On the Effectiveness of Sampling for Evolutionary Optimization in Noisy Environments

Chao Qian[1,2] chaoqian@ustc.edu.cn
Yang Yu[2] yuy@nju.edu.cn
Ke Tang[1] ketang@ustc.edu.cn
Yaochu Jin[3] yaochu.jin@surrey.ac.uk
Xin Yao[1,4] x.yao@cs.bham.ac.uk
Zhi-Hua Zhou[2*] zhouzh@nju.edu.cn
[1]USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications, School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230027, China
[2]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China
[3]Department of Computer Science, University of Surrey, Guildford, GU2 7XH, UK
[4]Center of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK

Abstract

In real-world optimization tasks, the objective (i.e., fitness) function evaluation is often disturbed by noise due to a wide range of uncertainties. Evolutionary algorithms are often employed in noisy optimization, where reducing the negative effect of noise is a crucial issue. Sampling is a popular strategy for dealing with noise: to estimate the fitness of a solution, it evaluates the fitness multiple (k) times independently and then uses the sample average to approximate the true fitness. Obviously, sampling can make the fitness estimation closer to the true value, but also increases the estimation cost. Previous studies mainly focused on empirical analysis and design of efficient sampling strategies, while the impact of sampling is unclear from a theoretical viewpoint. In this paper, we show that sampling can speed up noisy evolutionary optimization exponentially via rigorous running time analysis. For the (1+1)-EA solving the OneMax and the LeadingOnes problems under prior (e.g., one-bit) or posterior (e.g., additive Gaussian) noise, we prove that, under a high noise level, the running time can be reduced from exponential to polynomial by sampling. The analysis also shows that a gap of one on the value of k for sampling can lead to an exponential difference on the expected running time, cautioning for a careful selection of k. We further prove by using two illustrative examples that sampling can be more effective for noise handling than parent populations and threshold selection, two strategies that have shown to be robust to noise. Finally, we also show that sampling can be ineffective when noise does not bring a negative impact.

Keywords

Robust optimization, optimization in noisy environments, evolutionary algorithms, running time analysis, computational complexity.

* Corresponding author

1 Introduction

In many real-world optimization tasks, the exact objective (i.e., fitness) evaluation of candidate solutions is almost impossible, while we can obtain only a noisy one. Evolutionary algorithms (EAs) (Bäck, 1996) are general-purpose optimization algorithms inspired from natural phenomena, and have been widely and successfully applied to solve noisy optimization problems (Jin and Branke, 2005; Bianchi et al., 2009; Zeng et al., 2015). During evolutionary optimization, handling noise in fitness evaluation is very important, since noise may mislead the search direction and then deteriorate the efficiency of EAs. Many studies thus have focused on reducing the negative effect of noise in evolutionary optimization (Arnold, 2002; Beyer, 2000; Jin and Branke, 2005).

One popular way to cope with noise in fitness evaluation is sampling (Arnold and Beyer, 2006), which, instead of evaluating the fitness of one solution only once, evaluates the fitness k times and then uses the average to approximate the true fitness. Sampling obviously can reduce the standard deviation of the noise by a factor of $\sqrt{k}$, while also increasing the computation cost k times. This makes the fitness estimation closer to the true value, but computationally more expensive. In order to reduce the sampling cost as much as possible, many smart sampling approaches have been proposed, including adaptive (Aizawa and Wah, 1994; Stagge, 1998) and sequential (Branke and Schmidt, 2003, 2004) methods, which dynamically decide the size of k for each solution in each generation.

The impact of sampling on the convergence of EAs in noisy optimization has been empirically and theoretically investigated (Gutjahr, 2003; Arnold and Beyer, 2006; Heidrich-Meisner and Igel, 2009; Rolet and Teytaud, 2010). On the running time, a more practical performance measure for how soon an algorithm can solve a problem, previous experimental studies have reported conflicting conclusions. In (Aizawa and Wah, 1994), it was shown that sampling can speed up a standard genetic algorithm on two test functions; while in (Cantú-Paz, 2004), sampling led to a larger computation time for a simple generational genetic algorithm on the OneMax function. However, little work has been done on theoretically analyzing the impact of sampling on the running time. Thus, there are many fundamental theoretical issues on sampling that have not been addressed, e.g., if sampling can reduce the running time of EAs from exponential to polynomial in noisy environments, and if sampling will increase the running time in some cases.

The running time is usually counted by the number of fitness evaluations needed to find an optimal solution for the first time, because the fitness evaluation is deemed as the most costly computational process (Droste et al., 2002; Yu and Zhou, 2008; Qian et al., 2015b). Rigorous running time analysis has been a leading theoretical aspect for randomized search heuristics (Neumann and Witt, 2010; Auger and Doerr, 2011). Recently, progress has been made on the running time analysis of EAs. Numerous analytical results for EAs solving synthetic problems as well as combinatorial problems have been reported, e.g., (Neumann and Witt, 2010; Auger and Doerr, 2011). Meanwhile, general running time analysis approaches have also been proposed, e.g., drift analysis (He and Yao, 2001; Doerr et al., 2012b; Doerr and Goldberg, 2013), fitness-level methods (Wegener, 2002; He and Yao, 2003; Sudholt, 2013; Dang and Lehre, 2015b), and switch analysis (Yu et al., 2015; Yu and Qian, 2015). However, most of them focus on noise-free environments, where the fitness evaluation is exact.

For EAs in noisy environments, few results have been reported on running time analysis. Droste (2004) first analyzed the (1+1)-EA on the OneMax problem in the presence of one-bit noise and showed the maximal noise level $\log(n)/n$ allowing a polynomial running time, where the noise level is characterized by the noise probability $p \in [0,1]$ and n is the problem size. This result was later extended to the LeadingOnes problem and to many different noise models in (Gießen and Kötzing, 2016), which also proved that small populations of size $\Theta(\log n)$ can make elitist EAs i.e., (μ+1)-EA and (1+λ)-EA, perform well in high noise levels. The robustness of populations to noise was also proved in the setting of non-elitist EAs with mutation only (Dang and Lehre, 2015a) or uniform crossover only (Prugel-Bennett et al., 2015). However, Friedrich et al. (2015) showed the limitation of parent populations to cope with noise by proving that the (μ+1)-EA needs super-polynomial time for solving OneMax in the presence of additive Gaussian noise $\mathcal{N}(0,\sigma^2)$ with $\sigma^2 \geq n^3$. This difficulty can be overcome by the compact genetic algorithm (cGA) (Friedrich et al., 2015) and a simple Ant Colony Optimization (ACO) algorithm (Friedrich et al., 2016), both of which find the optimal solution in polynomial time with a high probability. Recently, Qian et al. (2015a) proved that the threshold selection strategy is also robust to noise: the expected running time of the (1+1)-EA using threshold selection on OneMax in the presence of one-bit noise is always polynomial regardless of the noise level. They also showed the limitation of threshold selection under asymmetric one-bit noise and further proposed smooth threshold selection, which can overcome the difficulty. Note that there was also a sequence of papers analyzing the running time of ACO on single destination shortest paths (SDSP) problems with edge weights disturbed by noise (Sudholt and Thyssen, 2012; Doerr et al., 2012a; Feldmann and Kötzing, 2013).

In addition to the above results, there exist two other pieces of work on running time analysis in noisy evolutionary optimization that involve sampling. Akimoto et al. (2015) proved that sampling with a large enough k can make optimization under additive unbiased noise behave as optimization in a noise-free environment, and thus concluded that noisy optimization using sampling can be solved in $k*r$ running time, where r is the noise-free running time. A similar result was also achieved for an adaptive Pareto sampling (APS) algorithm solving bi-objective optimization problems under additive Gaussian noise $\mathcal{N}(0,\sigma^2)$ (Gutjahr, 2012). These results, however, do not describe any impact of sampling on the running time, because they do not compare the running time in noisy optimization without sampling.

In this paper, we show that sampling can speed up noisy evolutionary optimization exponentially via rigorous running time analysis. For the (1+1)-EA solving the OneMax and the LeadingOnes problems under prior (e.g., one-bit) or posterior (e.g., additive Gaussian) noise, we prove that the running time is exponential when the noise level is high (i.e., Theorems 1, 4, 6, 7), while sampling can reduce the running time to be polynomial (i.e., Theorems 3, 5, Corollaries 1, 2). Particularly, for the (1+1)-EA solving OneMax under one-bit noise with $p = 1$, the analysis also shows that a gap of one on the value of k for sampling can lead to an exponential difference on the expected running time (i.e., Theorems 2, 3), which reveals that a careful selection of k is important for the effectiveness of sampling.

As previous studies (Qian et al., 2015a; Gießen and Kötzing, 2016) have shown that parent populations and threshold selection can bring about robustness to noise, we also compare sampling with these two strategies. On the OneMax problem under additive Gaussian noise $\mathcal{N}(0,\sigma^2)$ with $\sigma^2 \geq n^3$, the (μ+1)-EA needs super-polynomial time (Friedrich et al., 2015) (i.e., Theorem 8), while the (1+1)-EA using sampling can solve the problem in polynomial time (i.e., Corollary 1). On the OneMax problem under asymmetric one-bit noise with $p = 1$, the (1+1)-EA using threshold selection needs at least exponential time (Qian et al., 2015a) (i.e., Theorem 9), while the (1+1)-EA using sampling can solve it in $O(n \log^2 n)$ time (i.e., Theorem 10). Therefore, these results show that sampling can be more tolerant of noise than parent populations and threshold selection, respectively.

Finally, for the (1+1)-EA solving the Trap problem under additive Gaussian noise, we prove that noise does not bring a negative impact. Under the assumption that the positive impact of noise increases with the noise level, we conjecture that sampling is ineffective in this case since it will decrease the noise level. The conjecture is verified by experiments. Note that the conjecture is consistent with that in (Qian et al., 2015a). In that work it is hypothesized that the impact of noise is correlated with the problem hardness: when the problem is EA-hard (He and Yao, 2004) w.r.t. a specific EA (e.g., the Trap problem for the (1+1)-EA), noise can be helpful and does not need to be handled, but when the problem is EA-easy (He and Yao, 2004), noise can be harmful and needs to be tackled.

This paper extends our preliminary work (Qian et al., 2014) and improves one previous statement. In (Qian et al., 2014), we proved a sufficient condition under which sampling is ineffective, and applied it to the cases that the (1+1)-EA solving OneMax and Trap under additive Gaussian noise. The proof assumed the monotonicity of a quantity. By finding that an upper/lower-bound of the quantity is monotonic, we hypothesized that the quantity itself is also monotonic. Considering that this property does not always hold, we have corrected our previous statement on the OneMax problem by proving that sampling with a moderate sample size is possible to exponentially reduce the running time of the (1+1)-EA from no sampling (i.e., Theorem 6, Corollary 1). Meanwhile, both analysis and experiments (i.e., Section 6) show that sampling is ineffective on the Trap problem.

