λ) selection, respectively. The noise essentially affects the maximal reproductive rate and the error threshold of a non-elitist population. Fur-

thermore, in these scenarios, non-elitist EAs can outperform the known best results, i.e., for the (1+λ) EA and the (μ+1) EA.

Finally, we prove for the first time that with appropriate parameter settings, non-elitist EAs can optimise the DYNBV problem in expected polynomial time. In overall, we provide advice on how to choose the mutation rate, the selective pressure and the population size (see Theorems 4–12), for non-elitist EAs with uncertain objectives. In general, non-elitist EAs tolerate high uncertainties when the mutation rate is sufficiently low, the selective pressure is sufficiently high, and the population size is sufficiently high, all relative to the level of uncertainty.

Although we can determine the appropriate mutation rate for a given level of uncertainty, this level is often unknown in real-world optimisation. Thus, future work should investigate the performance of mutation rate or selective pressure adjusting mechanisms, e.g., self-adaptation [46, 52–54] and self-adjusting [55–57] under uncertainties.

Appendix A Useful Inequalities

Lemma 2 $\ln(1 + x) < \sqrt{x}$ *for* $0 \leq x < \infty$.

Proof By Eq. (14) in [58],

$$\ln(1 + x) \leq \frac{x}{\sqrt{1 + x}} < \frac{x}{\sqrt{x}} = \sqrt{x}.$$

$\square$

Lemma 3 [59] $\left(1 + \frac{x}{n}\right)^n \geq e^x \left(1 - \frac{x^2}{n}\right)$ *for* $n \in \mathbb{N}^*, |x| \leq n$.

Lemma 4 $\left((1 - x)^{1/x - 1}\right)^y \geq e^{-y}$ *for* $0 < x < 1$ *and* $y > 0$.

Proof By Lemma 3,

$$(1 - x)^{1/x} \geq e^{-1} (1 - x) \tag{A1}$$

$$(1 - x)^{1/x - 1} \geq e^{-1} \tag{A2}$$

$$\left((1 - x)^{1/x - 1}\right)^y \geq e^{-y} \tag{A3}$$

$\square$

Lemma 5 $\left(1 - \frac{\chi}{n}\right)^i \frac{\chi}{n} \geq e^{-\chi} (1 - o(1)) \frac{\chi}{n}$ *for* $0 < \chi = O(1), n \in \mathbb{N}^*$ *and* $0 \leq i \leq n$.

Proof

$$\left(1 - \frac{\chi}{n}\right)^i \frac{\chi}{n} \geq \left(1 - \frac{\chi}{n}\right)^n \frac{\chi}{n}$$

by Lemma 3,

$$\geq e^{-\chi} \left(1 - \frac{\chi^2}{n}\right) \frac{\chi}{n} = e^{-\chi} (1 - o(1)) \chi/n$$

□

Lemma 6 [22] *Let* $F(x)$ *be the cumulative density function of normal distribution* $\mathcal{N}\left(0, \sigma^2\right)$, *for* $x > 0$ *we have*

$$F(x) > 1 - \frac{1}{\sqrt{\frac{x\pi}{\sigma\sqrt{2}} + 4}}$$

Lemma 7 *For any* $\theta \in (0, 1/2]$ *and any constant* $\zeta \in (0, 1)$, $\theta\zeta < \ln(1 + 2\theta\zeta)$.

Proof By $\frac{2x}{2+x} \leq \ln(1 + x)$ from Eq. (3) in [58], we obtain

$$\ln(1 + 2\theta\zeta) \geq \frac{4\theta\zeta}{2 + 2\theta\zeta} = \frac{2\theta\zeta}{1 + \theta\zeta} > \frac{2\theta\zeta}{2} = \theta\zeta.$$

□

Lemma 8 [32] *Let X and Y be identically distributed independent random variables with integer support, finite expected value* μ *and finite non-zero variance* σ^2, *it holds that*

$$\Pr(X = Y) \geq \frac{\left(1 - 1/d^2\right)^2}{2d\sigma + 1} \text{ for any } d \geq 1$$

References

1. Lehre, P.K., Qin, X.: More precise runtime analyses of non-elitist EAs in uncertain environments. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1160–1168. ACM, Lille (2021)
2. Moriarty, D.E., Schultz, A.C., Grefenstette, J.J.: Evolutionary algorithms for reinforcement learning. J. Artif. Intell. Res. **11**, 241–276 (1999)
3. Fleming, P.J., Purshouse, R.C.: Evolutionary algorithms in control systems engineering: a survey. Control Eng. Pract. **10**(11), 1223–1241 (2002)
4. Freitas, A.A.: A survey of evolutionary algorithms for data mining and knowledge discovery. In: Advances in Evolutionary Computing, pp. 819–845. Springer, Berlin, Heidelberg (2003)
5. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., de Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) 39(2), 133–155 (2009)