The rest of this paper is organized as follows. Section 2 introduces some preliminaries. The robustness analysis of sampling to prior and posterior noise is presented in Sections 3 and 4, respectively. Section 5 compares sampling with the other two strategies, parent populations and threshold selection, on the robustness to noise. Section 6 gives a case where sampling is ineffective. Section 7 concludes the paper.

2 Preliminaries

In this section, we first introduce the noise models, problems and evolutionary algorithms studied in this paper, respectively, then describe the sampling strategy, and finally present the analysis tools that we use throughout this paper.

2.1 Noise Models

Noise models can be generally divided into two categories: prior and posterior (Jin and Branke, 2005; Gießen and Kötzing, 2016). For prior noise, the noise comes from the variation on a solution instead of the evaluation process. One-bit noise as presented in Definition 1 is a representative one, which flips a random bit of a solution before evaluation with probability p. For posterior noise, the noise comes from the variation on the fitness of a solution. A representative model is additive Gaussian noise as presented in Definition 2, which adds a value drawn from a Gaussian distribution. Both one-bit noise and additive Gaussian noise have been widely used in previous empirical and theoretical studies, e.g., (Beyer, 2000; Droste, 2004; Jin and Branke, 2005; Gießen and Kötzing, 2016). In this paper, we will also use these two kinds of noise models.

Definition 1 (One-bit Noise)**.** *Given a parameter $p \in [0,1]$, let $f^n(x)$ and $f(x)$ denote the noisy and true fitness of a binary solution $x \in \{0,1\}^n$, respectively, then*

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

where x' is generated by flipping a uniformly randomly chosen bit of x.

Definition 2 (Additive Gaussian Noise)**.** *Given a Gaussian distribution $\mathcal{N}(\theta, \sigma^2)$, let $f^n(x)$ and $f(x)$ denote the noisy and true fitness of a solution x, respectively, then*

$$f^n(x) = f(x) + \delta,$$

where δ is randomly drawn from $\mathcal{N}(\theta, \sigma^2)$, denoted by $\delta \sim \mathcal{N}(\theta, \sigma^2)$.

In addition to the above noises, we also consider a variant of one-bit noise called asymmetric one-bit noise (Qian et al., 2015a), in Definition 3. For the flipping of asymmetric one-bit noise on a solution $x \in \{0,1\}^n$, if $|x|_0 = 0$, a random 1 bit is flipped; if $|x|_0 = n$, a random 0 bit is flipped; otherwise, the probability of flipping a specific 0 bit is $\frac{1}{2} \cdot \frac{1}{|x|_0}$, and the probability of flipping a specific 1 bit is $\frac{1}{2} \cdot \frac{1}{n-|x|_0}$, where $|x|_0 = n - \sum_{i=1}^{n} x_i$ is the number of 0-bits of x. Note that for one-bit noise, the probability of flipping any specific bit is $\frac{1}{n}$.

Definition 3 (Asymmetric One-bit Noise)**.** *Given a parameter $p \in [0,1]$, let $f^n(x)$ and $f(x)$ denote the noisy and true fitness of a binary solution $x \in \{0,1\}^n$, respectively, then $f^n(x) = f(x)$ with probability $(1-p)$, otherwise $f^n(x) = f(x')$, where x' is generated by flipping the j-th bit of x, and j is a uniformly randomly chosen position of*

$$\begin{cases} \textit{all bits of } x, & \textit{if } |x|_0 = 0 \textit{ or } n; \\ \begin{cases} \textit{0 bits of } x, & \textit{with probability } 1/2; \\ \textit{1 bits of } x, & \textit{with probability } 1/2. \end{cases}, & \textit{otherwise.} \end{cases}$$

2.2 Optimization Problems

As most theoretical analyses of EAs start from simple synthetic problems, we also use two well-known test functions OneMax and LeadingOnes, which have been widely studied in both noise-free (e.g., (He and Yao, 2001; Droste et al., 2002; Sudholt, 2013)) and noisy (e.g., (Droste, 2004; Dang and Lehre, 2015a; Gießen and Kötzing, 2016)) evolutionary optimization.

The OneMax problem as presented in Definition 4 aims to maximize the number of 1-bits of a solution. Its optimal solution is $11\ldots1$ (briefly denoted as 1^n) with the function value n. It has been shown that the expected running time of the (1+1)-EA on OneMax is $\Theta(n \log n)$ (Droste et al., 2002).

Definition 4 (OneMax)**.** *The OneMax Problem of size n is to find an n bits binary string x^* such that*

$$x^* = \arg\max\nolimits_{x \in \{0,1\}^n} \left(f(x) = \sum\nolimits_{i=1}^{n} x_i \right).$$

The LeadingOnes problem as presented in Definition 5 aims to maximize the number of consecutive 1-bits counting from the left of a solution. Its optimal solution is 1^n with the function value n. It has been proved that the expected running time of the (1+1)-EA on LeadingOnes is $\Theta(n^2)$ (Droste et al., 2002).

Definition 5 (LeadingOnes)**.** *The LeadingOnes Problem of size n is to find an n bits binary string x^* such that*

$$x^* = \arg\max_{x \in \{0,1\}^n} \left(f(x) = \sum\nolimits_{i=1}^{n} \prod\nolimits_{j=1}^{i} x_j \right).$$

We will also use an EA-hard problem Trap in Definition 6, the aim of which is to maximize the number of 0-bits of a solution except for the optimal solution 1^n. Its optimal function value is $C - n > 0$, and the function value for any non-optimal solution is not larger than 0. The expected running time of the (1+1)-EA on Trap has been proven to be $\Theta(n^n)$ (Droste et al., 2002).

Definition 6 (Trap)**.** *The Trap Problem of size n is to find an n bits binary string x^* such that, let $C > n$,*

$$x^* = \arg\max_{x \in \{0,1\}^n} \left(f(x) = C \cdot \prod\nolimits_{i=1}^{n} x_i - \sum\nolimits_{i=1}^{n} x_i \right).$$

2.3 Evolutionary Algorithms

In this paper, we consider the (1+1)-EA as described in Algorithm 1, which is a simple EA for maximizing pseudo-Boolean problems over $\{0,1\}^n$. The (1+1)-EA reflects the common structure of EAs. It maintains only one solution (i.e., the population size is 1), and repeatedly improves the current solution by using bit-wise mutation (i.e., step 3) and selection (i.e., steps 4 and 5). The (1+1)-EA has been widely used in the running time analysis of EAs, see (Neumann and Witt, 2010; Auger and Doerr, 2011).

Algorithm 1 ((1+1)-EA)**.** *Given a function f over $\{0,1\}^n$ to be maximized, it consists of the following steps:*

1. $x :=$ *uniformly randomly selected from* $\{0,1\}^n$.
2. *Repeat until the termination condition is met*
3. $\quad x' :=$ *flip each bit of x independently with probability* $1/n$.
4. $\quad$ *if* $f(x') \geq f(x)$
5. $\quad\quad x := x'$.

For the (1+1)-EA in noisy environments, only a noisy fitness value $f^n(x)$ is available, and thus step 4 of Algorithm 1 changes to be "if $f^n(x') \geq f^n(x)$". Note that we assume that the reevaluation strategy is used as in (Droste, 2004; Doerr et al., 2012a; Gießen and Kötzing, 2016), that is, when accessing the fitness of a solution, it is always calculated by sampling a new random variate, or drawing a new random single-bit mask. For example, for the (1+1)-EA, both $f^n(x')$ and $f^n(x)$ will be evaluated and reevaluated in each iteration. The running time in noisy optimization is usually defined as the number of fitness evaluations needed to find an optimal solution w.r.t. the true fitness function f for the first time (Droste, 2004; Akimoto et al., 2015; Gießen and Kötzing, 2016).

In noisy optimization, a worse solution may appear to have a "better" fitness and then survive to replace the true better solution which has a "worse" fitness. This may mislead the search direction of EAs, and then deteriorate the efficiency of EAs. To deal with this problem, a selection strategy for EAs handling noise was proposed (Markon et al., 2001; Bartz-Beielstein, 2005).

- **threshold selection:** an offspring solution will be accepted only if its fitness is larger than the parent solution by at least a predefined threshold $\tau \geq 0$.

For example, when using threshold selection, the 4th step of the (1+1)-EA in Algorithm 1 changes to be "if $f(x') \geq f(x) + \tau$" rather than "if $f(x') \geq f(x)$". Such a strategy can reduce the risk of accepting a bad solution due to noise. In (Qian et al., 2015a), it has been proved that threshold selection with $\tau = 1$ can make the (1+1)-EA solve the OneMax problem in polynomial time even if one-bit noise occurs with probability 1.

2.4 Sampling

In noisy evolutionary optimization, sampling as described in Definition 7 has often been used to reduce the negative effect of noise (Aizawa and Wah, 1994; Stagge, 1998; Branke and Schmidt, 2003, 2004). It approximates the true fitness $f(x)$ using the average of a number of random evaluations. Sampling can estimate the true fitness more accurately. For example, the output fitness $\hat{f}(x)$ by sampling under additive Gaussian noise $\mathcal{N}(\theta, \sigma^2)$ can be represented by $f(x) + \delta$ with $\delta \sim \mathcal{N}(\theta, \sigma^2/k)$, that is, sampling reduces the variance of noise by a factor of k. However, the computation time for the fitness estimation of a solution is also increased by k times.

Definition 7 (Sampling)**.** *Sampling first evaluates the fitness of a solution k times independently and obtains the noisy fitness values $f_1^n(x), \ldots, f_k^n(x)$, and then outputs their average as*

$$\hat{f}(x) = \frac{1}{k} \sum\nolimits_{i=1}^{k} f_i^n(x).$$

For the (1+1)-EA using sampling, the 4th step of Algorithm 1 changes to be "if $\hat{f}(x') \geq \hat{f}(x)$". Note that $k = 1$ is equivalent to that sampling is not used.

2.5 Analysis Tools

To derive running time bounds in this paper, we first model EAs as Markov chains, and then use a variety of drift theorems.

The evolution process usually goes forward only based on the current population, thus, an EA can be modeled as a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ (e.g., in (He and Yao, 2001; Yu and Zhou, 2008)) by taking the EA's population space $\mathcal{X}$ as the chain's state space, i.e. $\xi_t \in \mathcal{X}$. Note that the population space $\mathcal{X}$ consists of all possible populations. Let $\mathcal{X}^* \subset \mathcal{X}$ denote the set of all optimal populations, which contain at least one optimal solution. The goal of the EA is to reach $\mathcal{X}^*$ from an initial population. Thus, the process of an EA seeking $\mathcal{X}^*$ can be analyzed by studying the corresponding Markov chain with the optimal state space $\mathcal{X}^*$. Note that we consider the discrete state space (i.e., $\mathcal{X}$ is discrete) in this paper.

Given a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ and $\xi_{\hat{t}} = x$, we define its *first hitting time* (FHT) as a random variable τ such that $\tau = \min\{t \mid \xi_{\hat{t}+t} \in \mathcal{X}^*, t \geq 0\}$. That is, τ is the number of steps needed to reach the optimal space for the first time starting from $\xi_{\hat{t}} = x$. The mathematical expectation of τ, $\mathbb{E}[\![\tau \mid \xi_{\hat{t}} = x]\!] = \sum_{i=0}^{+\infty} iP(\tau = i)$, is called the *expected first hitting time* (EFHT) of this chain starting from $\xi_{\hat{t}} = x$. If ξ_0 is drawn from a distribution π_0, $\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!] = \sum_{x \in \mathcal{X}} \pi_0(x)\mathbb{E}[\![\tau \mid \xi_0 = x]\!]$ is called the EFHT of the Markov chain over the initial distribution π_0. Thus, the expected running time of the corresponding EA starting from $\xi_0 \sim \pi_0$ is equal to $N_1 + N_2 \cdot \mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!]$, where N_1 and N_2 are the number of fitness evaluations for the initial population and each iteration, respectively. For example, for the (1+1)-EA using sampling, $N_1 = k$ and $N_2 = 2k$ due to the reevaluation strategy. Note that when involving the expected running time of an EA on a problem in this paper, it is the expected running time starting from a uniform initial distribution π_u, i.e., $N_1 + N_2 \cdot \mathbb{E}[\![\tau \mid \xi_0 \sim \pi_u]\!] = N_1 + N_2 \cdot \sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \mathbb{E}[\![\tau \mid \xi_0 = x]\!]$.

Thus, in order to analyze the expected running time of EAs, we just need to analyze the EFHT of the corresponding Markov chains. In the following, we introduce the drift theorems which will be used to derive the EFHT of Markov chains in the paper.

Drift analysis was first introduced to the running time analysis of EAs by He and Yao (2001). Since then, it has become a popular tool in this field, and many variants have been proposed (e.g., in (Doerr et al., 2012b; Doerr and Goldberg, 2013)). In this paper, we will use its additive (i.e., Lemma 1) as well as multiplicative (i.e., Lemma 2) version. To use them, a function $V(x)$ has to be constructed to measure the distance of a state x to the optimal state space $\mathcal{X}^*$. The distance function $V(x)$ satisfies that $V(x \in \mathcal{X}^*) = 0$ and $V(x \notin \mathcal{X}^*) > 0$. Then, we need to investigate the progress on the distance to $\mathcal{X}^*$ in each step, i.e., $\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]\!]$. For additive drift analysis (i.e., Lemma 1), an upper bound of the EFHT can be derived through dividing the initial distance by a lower bound of the progress. Multiplicative drift analysis (i.e., Lemma 2) is much easier to use when the progress is roughly proportional to the current distance to the optimum.

Lemma 1 (Additive Drift Analysis (He and Yao, 2001))**.** *Given a Markov chain* $\{\xi_t\}_{t=0}^{+\infty}$ *and a distance function* $V(x)$*, if for any* $t \geq 0$ *and any* ξ_t *with* $V(\xi_t) > 0$*, there exists a real number* $c > 0$ *such that*

$$\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]\!] \geq c,$$

then the EFHT satisfies that $\mathbb{E}[\![\tau \mid \xi_0]\!] \leq V(\xi_0)/c$.

Lemma 2 (Multiplicative Drift Analysis (Doerr et al., 2012b))**.** *Given a Markov chain* $\{\xi_t\}_{t=0}^{+\infty}$ *and a distance function* $V(x)$*, if for any* $t \geq 0$ *and any* ξ_t *with* $V(\xi_t) > 0$*, there exists a real number* $c > 0$ *such that*

$$\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]\!] \geq c \cdot V(\xi_t),$$

then the EFHT satisfies that

$$\mathbb{E}[\![\tau \mid \xi_0]\!] \leq \frac{1 + \log(V(\xi_0)/V_{\min})}{c},$$

where $V_{\min} = \min\{V(x) \mid V(x) > 0\}$.

The simplified drift theorem (Oliveto and Witt, 2011, 2012) as presented in Lemma 3 was proposed to prove exponential lower bounds on the FHT of Markov chains, where X_t is usually represented by a mapping of ξ_t. It requires two conditions: a constant negative drift and exponentially decaying probabilities of jumping towards or away from the goal state. To relax the requirement of a constant negative drift, the simplified drift theorem with self-loops (Rowe and Sudholt, 2014) as presented in Lemma 4 has been proposed, which takes into account large self-loop probabilities.

Lemma 3 (Simplified Drift Theorem (Oliveto and Witt, 2011, 2012))**.** *Let* X_t*,* $t \geq 0$*, be real-valued random variables describing a stochastic process over some state space. Suppose there exists an interval* $[a, b] \subseteq \mathbb{R}$*, two constants* $\delta, \epsilon > 0$ *and, possibly depending on* $l := b - a$*, a function* $r(l)$ *satisfying* $1 \leq r(l) = o(l/\log(l))$ *such that for all* $t \geq 0$ *the following two conditions hold:*

1. $\mathbb{E}[\![X_t - X_{t+1} \mid a < X_t < b]\!] \leq -\epsilon,$

2. $P(|X_{t+1} - X_t| \geq j \mid X_t > a) \leq \dfrac{r(l)}{(1+\delta)^j}$ *for* $j \in \mathbb{N}_0$.

Then there is a constant $c > 0$ *such that for* $T := \min\{t \geq 0 : X_t \leq a \mid X_0 \geq b\}$ *it holds* $P(T \leq 2^{cl/r(l)}) = 2^{-\Omega(l/r(l))}$.

Lemma 4 (Simplified Drift Theorem with Self-loops (Rowe and Sudholt, 2014))**.** *Let* X_t, $t \geq 0$, *be real-valued random variables describing a stochastic process over some state space. Suppose there exists an interval* $[a, b] \subseteq \mathbb{R}$, *two constants* $\delta, \epsilon > 0$ *and, possibly depending on* $l := b - a$, *a function* $r(l)$ *satisfying* $1 \leq r(l) = o(l/\log(l))$ *such that for all* $t \geq 0$ *the following two conditions hold:*

1. $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \leq -\epsilon \cdot P(X_{t+1} \neq i \mid X_t = i)$ *for* $a < i < b$,
2. $P(|X_{t+1} - X_t| \geq j \mid X_t = i) \leq \dfrac{r(l)}{(1+\delta)^j} \cdot P(X_{t+1} \neq i \mid X_t = i)$ *for* $i > a, j \in \mathbb{N}_0$.

Then there is a constant $c > 0$ *such that for* $T := \min\{t \geq 0 : X_t \leq a \mid X_0 \geq b\}$ *it holds* $P(T \leq 2^{cl/r(l)}) = 2^{-\Omega(l/r(l))}$.

3 Robustness to Prior Noise

In this section, by comparing the expected running time of the (1+1)-EA with or without sampling for solving the OneMax and the LeadingOnes problems under one-bit noise, we show the robustness of sampling to prior noise.

3.1 The OneMax Problem

One-bit noise with $p = 1$ is considered here. We first analyze the case in which sampling is not used. Note that Droste (2004) proved that the expected running time is super-polynomial for $p \in \omega(\log(n)/n)$. Gießen and Kötzing (2016) have recently reproved the super-polynomial lower bound for $p \in \omega(\log(n)/n) \cap 1 - \omega(\log(n)/n)$ by using the simplified drift theorem (Oliveto and Witt, 2011, 2012). However, their proof does not cover $p = 1$. Here, we use the simplified drift theorem with self-loops (Rowe and Sudholt, 2014) to prove the lower bound of the exponential running time for $p = 1$ as shown in Theorem 1.

Theorem 1. *For the (1+1)-EA solving the OneMax problem under one-bit noise with* $p = 1$, *the expected running time is exponential.*

Proof. We use Lemma 4 to prove this theorem. Let X_t be the number of 0-bits of the solution after t iterations of the (1+1)-EA. We consider the interval $[0, n^{1/4}]$, i.e., the parameters $a = 0$ (i.e., the global optimum) and $b = n^{1/4}$ in Lemma 4.

Then, we analyze the drift $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!]$ for $1 \leq i < n^{1/4}$. Let $p_{i,i+d}$ denote the probability that the next solution after bit-wise mutation and selection has $i + d$ $(-i \leq d \leq n - i)$ number of 0-bits (i.e., $X_{t+1} = i + d$). We thus have

$$\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] = \sum_{d=1}^{i} d \cdot p_{i,i-d} - \sum_{d=1}^{n-i} d \cdot p_{i,i+d}. \quad (1)$$

We then analyze the probabilities $p_{i,i+d}$ for $i \geq 1$. Let P_d denote the probability that the offspring solution x' generated by bit-wise mutation has $i + d$ number of 0-bits. Note that one-bit noise with $p = 1$ makes the noisy fitness and the true fitness of a solution have a gap of one, i.e., $|f^n(x) - f(x)| = 1$. For a solution x with $|x|_0 = i$,

$f^n(x) = n-i+1$ with a probability of $\frac{i}{n}$; otherwise, $f^n(x) = n-i-1$. Let x and x' denote the current solution and the offspring solution, respectively.
(1) When $d \geq 3$, $f^n(x') \leq n-i-d+1 \leq n-i-2 < f^n(x)$. Thus, the offspring x' will be discarded in this case, which implies that $\forall d \geq 3: p_{i,i+d} = 0$.
(2) When $d = 2$, the offspring solution x' will be accepted if and only if $f^n(x') = n-i-1 = f^n(x)$, the probability of which is $\frac{i+2}{n} \cdot \frac{n-i}{n}$, since it needs to flip one 0-bit of x' and flip one 1-bit of x in noise. Thus, $p_{i,i+2} = P_2 \cdot (\frac{i+2}{n}\frac{n-i}{n})$.
(3) When $d = 1$, x' will be accepted if and only if $f^n(x') = n-i \wedge f^n(x) = n-i-1$, the probability of which is $\frac{i+1}{n} \cdot \frac{n-i}{n}$, since it needs to flip one 0-bit of x' and flip one 1-bit of x in noise. Thus, $p_{i,i+1} = P_1 \cdot (\frac{i+1}{n}\frac{n-i}{n})$.
(4) When $d = -1$, x' will be rejected if and only if $f^n(x') = n-i \wedge f^n(x) = n-i+1$, the probability of which is $\frac{n-i+1}{n} \cdot \frac{i}{n}$, since it needs to flip one 1-bit of x' and flip one 0-bit of x in noise. Thus, $p_{i,i-1} = P_{-1} \cdot (1 - \frac{n-i+1}{n}\frac{i}{n})$.
(5) When $d \leq -2$, $f^n(x') \geq n-i-d-1 \geq n-i+1 \geq f^n(x)$. Thus, the offspring x' will always be accepted in this case, which implies that $\forall d \leq -2: p_{i,i+d} = P_d$.