6. Barros, R.C., Basgalupp, M.P., de Carvalho, A.C.P.L.F., Freitas, A.A.: A survey of evolutionary algorithms for decision-tree induction. IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) 42(3), 291–312 (2012)

7. Slowik, A., Kwasnicka, H.: Evolutionary algorithms and their applications to engineering problems. Neural Comput. Appl. **32**(16), 12363–12379 (2020)

8. He, X., Zhao, K., Chu, X.: AutoML: a survey of the state-of-the-art. Knowl.-Based Syst. **212**, 106622 (2021)

9. Droste, S., Jansen, T., Wegener, I.: Upper and lower bounds for randomized search heuristics in black-box optimization. Theory Comput. Syst. **39**(4), 525–544 (2006)

10. Hausmann, D., Korte, B.: Lower bounds on the worst-case complexity of some oracle algorithms. Discrete Math. **24**(3), 261–276 (1978). Publisher: Elsevier

11. Winzen, C.: Toward a complexity theory for randomized search heuristics: black-box models. Ph.D Thesis, Saarland University, Saarbrücken (2011)

12. Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., Sculley, D.: Google vizier: a service for black-box optimization. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1487–1495. ACM, Halifax (2017)

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

14. Aishwaryaprajna, Rowe, J.E.: Evolutionary algorithms for solving unconstrained, constrained and multi-objective noisy combinatorial optimisation problems. arXiv:2110.02288 [cs] (2021)

15. Lengler, J., Meier, J.: Large population sizes and crossover help in dynamic environments. In: Parallel Problem Solving from Nature—PPSN XVI, pp. 610–622. Springer, Cham (2020)

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

17. 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**(2), 749–795 (2019)

18. Sudholt, D.: Analysing the Robustness of Evolutionary Algorithms to Noise: Refined Runtime Bounds and an Example Where Noise is Beneficial. Algorithmica **83**, 976–1011 (2021)

19. Friedrich, T., Kötzing, T., Krejca, M.S., Sutton, A.M.: Robustness of ant colony optimization to noise. Evolut. Comput. **24**(2), 237–254 (2016). Publisher: MIT Press

20. Lehre, P.K., Nguyen, P.T.H.: Runtime analysis of the univariate marginal distribution algorithm under low selective pressure and prior noise. In: Proceedings of the genetic and evolutionary computation conference, pp. 1497–1505. ACM, Prague Czech Republic (2019)

21. 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**, 237–267 (2018)

22. Dang, D.-C., Lehre, P.K.: Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In: Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII—FOGA '15, pp. 62–68. ACM Press, Aberystwyth (2015)

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

24. Qian, C., Bian, C., Yu, Y., Tang, K., Yao, X.: Analysis of noisy evolutionary optimization when sampling fails. Algorithmica **83**(4), 940–975 (2021)

25. Friedrich, T., Kötzing, T., Krejca, M., Sutton, A.M.: The benefit of recombination in noisy evolutionary search. In: Algorithms and Computation—26th International Symposium. ISAAC 2015, Nagoya, Japan, December 9–11, 2015, Proceedings, vol. 9472, pp. 140–150. Springer, Nagoya (2015)

26. Rowe, J.E., Aishwaryaprajna: The benefits and limitations of voting mechanisms in evolutionary optimisation. In: Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms—FOGA '19, pp. 34–42. ACM Press, Potsdam (2019)

27. Rohlfshagen, P., Lehre, P.K., Yao, X.: Dynamic evolutionary optimisation: an analysis of frequency and magnitude of change. Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 1713–1720. ACM, Montreal (2009)

28. Kötzing, T., Lissovoi, A., Witt, C.: $(1 + 1)$ EA on generalized dynamic OneMax. In: Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII, pp. 40–51. ACM, Aberystwyth (2015)

29. Lissovoi, A., Witt, C.: Runtime analysis of ant colony optimization on dynamic shortest path problems. Theoret. Comput. Sci. **561**, 73–85 (2015)