We then bound the probabilities P_d. For $d > 0$, $P_d \geq \binom{n-i}{d}\frac{1}{n^d}(1-\frac{1}{n})^{n-d}$, since it is sufficient to flip d 1-bits and keep other bits unchanged; $P_{-d} \leq \binom{i}{d}\frac{1}{n^d}$, since it is necessary to flip at least d 0-bits. Thus, we can upper bound $\sum_{d=2}^{i} dP_{-d}$ as follows:

$$\begin{aligned}\sum_{d=2}^{i} dP_{-d} &\leq \sum_{d=2}^{i} d\binom{i}{d}\frac{1}{n^d} = \sum_{d=1}^{i} d\binom{i}{d}\frac{1}{n^d} - \frac{i}{n} \\ &= \frac{i}{n}\sum_{d=0}^{i-1}\binom{i-1}{d}\frac{1}{n^d} - \frac{i}{n} = \frac{i}{n}\left(\left(1+\frac{1}{n}\right)^{i-1} - 1\right).\end{aligned}$$

For P_{-1}, we also need a tighter upper bound (see Lemma 2 in (Paixão et al., 2015))

$$P_{-1} \leq \frac{i}{n}\left(1-\frac{1}{n}\right)^{n-1} \cdot 1.14.$$

By applying these probabilities to Eq. (1), we have

$$\begin{aligned}&\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \\ &= \left(1 - \frac{n-i+1}{n}\frac{i}{n}\right)P_{-1} + \sum_{d=2}^{i} dP_{-d} - \frac{i+1}{n}\frac{n-i}{n}P_1 - 2\frac{i+2}{n}\frac{n-i}{n}P_2 \\ &\leq \left(1 - \frac{n-i+1}{n}\frac{i}{n}\right)\frac{i}{n}\left(1-\frac{1}{n}\right)^{n-1} \cdot 1.14 + \frac{i}{n}\left(\left(1+\frac{1}{n}\right)^{i-1} - 1\right) \\ &\quad - \frac{i+1}{n}\frac{n-i}{n}\frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1} - 2\frac{i+2}{n}\frac{n-i}{n}\frac{(n-i)(n-i-1)}{2n^2}\left(1-\frac{1}{n}\right)^{n-2} \\ &\leq \frac{i}{n}\left(1-\frac{1}{n}\right)^{n-1}\left(1.14 - 1 - 2\cdot\frac{1}{2}\right) + O\left(\left(\frac{i}{n}\right)^2\right) \quad (\text{since } i < n^{1/4}) \\ &\leq -0.3 \cdot \frac{i}{n} + O\left(\left(\frac{i}{n}\right)^2\right). \quad (\text{by } \left(1-\frac{1}{n}\right)^{n-1} \geq \frac{1}{e})\end{aligned}$$

To investigate the condition of Lemma 4, we also need to analyze the probability

$P(X_{t+1} \neq i \mid X_t = i)$ for $1 \leq i < n^{1/4}$. We have

$$P(X_{t+1} \neq i \mid X_t = i) = \left(1 - \frac{n-i+1}{n}\frac{i}{n}\right) P_{-1} + \sum_{d=2}^{i} P_{-d} + \frac{i+1}{n}\frac{n-i}{n} P_1 + \frac{i+2}{n}\frac{n-i}{n} P_2.$$

It is easy to verify that $P(X_{t+1} \neq i \mid X_t = i) = \Theta(\frac{i}{n})$. Thus, $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] = -\Omega(P(X_{t+1} \neq i \mid X_t = i))$, which implies that condition 1 of Lemma 4 holds.

For condition 2 of Lemma 4, we need to compare $P(|X_{t+1} - X_t| \geq j \mid X_t = i)$ with $\frac{r(l)}{(1+\delta)^j} \cdot P(X_{t+1} \neq i \mid X_t = i)$ for $i \geq 1$. We rewrite $P(X_{t+1} \neq i \mid X_t = i)$ as $P(|X_{t+1} - X_t| \geq 1 \mid X_t = i)$, and show that condition 2 holds with $\delta = 1$ and $r(l) = \frac{32e}{7}$. For $j \in \{1, 2, 3\}$, it trivially holds, because $\frac{r(l)}{(1+\delta)^j} > 1$. For $j \geq 4$, according to the analysis on $p_{i,i+d}$, we have

$$P(|X_{t+1} - X_t| \geq j \mid X_t = i) = \sum_{d=j}^{i} P_{-d} \leq \binom{i}{j}\frac{1}{n^j} \leq \frac{1}{j!}\left(\frac{i}{n}\right)^j \leq \frac{2}{2^j} \cdot \frac{i}{n},$$

where the first inequality is because for decreasing the number of 0-bits by at least j in mutation, it is necessary to flip at least j 0-bits. Furthermore, we have

$$\begin{aligned} P(|X_{t+1} - X_t| \geq 1 \mid X_t = i) \geq p_{i,i-1} &= \left(1 - \frac{n-i+1}{n}\frac{i}{n}\right) \cdot P_{-1} \\ &\geq \left(1 - \frac{n-i+1}{n}\frac{i}{n}\right) \cdot \frac{i}{n}\left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{7}{16e} \cdot \frac{i}{n}, \end{aligned}$$

where the last inequality holds with $n \geq 2$. Thus,

$$\begin{aligned} \frac{r(l)}{(1+\delta)^j} \cdot P(|X_{t+1} - X_t| \geq 1 \mid X_t = i) &\geq \frac{32e}{7}\frac{1}{2^j} \cdot \frac{7}{16e}\frac{i}{n} \\ &= \frac{2}{2^j}\frac{i}{n} \geq P(|X_{t+1} - X_t| \geq j \mid X_t = i), \end{aligned}$$

which implies that condition 2 of Lemma 4 holds.

Note that $l = b - a = n^{1/4}$. Thus, by Lemma 4, the probability that the running time is $2^{O(n^{1/4})}$ when starting from a solution x with $|x|_0 \geq n^{1/4}$ is exponentially small. Due to the uniform initial distribution, the probability that the initial solution x has $|x|_0 < n^{1/4}$ is exponentially small by Chernoff's inequality. Thus, the expected running time is exponential. □

Then, we analyze the case in which sampling with $k = 2$ is used. The expected running time is still exponential, as shown in Theorem 2. The proof is very similar to that of Theorem 1. The change of the probabilities $p_{i,i+d}$ led by increasing k from 1 to 2 does not affect the application of the simplified drift theorem with self-loops (i.e., Lemma 4). The detailed proofs are shown in the supplementary material due to space limitations.

Theorem 2. *For the (1+1)-EA solving the OneMax problem under one-bit noise with $p = 1$, if using sampling with $k = 2$, the expected running time is exponential.*

We have shown that sampling with $k = 2$ is not effective. In the following, we prove that increasing k from 2 to 3 can reduce the expected running time to be polynomial as shown in Theorem 3, the proof of which is accomplished by applying multiplicative drift analysis (Doerr et al., 2012b).

Theorem 3. *For the (1+1)-EA solving the OneMax problem with $n \geq 18$ under one-bit noise with $p = 1$, if using sampling with $k = 3$, the expected running time is $O(n \log n)$.*

Proof. We use Lemma 2 to prove this theorem. We first construct a distance function $V(x)$ as $\forall x \in \mathcal{X} = \{0,1\}^n, V(x) = |x|_0$, where $|x|_0 = n - \sum_{i=1}^{n} x_i$ is the number of 0-bits of the solution x. It is easy to verify that $V(x \in \mathcal{X}^* = \{1^n\}) = 0$ and $V(x \notin \mathcal{X}^*) > 0$.

Then, we investigate $\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!]$ for any x with $V(x) > 0$ (i.e., $x \notin \mathcal{X}^*$). We denote the number of 0-bits of the current solution x by i (where $1 \leq i \leq n$). Let $p_{i,i+d}$ be the probability that the next solution after bit-wise mutation and selection has $i + d$ number of 0-bits (where $-i \leq d \leq n - i$). Note that we are referring to the true number of 0-bits of a solution instead of the effective number of 0-bits after noisy evaluation. Thus,

$$\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] = \sum_{d=1}^{i} d \cdot p_{i,i-d} - \sum_{d=1}^{n-i} d \cdot p_{i,i+d}. \tag{2}$$

We then analyze $p_{i,i+d}$ for $1 \leq i \leq n$ as in the proof of Theorem 1. Note that for a solution x, the fitness value output by sampling with $k = 3$ is the average of noisy fitness values output by three independent fitness evaluations, i.e., $\hat{f}(x) = (f_1^n(x) + f_2^n(x) + f_3^n(x))/3$.

(1) When $d \geq 3$, $\hat{f}(x') \leq n - i - d + 1 \leq n - i - 2 < \hat{f}(x)$. Thus, the offspring x' will be discarded, then we have $\forall d \geq 3 : p_{i,i+d} = 0$.

(2) When $d = 2$, x' will be accepted if and only if $\hat{f}(x') = n - i - 1 = \hat{f}(x)$, the probability of which is $(\frac{i+2}{n})^3 \cdot (\frac{n-i}{n})^3$, since it needs to always flip one 0-bit of x' and flip one 1-bit of x in three noisy fitness evaluations. Thus, $p_{i,i+2} = P_2 \cdot (\frac{i+2}{n})^3 (\frac{n-i}{n})^3$.

(3) When $d = 1$, there are three possible cases for the acceptance of x': $\hat{f}(x') = n - i \wedge \hat{f}(x) = n - i - 1$, $\hat{f}(x') = n - i \wedge \hat{f}(x) = n - i - \frac{1}{3}$ and $\hat{f}(x') = n - i - \frac{2}{3} \wedge \hat{f}(x) = n - i - 1$. The probability of $\hat{f}(x') = n - i$ is $(\frac{i+1}{n})^3$, since it needs to always flip one 0-bit of x in three noisy evaluations. The probability of $\hat{f}(x') = n - i - \frac{2}{3}$ is $3(\frac{i+1}{n})^2 \frac{n-i-1}{n}$, since it needs to flip one 0-bit of x in two noisy evaluations and flip one 1-bit in the other noisy evaluation. Similarly, we can derive that the probabilities of $\hat{f}(x) = n - i - 1$ and $\hat{f}(x) = n - i - \frac{1}{3}$ are $(\frac{n-i}{n})^3$ and $3(\frac{n-i}{n})^2 \frac{i}{n}$, respectively. Thus, $p_{i,i+1} = P_1 \cdot ((\frac{i+1}{n})^3 ((\frac{n-i}{n})^3 + 3(\frac{n-i}{n})^2 \frac{i}{n}) + 3(\frac{i+1}{n})^2 \frac{n-i-1}{n} (\frac{n-i}{n})^3)$.

(4) When $d = -1$, there are three possible cases for the rejection of x': $\hat{f}(x') = n - i \wedge \hat{f}(x) = n - i + 1$, $\hat{f}(x') = n - i \wedge \hat{f}(x) = n - i + \frac{1}{3}$ and $\hat{f}(x') = n - i + \frac{2}{3} \wedge \hat{f}(x) = n - i + 1$. The probability of $\hat{f}(x') = n - i$ is $(\frac{n-i+1}{n})^3$, since it needs to always flip one 1-bit of x' in three noisy evaluations. The probability of $\hat{f}(x') = n - i + \frac{2}{3}$ is $3(\frac{n-i+1}{n})^2 \frac{i-1}{n}$, since it needs to flip one 1-bit of x' in two noisy evaluations and flip one 0-bit in the other evaluation. Similarly, we can derive that the probabilities of $\hat{f}(x) = n - i + 1$ and $\hat{f}(x) = n - i + \frac{1}{3}$ are $(\frac{i}{n})^3$ and $3(\frac{i}{n})^2 \frac{n-i}{n}$, respectively. Thus, $p_{i,i-1} = P_{-1} \cdot (1 - (\frac{n-i+1}{n})^3 ((\frac{i}{n})^3 + 3(\frac{i}{n})^2 \frac{n-i}{n}) - 3(\frac{n-i+1}{n})^2 \frac{i-1}{n} (\frac{i}{n})^3)$.

(5) When $d \leq -2$, $\hat{f}(x') \geq n - i - d - 1 \geq n - i + 1 \geq \hat{f}(x)$. Thus, x' will always be accepted, then we have $\forall d \leq -2 : p_{i,i+d} = P_d$.

By applying these probabilities to Eq. (2), we have

$$\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] \geq p_{i,i-1} - p_{i,i+1} - 2 \cdot p_{i,i+2} \tag{3}$$
$$= P_{-1} \cdot \left(1 - \left(\frac{n-i+1}{n}\right)^3 \left(\left(\frac{i}{n}\right)^3 + 3\left(\frac{i}{n}\right)^2 \frac{n-i}{n}\right) - 3\left(\frac{n-i+1}{n}\right)^2 \frac{i-1}{n} \left(\frac{i}{n}\right)^3\right)$$

$$-P_1\cdot\left(\left(\frac{i+1}{n}\right)^3\left(\left(\frac{n-i}{n}\right)^3+3\left(\frac{n-i}{n}\right)^2\frac{i}{n}\right)+3\left(\frac{i+1}{n}\right)^2\frac{n-i-1}{n}\left(\frac{n-i}{n}\right)^3\right)$$
$$-2\cdot P_2\cdot\left(\frac{i+2}{n}\right)^3\left(\frac{n-i}{n}\right)^3.$$

We simplify the above equation by using simple mathematical calculations.

$$\begin{aligned}&\left(\frac{i+1}{n}\right)^3\left(\left(\frac{n-i}{n}\right)^3+3\left(\frac{n-i}{n}\right)^2\frac{i}{n}\right)+3\left(\frac{i+1}{n}\right)^2\frac{n-i-1}{n}\left(\frac{n-i}{n}\right)^3\\&=\frac{i+1}{n}\frac{n-i}{n}\cdot\left(-5\left(\frac{i+1}{n}\frac{n-i}{n}\right)^2+3\left(1+\frac{1}{n}\right)\cdot\frac{i+1}{n}\frac{n-i}{n}\right)\\&\le\frac{9}{20}\frac{i+1}{n}\frac{n-i}{n}\left(\frac{n+1}{n}\right)^2,\end{aligned}$$

where the inequality is because $-5x^2+3(1+\frac{1}{n})x\le\frac{9}{20}(\frac{n+1}{n})^2$.
By replacing i with $n-i$ in the above equation, we get

$$\begin{aligned}&\left(\frac{n-i+1}{n}\right)^3\left(\left(\frac{i}{n}\right)^3+3\left(\frac{i}{n}\right)^2\frac{n-i}{n}\right)+3\left(\frac{n-i+1}{n}\right)^2\frac{i-1}{n}\left(\frac{i}{n}\right)^3\\&\le\frac{9}{20}\frac{i}{n}\frac{n-i+1}{n}\left(\frac{n+1}{n}\right)^2.\end{aligned}$$

Thus, Eq. (3) becomes

$$\begin{aligned}&\mathbb{E}[\![V(\xi_t)-V(\xi_{t+1})\mid\xi_t=x]\!]\\&\ge P_{-1}\cdot\left(1-\frac{9}{20}\frac{i}{n}\frac{n-i+1}{n}\left(\frac{n+1}{n}\right)^2\right)-P_1\cdot\frac{9}{20}\frac{i+1}{n}\frac{n-i}{n}\left(\frac{n+1}{n}\right)^2\\&\quad-2\cdot P_2\cdot\left(\frac{i+2}{n}\right)^3\left(\frac{n-i}{n}\right)^3.\end{aligned}\tag{4}$$

We then bound the three mutation probabilities P_{-1}, P_1 and P_2. For decreasing the number of 0-bits by 1 in mutation, it is sufficient to flip one 0-bit and keep other bits unchanged, thus we have $P_{-1}\ge\frac{i}{n}(1-\frac{1}{n})^{n-1}$. For increasing the number of 0-bits by 2, it is necessary to flip at least two 1-bits, thus we have $P_2\le\binom{n-i}{2}\frac{1}{n^2}=\frac{(n-i)(n-i-1)}{2n^2}$. For increasing the number of 0-bits by 1, it needs to flip one more 1-bit than the number of 0-bits it flips, thus we have

$$\begin{aligned}P_1&=\sum_{k=1}^{\min\{n-i,i+1\}}\binom{n-i}{k}\binom{i}{k-1}\frac{1}{n^{2k-1}}\left(1-\frac{1}{n}\right)^{n-2k+1}\\&\le\frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1}+\sum_{k=2}^{\min\{n-i,i+1\}}\frac{1}{k!(k-1)!}\frac{(n-i)^k}{n^k}\frac{i^{k-1}}{n^{k-1}}\left(1-\frac{1}{n}\right)^{n-2k+1}\\&\le\frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1}+\frac{i}{n}\cdot\sum_{k=2}^{\min\{n-i,i+1\}}\frac{1}{k!(k-1)!}\left(1-\frac{1}{n}\right)^{n-1}\end{aligned}$$

$$\le \frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1} + \frac{i}{n}\cdot\sum_{k=2}^{+\infty}\frac{1}{k!}\left(1-\frac{1}{n}\right)^{n-1}$$
$$= \frac{n-i}{n}\left(1-\frac{1}{n}\right)^{n-1} + (e-2)\frac{i}{n}\left(1-\frac{1}{n}\right)^{n-1}.$$

By applying these probability bounds to Eq. (4), we have

$$\mathbb{E}[\![V(\xi_t)-V(\xi_{t+1}) \mid \xi_t = x]\!]$$
$$\ge \frac{i}{n}\left(1-\frac{1}{n}\right)^{n-1}\left(1-\frac{9}{20}\left(\frac{n+1}{n}\right)^2\left(\frac{i}{n}\frac{n-i+1}{n}+\frac{i+1}{i}\frac{n-i}{n}\left(\frac{n-i}{n}+(e-2)\frac{i}{n}\right)\right)\right)$$
$$-\frac{i}{n}\cdot 2\cdot\frac{(n-i)(n-i-1)}{2n^2}\cdot\frac{i+2}{i}\left(\frac{i+2}{n}\right)^2\left(\frac{n-i}{n}\right)^3.$$

When $i \ge 2$, $1+1/i \le 3/2$ and $1+2/i \le 2$, thus we get

$$\mathbb{E}[\![V(\xi_t)-V(\xi_{t+1}) \mid \xi_t = x]\!] \ge \frac{i}{en}\left(1-\frac{9}{20}\left(\frac{n+1}{n}\right)^2\cdot\frac{3}{2}\right)-\frac{32}{729}\frac{i}{n}\left(\frac{n+2}{n}\right)^5$$
$$=\frac{i}{n}\cdot\left(\frac{1}{e}-\frac{27}{40e}\left(\frac{n+1}{n}\right)^2-\frac{32}{729}\left(\frac{n+2}{n}\right)^5\right)\ge 0.003\cdot\frac{i}{n},$$

where the first inequality is by using $\frac{(n-i)(n-i-1)}{n^2}(\frac{i+2}{n})^2(\frac{n-i}{n})^3 \le \frac{n-2}{n}(\frac{i+2}{n}(\frac{n-i}{n})^2)^2 \le \frac{n-2}{n}(\frac{4}{27}(\frac{n+2}{n})^3)^2 \le \frac{16}{729}(\frac{n+2}{n})^5$, and the last inequality holds with $n \ge 15$.
When $i = 1$, using Eq. (3), we get

$$\mathbb{E}[\![V(\xi_t)-V(\xi_{t+1}) \mid \xi_t = x]\!]$$
$$= P_{-1}\cdot\left(1-\left(\left(\frac{1}{n}\right)^3+3\left(\frac{1}{n}\right)^2\frac{n-1}{n}\right)\right)-2\cdot P_2\cdot\left(\frac{3}{n}\right)^3\left(\frac{n-1}{n}\right)^3$$
$$-P_1\cdot\left(\left(\frac{2}{n}\right)^3\left(\left(\frac{n-1}{n}\right)^3+3\left(\frac{n-1}{n}\right)^2\frac{1}{n}\right)+3\left(\frac{2}{n}\right)^2\frac{n-2}{n}\left(\frac{n-1}{n}\right)^3\right)$$
$$\ge\frac{1}{en}\left(1-\frac{3}{n^2}-\frac{16}{n^2}-\frac{12}{n}\right)-\frac{27}{n^3}\ge 0.01\cdot\frac{1}{n},$$

where the last inequality holds with $n \ge 18$.
Thus, the condition of Lemma 2 holds with $\mathbb{E}[\![V(\xi_t)-V(\xi_{t+1}) \mid \xi_t = x]\!] \ge \frac{0.003}{n}\cdot V(\xi_t)$.
We then get, noting that $V_{\min} = 1$ and $V(x) \le n$,

$$\mathbb{E}[\![\tau \mid \xi_0]\!] \le \frac{n}{0.003}(1+\log V(\xi_0)) \in O(n\log n),$$