30. Lengler, J., Schaller, U.: The $(1+1)$-EA on noisy linear functions with random positive weights. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 712–719 (2018)
31. Lengler, J., Riedi, S.: Runtime analysis of the $(\mu+1)$-EA on the dynamic BinVal function. In: Zarges, C., Verel, S. (eds.) Evolutionary Computation in Combinatorial Optimization, pp. 84–99. Springer, Cham (2021)
32. Dang, D.-C., Lehre, P.K.: Runtime analysis of non-elitist populations: from classical optimisation to partial information. Algorithmica **75**(3), 428–461 (2016)
33. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. Algorithmica **64**(4), 673–697 (2012)
34. Droste, S.: Analysis of the $(1+1)$ EA for a noisy OneMax. In: Genetic and Evolutionary Computation—GECCO 2004, vol. 3102, pp. 1088–1099. Springer, Berlin, Heidelberg (2004)
35. Prugel-Bennett, A., Rowe, J., Shapiro, J.: Run-time analysis of population-based evolutionary algorithm in noisy environments. In: Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII—FOGA '15, pp. 69–75. ACM Press, Aberystwyth (2015)
36. Corus, D., Dang, D.-C., Eremeev, A.V., Lehre, P.K.: Level-based analysis of genetic algorithms and other search processes. IEEE Trans. Evol. Comput. **22**(5), 707–719 (2018)
37. Lehre, P.K.: Negative drift in populations. In: Parallel Problem Solving from Nature. PPSN XI, pp. 244–253. Springer, Berlin, Heidelberg (2010)
38. Lehre, P.K., Nguyen, P.T.H.: Runtime analyses of the population-based univariate estimation of distribution algorithms on LeadingOnes. Algorithmica **83**(10), 3238–3280 (2021)
39. Lehre, P.K.: Fitness-levels for non-elitist populations. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation—GECCO '11, p. 2075. ACM Press, Dublin (2011)
40. Eremeev, A.V.: Runtime analysis of non-elitist evolutionary algorithms with fitness-proportionate selection on royal road functions. In: 2020 Cognitive Sciences. Genomics and Bioinformatics (CSGB), pp. 228–232. IEEE, Novosibirsk (2020)
41. Doerr, B.: Does comma selection help to cope with local optima? In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, pp. 1304–1313. ACM, Cancún (2020)
42. Zheng, W., Zhang, Q., Chen, H., Yao, X.: When non-elitism meets time-linkage problems. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 741–749. ACM, Lille (2021)
43. Oliveto, P.S., Paixão, T., Pérez Heredia, J., Sudholt, D., Trubenová, B.: When non-elitism outperforms elitism for crossing fitness valleys. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, pp. 1163–1170. ACM, Denver (2016)
44. Dang, D.-C., Eremeev, A., Lehre, P.K.: Escaping local optima with non-elitist evolutionary algorithms. In: Proceedings of AAAI 2021. AAAI Press, Palo Alto (2020)
45. Dang, D.-C., Eremeev, A., Lehre, P.K.: Non-elitist evolutionary algorithms excel in fitness landscapes with sparse deceptive regions and dense valleys. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, Lille (2021)
46. Dang, D.-C., Lehre, P.K.: Self-adaptation of mutation rates in non-elitist populations. In: Parallel Problem Solving from Nature—PPSN XIV, vol. 9921, pp. 803–813. Springer, Cham (2016)
47. Goldberg, D.E., Deb, K.: A Comparative analysis of selection schemes used in genetic algorithms. In: Foundations of Genetic Algorithms vol. 1, pp. 69–93. Elsevier (1991)
48. Doerr, B., Kötzing, T.: Multiplicative Up-Drift. Algorithmica **83**, 3017–3058 (2021)
49. Doerr, B.: Lower bounds for non-elitist evolutionary algorithms via negative multiplicative drift. In: Parallel Problem Solving from Nature—PPSN XVI, vol. 12270, pp. 604–618. Springer, Cham (2020). Series Title: Lecture Notes in Computer Science
50. Dang, D.-C., Lehre, P.K., Nguyen, P.T.H.: Level-based analysis of the univariate marginal distribution algorithm. Algorithmica **81**(2), 668–702 (2019)
51. Lehre, P.K., Yao, X.: On the impact of mutation-selection balance on the runtime of evolutionary algorithms. IEEE Trans. Evol. Comput. **16**(2), 225–241 (2012)
52. Case, B., Lehre, P.K.: Self-adaptation in non-elitist evolutionary algorithms on discrete problems with unknown structure. IEEE Trans. Evolut. Comput. **24**, 650–663 (2020)
53. Lehre, P.K., Qin, X.: Self-adaptation via multi-objectivisation: a theoretical study. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1417–1425. ACM, Boston (2022)
54. Qin, X., Lehre, P.K.: Self-adaptation via multi-objectivisation: an empirical study. In: Parallel Problem Solving from Nature—PPSN XVII. Springer, Cham (2022)
55. Doerr, B., Doerr, C., Lengler, J.: Self-adjusting mutation rates with provably optimal success rules. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1479–1487. ACM, Prague Czech Republic (2019)

56. Hevia Fajardo, M.A., Sudholt, D.: Self-adjusting offspring population sizes outperform fixed parameters on the cliff function. In: Proceedings of the 16th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, pp. 1–15. ACM, Virtual Event Austria (2021)

57. Hevia Fajardo, M.A.H., Sudholt, D.: Self-adjusting population sizes for non-elitist evolutionary algorithms: why success rates matter. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1151–1159. ACM, Lille (2021)

58. Topsoe, F.: Some bounds for the logarithmic function. Inequal. Theory Appl. **4**, 137 (2007)

59. Niculescu, C.P., Vernescu, A.: A two sided estimate of $e^x - (1 + x/n)^n$. J. Inequal. Pure Appl. Math. **5**(3), 55 (2004)