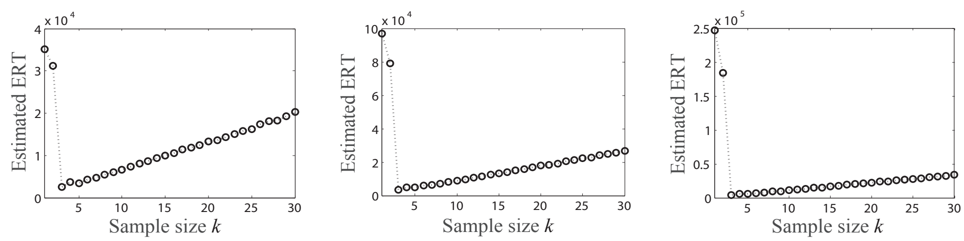
i.e., the expected running time is upper bounded by $O(n\log n)$. □

Thus, we have shown that sampling is robust to noise for the (1+1)-EA solving the OneMax problem in the presence of one-bit noise. By comparing Theorem 2 with Theorem 3, we also find that a gap of one on the value of k can lead to an exponential difference on the expected running time, which reveals that a careful selection of k is important for the effectiveness of sampling. The complexity transition from $k = 2$ to

$k = 3$ is because sampling with $k = 3$ can make false progress (i.e., accepting solutions with more 0-bits) dominated by true progress (i.e., accepting solutions with fewer 0-bits), while sampling with $k = 2$ is not sufficient.

We have also conducted experiments to complement the theoretical results, which give bounds only. For each value of n and k, we run the (1+1)-EA 1000 times independently. In each run, we record the number of fitness evaluations until an optimal solution w.r.t. the true fitness function is found for the first time. Then the total number of evaluations of the 1000 runs are averaged as the estimation of the expected running time, called as the estimated ERT. We will always compute the estimated ERT in this way for the experiments throughout this paper.

We estimate the expected running time of the (1+1)-EA using sampling with k from 1 to 30. The results for $n = 40, 50, 60$ are plotted in Figure 1. We can observe that the curves are high at $k = 1, 2$ and drop suddenly at $k = 3$, which is consistent with our theoretical results in Theorems 1-3. Note that the curves grow linearly since $k = 3$, which is because ERT $= 2k\cdot$ EFHT (i.e., the number of fitness evaluations in each iteration $\times$ the number of iterations), and when the noise has been sufficiently reduced by sampling, the number of iterations cannot further reduce as k increases, but the sampling cost increases linearly with k.



(a) $n = 40$ (b) $n = 50$ (c) $n = 60$

Figure 1: Estimated ERT for the (1+1)-EA using sampling on the OneMax problem under one-bit noise with $p = 1$.

3.2 The LeadingOnes Problem

One-bit noise with $p = \frac{1}{2}$ is considered here. For the case in which sampling is not used, Gießen and Kötzing (2016) have proved the exponential running time lower bound as shown in Theorem 4. We prove in Theorem 5 that sampling can reduce the expected running time to be polynomial.

Theorem 4. *(Gießen and Kötzing, 2016) For the (1+1)-EA solving the LeadingOnes problem under one-bit noise with* $p = \frac{1}{2}$*, the expected running time is* $2^{\Omega(n)}$.

Theorem 5. *For the (1+1)-EA solving the LeadingOnes problem under one-bit noise with* $p = \frac{1}{2}$*, if using sampling with* $k = 10n^4$*, the expected running time is* $O(n^6)$.

Proof. We use Lemma 1 to prove this theorem. Let $LO(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$ denote the number of leading 1-bits of a solution x. We first construct a distance function $V(x)$ as $\forall x \in \mathcal{X} = \{0,1\}^n, V(x) = n - LO(x)$. It is easy to verify that $V(x \in \mathcal{X}^* = \{1^n\}) = 0$ and $V(x \notin \mathcal{X}^*) > 0$.

Then, we analyze $\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!]$ for any x with $V(x) > 0$. For the current solution x, assume that $LO(x) = i$ (where $0 \le i \le n-1$). Let x' be the offspring solution produced by mutating x. We consider three mutation cases for $LO(x')$:

(1) The l-th leading 1-bit is flipped and the first $(l-1)$ leading 1-bits remain unchanged, which leads to $LO(x') = l-1$. Thus, $\forall 1 \leq l \leq i : P(LO(x') = l-1) = (1-\frac{1}{n})^{l-1}\frac{1}{n}$.
(2) The $(i+1)$-th bit (which must be 0) is flipped and the first i leading 1-bits remain unchanged, which leads to $LO(x') \geq i+1$. Thus, we have $P(LO(x') \geq i+1) = (1-\frac{1}{n})^{i}\frac{1}{n}$.
(3) The first $(i+1)$ bits remain unchanged, which leads to $LO(x') = i$. Thus, $P(LO(x') = i) = (1-\frac{1}{n})^{i+1}$.

Assume that $LO(x') = j$. We then analyze the acceptance probability of x', i.e., $P(\hat{f}(x') \geq \hat{f}(x))$. Note that $\hat{f}(x) = (\sum_{i=1}^{k} f_i^n(x))/k$, where $f_i^n(x)$ is the fitness output by one independent noisy evaluation. By one-bit noise with $p = \frac{1}{2}$, the $f^n(x)$ value can be calculated as follows:
(1) The noise does not occur, whose probability is $1-p = \frac{1}{2}$. Thus, $P(f^n(x) = i) = \frac{1}{2}$.
(2) The noise occurs, the probability of which is $p = \frac{1}{2}$.
(2.1) It flips the l-th leading 1-bit, then $f^n(x) = l-1$. Thus, we have $\forall 1 \leq l \leq i : P(f^n(x) = l-1) = \frac{1}{2n}$.
(2.2) It flips the $(i+1)$-th bit, which leads to $f^n(x) \geq i+1$. Thus, we have $P(f^n(x) \geq i+1) = \frac{1}{2n}$. Note that $f^n(x)$ reaches the minimum $i+1$ when x has 0-bit at position $i+2$, and $f^n(x)$ reaches the maximum n when x has all 1-bits since position $i+2$.
(2.3) Otherwise, $f^n(x)$ remains unchanged. Thus, we have $P(f^n(x) = i) = \frac{1}{2}(1-\frac{i+1}{n})$.
For each i, let x_i^{opt} be the solution which has only 1-bits except for the $(i+1)$-th bit (i.e., $x_i^{\text{opt}} = 1^i 0 1^{n-i-1}$), and let x_i^{pes} be the solution with i leading 1-bits and otherwise only 0-bits (i.e., $x_i^{\text{pes}} = 1^i 0^{n-i}$). Then we have the stochastic ordering $f^n(x_i^{\text{pes}}) \preceq f^n(x) \preceq f^n(x_i^{\text{opt}})$, which implies that $\hat{f}(x_i^{\text{pes}}) \preceq \hat{f}(x) \preceq \hat{f}(x_i^{\text{opt}})$. We can similarly get $\hat{f}(x_j^{\text{pes}}) \preceq \hat{f}(x') \preceq \hat{f}(x_j^{\text{opt}})$. Thus, it is easy to see that

$$P(\hat{f}(x_j^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}})) \leq P(\hat{f}(x') \geq \hat{f}(x)) \leq P(\hat{f}(x_j^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}})). \tag{5}$$

Let $P_{mut}(x, x')$ be the probability that x' is generated by mutating x. By combining the mutation probability with the acceptance probability, we have

$$\begin{aligned}
&\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] \qquad (6)\\
&= \sum_{j=0}^{n} \sum_{LO(x')=j} P_{mut}(x,x') \cdot P(\hat{f}(x') \geq \hat{f}(x)) \cdot (n-i-(n-j))\\
&\geq \sum_{j=0}^{i-1} \sum_{LO(x')=j} P_{mut}(x,x') \cdot P(\hat{f}(x_j^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}})) \cdot (j-i)\\
&\quad + \sum_{j=i+1}^{n} \sum_{LO(x')=j} P_{mut}(x,x') \cdot P(\hat{f}(x_j^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}})) \cdot (j-i) \qquad \text{(by Eq. (5))}\\
&\geq \sum_{j=0}^{i-1} \sum_{LO(x')=j} P_{mut}(x,x') \cdot P(\hat{f}(x_{i-1}^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}})) \cdot (j-i)\\
&\quad + \sum_{j=i+1}^{n} \sum_{LO(x')=j} P_{mut}(x,x') \cdot P(\hat{f}(x_{i+1}^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}})) \cdot (i+1-i)\\
&\quad \text{(Because } P(\hat{f}(x_j^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}})) \leq P(\hat{f}(x_{i-1}^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}})) \text{ for } j \leq i-1,\\
&\qquad \text{and } P(\hat{f}(x_j^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}})) \geq P(\hat{f}(x_{i+1}^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}})) \text{ for } j \geq i+1)
\end{aligned}$$

$$
\begin{aligned}
&= \left(1-\frac{1}{n}\right)^i \frac{1}{n} P(\hat{f}(x_{i+1}^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}})) - \left(\sum_{j=0}^{i-1} \left(1-\frac{1}{n}\right)^j \frac{1}{n} \cdot (i-j)\right) P(\hat{f}(x_{i-1}^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}})) \\
&\quad (\text{Because } \sum_{j=i+1}^{n} \sum_{LO(x')=j} P_{mut}(x,x') = P(LO(x') \geq i+1) = \left(1-\frac{1}{n}\right)^i \frac{1}{n} \\
&\qquad \text{and } \forall 0 \leq j \leq i-1: \sum_{LO(x')=j} P_{mut}(x,x') = P(LO(x')=j) = \left(1-\frac{1}{n}\right)^j \frac{1}{n}) \\
&\geq \frac{1}{en} P(\hat{f}(x_{i+1}^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}})) - \frac{i(i+1)}{2n} P(\hat{f}(x_{i-1}^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}})). \\
&\quad (\text{Because } i \leq n-1 \text{ and } 1 \geq \left(1-\frac{1}{n}\right)^i \geq \left(1-\frac{1}{n}\right)^{n-1} \geq \frac{1}{e})
\end{aligned}
$$

We then bound the probabilities $P(\hat{f}(x_{i+1}^{\text{pes}}) \geq \hat{f}(x_i^{\text{opt}}))$ and $P(\hat{f}(x_{i-1}^{\text{opt}}) \geq \hat{f}(x_i^{\text{pes}}))$. First, we have

$$
\begin{aligned}
P(\hat{f}(x') \geq \hat{f}(x)) &= P\left(\left(\sum_{i=1}^{k} f_i^n(x')\right)/k \geq \left(\sum_{i=1}^{k} f_i^n(x)\right)/k\right) \\
&= P\left(\sum_{i=1}^{k} f_i^n(x') - \sum_{i=1}^{k} f_i^n(x) \geq 0\right) = P(Z(x',x) \geq 0),
\end{aligned}
$$

where the random variable $Z(x',x)$ is used to represent $\sum_{i=1}^{k} f_i^n(x') - \sum_{i=1}^{k} f_i^n(x)$ for convenience. We then calculate the expectation and variance of $f^n(x_i^{\text{opt}})$ and $f^n(x_i^{\text{pes}})$. Based on the analysis of $f^n(x)$, we can easily derive

$$
\begin{aligned}
&\mathbb{E}[\![f^n(x_i^{\text{opt}})]\!] = \frac{1}{2}i + \sum_{j=0}^{i-1} \frac{1}{2n} j + \frac{1}{2n} n + \frac{1}{2}\left(1-\frac{i+1}{n}\right) i = i + \frac{1}{2} - \frac{i^2+3i}{4n}, \\
&\mathbb{E}[\![f^n(x_i^{\text{pes}})]\!] = \frac{1}{2}i + \sum_{j=0}^{i-1} \frac{1}{2n} j + \frac{1}{2n}(i+1) + \frac{1}{2}\left(1-\frac{i+1}{n}\right) i = i - \frac{i^2+i-2}{4n}, \\
&\text{Var}(f^n(x_i^{\text{opt}})) = \mathbb{E}[\![(f^n(x_i^{\text{opt}}))^2]\!] - (\mathbb{E}[\![f^n(x_i^{\text{opt}})]\!])^2 \\
&= \frac{1}{2}i^2 + \sum_{j=0}^{i-1} \frac{1}{2n} j^2 + \frac{1}{2n} n^2 + \frac{1}{2}\left(1-\frac{i+1}{n}\right) i^2 - \left(i + \frac{1}{2} - \frac{i^2+3i}{4n}\right)^2 \\
&\leq \frac{1}{6}n^2 + \frac{3}{2}n + \frac{5}{6} \leq \frac{7}{6}n^2, \\
&\text{Var}(f^n(x_i^{\text{pes}})) = \mathbb{E}[\![(f^n(x_i^{\text{pes}}))^2]\!] - (\mathbb{E}[\![f^n(x_i^{\text{pes}})]\!])^2 \\
&= \frac{1}{2}i^2 + \sum_{j=0}^{i-1} \frac{1}{2n} j^2 + \frac{1}{2n}(i+1)^2 + \frac{1}{2}\left(1-\frac{i+1}{n}\right) i^2 - \left(i - \frac{i^2+i-2}{4n}\right)^2 \\
&\leq \frac{1}{6}n^2 + \frac{1}{4}n + \frac{1}{12} + \frac{1}{2n} \leq \frac{5}{12}n^2.
\end{aligned}
$$

Note that the last inequalities for $\text{Var}(f^n(x_i^{\text{opt}}))$ and $\text{Var}(f^n(x_i^{\text{pes}}))$ hold with $n \geq 2$. Thus, we have

$$
\mathbb{E}[\![Z(x_{i+1}^{\text{pes}}, x_i^{\text{opt}})]\!] = k(\mathbb{E}[\![f^n(x_{i+1}^{\text{pes}})]\!] - \mathbb{E}[\![f^n(x_i^{\text{opt}})]\!]) = \frac{k}{2},
$$

$$\mathbb{E}[\![Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}})]\!] = k(\mathbb{E}[\![f^n(x_{i-1}^{\mathrm{opt}})]\!] - \mathbb{E}[\![f^n(x_i^{\mathrm{pes}})]\!]) = -\frac{k}{2},$$
$$\mathrm{Var}(Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}})) = k(\mathrm{Var}(f^n(x_{i+1}^{\mathrm{pes}})) + \mathrm{Var}(f^n(x_i^{\mathrm{opt}}))) \leq \frac{19}{12}kn^2,$$
$$\mathrm{Var}(Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}})) = k(\mathrm{Var}(f^n(x_{i-1}^{\mathrm{opt}})) + \mathrm{Var}(f^n(x_i^{\mathrm{pes}}))) \leq \frac{19}{12}kn^2.$$

Then, we can get the bounds on the probabilities $P(\hat{f}(x_{i+1}^{\mathrm{pes}}) \geq \hat{f}(x_i^{\mathrm{opt}}))$ and $P(\hat{f}(x_{i-1}^{\mathrm{opt}}) \geq \hat{f}(x_i^{\mathrm{pes}}))$ by Chebyshev's inequality. Note that $Z(x', x)$ is integer-valued.

$$\begin{aligned} &P(\hat{f}(x_{i+1}^{\mathrm{pes}}) \geq \hat{f}(x_i^{\mathrm{opt}})) = P(Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}}) \geq 0) = 1 - P(Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}}) \leq -1) \\ &= 1 - P(Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}}) - \mathbb{E}[\![Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}})]\!] \leq -1 - \mathbb{E}[\![Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}})]\!]) \\ &\geq 1 - P(|Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}}) - \mathbb{E}[\![Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}})]\!]| \geq 1 + k/2) \\ &\geq 1 - \frac{\mathrm{Var}(Z(x_{i+1}^{\mathrm{pes}}, x_i^{\mathrm{opt}}))}{(1+k/2)^2} \quad \text{(by Chebyshev's inequality)} \quad \geq 1 - \frac{19n^2}{3k}. \end{aligned}$$

Similarly, we have

$$\begin{aligned} &P(\hat{f}(x_{i-1}^{\mathrm{opt}}) \geq \hat{f}(x_i^{\mathrm{pes}})) = P(Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}}) \geq 0) \\ &= P(Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}}) - \mathbb{E}[\![Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}})]\!] \geq -\mathbb{E}[\![Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}})]\!]) \\ &\leq P(|Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}}) - \mathbb{E}[\![Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}})]\!]| \geq k/2) \\ &\leq \frac{\mathrm{Var}(Z(x_{i-1}^{\mathrm{opt}}, x_i^{\mathrm{pes}}))}{(k/2)^2} \quad \text{(by Chebyshev's inequality)} \quad \leq \frac{19n^2}{3k}. \end{aligned}$$

By applying these two probability bounds to Eq. (6), we have

$$\begin{aligned} &\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] \geq \frac{1}{en}\left(1 - \frac{19n^2}{3k}\right) - \frac{i(i+1)}{2n}\frac{19n^2}{3k} \\ &\geq \frac{1}{en}\left(1 - \frac{19}{30n^2}\right) - \frac{(n-1)n}{2n}\frac{19}{30n^2} \qquad \text{(by } i \leq n-1 \text{ and } k = 10n^4\text{)} \\ &\geq \frac{0.05}{n} - \frac{19}{30en^3}. \end{aligned}$$

Thus, condition of Lemma 1 holds with $\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] \geq \Omega(\frac{1}{n})$. We can get, noting that $V(x) = n - LO(x) \leq n$,

$$\mathbb{E}[\![\tau \mid \xi_0]\!] \leq O(n) \cdot V(\xi_0) \in O(n^2),$$

i.e., the expected iterations of the (1+1)-EA for finding the optimal solution is upper bounded by $O(n^2)$. Because the expected running time is $2k$ i.e., (the number of fitness evaluations in each iteration) $\times$ the expected iterations and $k = 10n^4$, we conclude that the expected running time is $O(n^6)$. □

4 Robustness to Posterior Noise

In the above section, we have shown that sampling can be robust to one-bit noise (a kind of prior noise) for the (1+1)-EA solving the OneMax and the LeadingOnes problems. In this section, by comparing the expected running time of the (1+1)-EA with or without sampling for solving OneMax and LeadingOnes under additive Gaussian noise, we will prove that sampling can also be robust to posterior noise.

4.1 The OneMax Problem

Additive Gaussian noise $\mathcal{N}(\theta, \sigma^2)$ with $\sigma^2 \geq 1$ is considered here. We first analyze the case in which sampling is not used. By applying the original simplified drift theorem (Oliveto and Witt, 2011, 2012), we prove that the expected running time is exponential, as shown in Theorem 6.

Theorem 6. *For the (1+1)-EA solving the OneMax problem under additive Gaussian noise $\mathcal{N}(\theta, \sigma^2)$ with $\sigma^2 \geq 1$, the expected running time is exponential.*

Proof. We use Lemma 3 to prove this theorem. Let X_t be the number of 0-bits of the solution after t iterations of the (1+1)-EA. We consider the interval $[0, n^{1/4}]$, i.e., the parameters $a = 0$ and $b = n^{1/4}$ in Lemma 3. Then, we analyze the drift $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!]$ for $1 \leq i < n^{1/4}$. Let $p_{i,i+d}$ denote the probability that the next solution after bit-wise mutation and selection has $i + d$ $(-i \leq d \leq n - i)$ number of 0-bits (i.e., $X_{t+1} = i + d$), and let P_d denote the probability that the offspring solution generated by bit-wise mutation has $i + d$ number of 0-bits (i.e., $|x'|_0 = i + d$). Then, we have, for $d \neq 0$,

$$\begin{aligned} p_{i,i+d} &= P_d \cdot P(f^n(x') \geq f^n(x)) = P_d \cdot P(n - i - d + \delta_1 \geq n - i + \delta_2) \\ &= P_d \cdot P(\delta_1 - \delta_2 \geq d) = P_d \cdot P(\delta \geq d), \end{aligned}$$

where $\delta_1, \delta_2 \sim \mathcal{N}(\theta, \sigma^2)$ and $\delta \sim \mathcal{N}(0, 2\sigma^2)$. We thus have

$$\begin{aligned} \mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] &= \sum_{d=1}^{i} d \cdot p_{i,i-d} - \sum_{d=1}^{n-i} d \cdot p_{i,i+d} \leq \sum_{d=1}^{i} d \cdot p_{i,i-d} - p_{i,i+1} \\ &\leq \sum_{d=1}^{i} d \cdot P_{-d} - P_1 \cdot P(\delta \geq 1). \end{aligned} \tag{7}$$

Let $\delta' \sim \mathcal{N}(0, 1)$. Then, $P(\delta \geq 1) = P(\delta' \geq \frac{1}{\sqrt{2}\sigma}) \geq P(\delta' \geq \frac{1}{\sqrt{2}}) = P(\delta' \leq -\frac{1}{\sqrt{2}}) \geq 0.23$, where the first inequality is by $\sigma \geq 1$, and the last one is obtained by calculating the CDF of the standard normal distribution. Furthermore, $P_1 \geq \frac{n-i}{n}(1 - \frac{1}{n})^{n-1} \geq \frac{n-i}{en}$, and $P_{-d} \leq \binom{i}{d}\frac{1}{n^d}$. Applying these probability bounds to Eq. (7), we have

$$\begin{aligned} &\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \leq \sum_{d=1}^{i} d \cdot \binom{i}{d}\frac{1}{n^d} - \frac{n-i}{en} \cdot 0.23 \\ &= -\frac{0.23}{e} + \frac{0.23}{e}\frac{i}{n} + \frac{i}{n}\left(1 + \frac{1}{n}\right)^{i-1} \leq -\frac{0.23}{e} + \left(\frac{0.23}{e} + e\right)\frac{i}{n} \\ &= -\frac{0.23}{e} + O\left(\frac{n^{1/4}}{n}\right). \quad (\text{since } i < n^{1/4}) \end{aligned}$$

Thus, $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] = -\Omega(1)$, which implies that condition 1 of Lemma 3 holds. For condition 2, we need to investigate $P(|X_{t+1} - X_t| \geq j \mid X_t \geq 1)$. Because it is necessary to flip at least j bits, we have

$$P(|X_{t+1} - X_t| \geq j \mid X_t \geq 1) \leq \binom{n}{j}\frac{1}{n^j} \leq \frac{1}{j!} \leq 2 \cdot \frac{1}{2^j},$$

which implies that condition 2 of Lemma 3 holds with $\delta = 1$ and $r(l) = 2$. Note that $l = b - a = n^{1/4}$. Thus, by Lemma 3, the expected running time is exponential. □

Note that Friedrich et al. (2015) have proved that for solving OneMax under additive Gaussian noise $\mathcal{N}(0,\sigma^2)$ with $\sigma^2 \geq n^3$, the classical (μ+1)-EA needs superpolynomial expected running time. Our result in Theorem 6 is complementary to their result with $\mu = 1$, since it covers a constant variance. We then prove in Corollary 1 that using sampling can reduce the expected running time to be polynomial. The proof idea is that sampling with a large enough k can reduce the noise to be $\sigma^2 = O(\log n/n)$, which allows a polynomial running time, as shown in the following lemma. In the following analysis, let $poly(n)$ indicate any polynomial of n.

Lemma 5. *(Gießen and Kötzing, 2016) Suppose posterior noise, sampling from some distribution D with variance σ^2. Then we have that the (1+1)-EA optimizes OneMax in polynomial time if $\sigma^2 = O(\log n/n)$.*

Corollary 1. *For the (1+1)-EA solving the OneMax problem under additive Gaussian noise $\mathcal{N}(\theta,\sigma^2)$ with $\sigma^2 \geq 1$ and $\sigma^2 \in O(poly(n))$, if using sampling with $k = \lceil \frac{n\sigma^2}{\log n} \rceil$, the expected running time is polynomial.*

Proof. The noisy fitness is $f^n(x) = f(x) + \delta$, where $\delta \sim \mathcal{N}(\theta,\sigma^2)$. The fitness output by sampling is $\hat{f}(x) = (\sum_{i=1}^{k} f_i^n(x))/k = (\sum_{i=1}^{k} f(x) + \delta_i)/k = f(x) + \sum_{i=1}^{k} \delta_i/k$, where $\delta_i \sim \mathcal{N}(\theta,\sigma^2)$. Thus, $\hat{f}(x) = f(x) + \delta'$, where $\delta' \sim \mathcal{N}(\theta, \frac{\sigma^2}{k})$. That is, sampling reduces the variance σ^2 of noise to be $\frac{\sigma^2}{k}$. Because $k = \lceil \frac{n\sigma^2}{\log n} \rceil$, we have $\frac{\sigma^2}{k} \leq \frac{\log n}{n}$. By Lemma 5, the expected iterations of the (1+1)-EA for finding the optimal solution is polynomial. We know that the expected running time is $2k \times$ the expected iterations. Since $\sigma^2 \in O(poly(n))$, the expected running time is polynomial. □

Thus, the comparison between Theorem 6 and Corollary 1 correct our previous statement in (Qian et al., 2014), that sampling is ineffective for the (1+1)-EA solving OneMax under additive Gaussian noise. We have conducted experiments to complement the theoretical results, which give bounds only. For the additive Gaussian noise, we set $\theta = 0$ and $\sigma = 1$. The results for $n = 10, 20, 30$ are plotted in Figure 2. Note that the point with $k = 1$ in the figure corresponds to the ERT without sampling.

From Figure 2(b and c), we can observe that the ERT has a fast drop at the beginning of the curve, reaches the minimum at a small sample size, and consistently grows after that. The minimum is much smaller than the value at $k = 1$, thus it is clear that a moderate sampling can reduce the running time from no sampling, which is consistent with our theoretical result. However, in Figure 2(a), the ERT always increases with k, which is similar to what was observed in Figure 1 in (Qian et al., 2014). The setting in (Qian et al., 2014) is $n = 10$, $\theta = 0$ and $\sigma = 10$. A too small n (e.g., $n = 10$) makes the decrease of the number of iterations easily dominated by the increase of k, therefore, we did not observe the dropping stage of the curve.

4.2 The LeadingOnes Problem

Additive Gaussian noise $\mathcal{N}(\theta,\sigma^2)$ with $\sigma^2 \geq n^2$ is considered here. We first analyze the case in which sampling is not used. Using the original simplified drift theorem (Oliveto and Witt, 2011, 2012), we prove that the expected running time is exponential, as shown in Theorem 7.

Theorem 7. *For the (1+1)-EA solving the LeadingOnes problem under additive Gaussian noise $\mathcal{N}(\theta,\sigma^2)$ with $\sigma^2 \geq n^2$, the expected running time is exponential.*

Proof. We use Lemma 3 to prove this theorem. Let X_t be the number of 0-bits of the

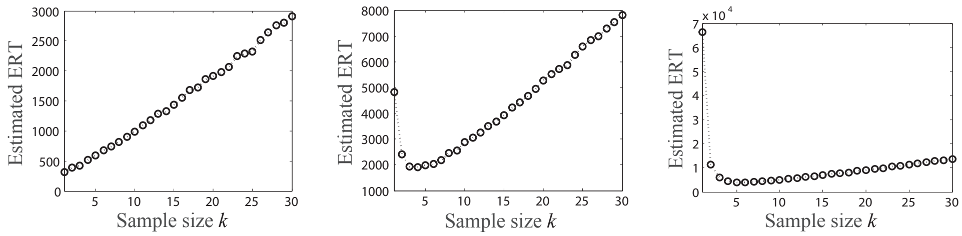


Figure 2: Estimated ERT for the (1+1)-EA using sampling on the OneMax problem under additive Gaussian noise with $\theta = 0$ and $\sigma = 1$.

solution after t iterations of the (1+1)-EA. As in the proof of Theorem 6, we have

$$\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \leq \sum_{d=1}^{i} d \cdot p_{i,i-d} - p_{i,i+1} \tag{8}$$

For $p_{i,i-d}$, we use $p_{i,i-d} \leq P_{-d} \leq \binom{i}{d}\frac{1}{n^d}$. For $p_{i,i+1}$, we consider $n-i$ possible cases such that one 1-bit is flipped and the other bits remain unchanged, the probability of which is $\frac{1}{n}(1-\frac{1}{n})^{n-1}$. Let x and x' denote the current solution and the offspring solution, respectively. Let LO denote the number of leading 1-bits of x, i.e., $LO = f(x)$. The noisy fitness of x is $f^n(x) = LO + \delta_1$, where $\delta_1 \sim \mathcal{N}(\theta, \sigma^2)$. Then, the acceptance probability of x' in these $n-i$ cases can be calculated as follows:
(1) If the flipped 1-bit is the j-th leading 1-bit, $f^n(x') = j - 1 + \delta_2$, where $1 \leq j \leq LO$ and $\delta_2 \sim \mathcal{N}(\theta, \sigma^2)$. Thus, the acceptance probability is $P(f^n(x') \geq f^n(x)) = P(j-1+\delta_2 \geq LO + \delta_1) = P(\delta_2 - \delta_1 \geq LO - j + 1) = P(\delta \geq LO - j + 1)$, where $\delta \sim \mathcal{N}(0, 2\sigma^2)$.
(2) Otherwise, $f^n(x') = LO + \delta_2$. Thus, the acceptance probability is $P(\delta \geq 0)$.

Applying these probability bounds to Eq. (8), we have

$$\begin{aligned}
&\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \\
&\leq \sum_{d=1}^{i} d \cdot \binom{i}{d}\frac{1}{n^d} - \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} \cdot \left(\sum_{j=1}^{LO} P(\delta \geq LO - j + 1) + (n-i-LO)P(\delta \geq 0)\right) \\
&\leq \frac{i}{n}\left(1 + \frac{1}{n}\right)^{i-1} - \frac{1}{en} \cdot \sum_{j=1}^{n-i} P(\delta \geq n - i - j + 1),
\end{aligned}$$

where the second "$\leq$" is since the term in $()$ reaches the minimum when $LO = n - i$. Let $\delta' \sim \mathcal{N}(0,1)$. Using $P(\delta' \geq u) \geq \frac{1}{2}(1 - \sqrt{1 - e^{-u^2}}) \geq \frac{1}{2}(1-u)$ for $u > 0$ (see Eq. (D.17) in (Mohri et al., 2012)), we have

$$\begin{aligned}
P(\delta \geq n - i - j + 1) &= P\left(\frac{\delta}{\sqrt{2}\sigma} \geq \frac{n-i-j+1}{\sqrt{2}\sigma}\right) = P\left(\delta' \geq \frac{n-i-j+1}{\sqrt{2}\sigma}\right) \\
&\geq \frac{1}{2}\left(1 - \frac{n-i-j+1}{\sqrt{2}\sigma}\right) \geq \frac{1}{2}\left(1 - \frac{n-i-j+1}{\sqrt{2}n}\right),
\end{aligned}$$

where the last inequality is by $\sigma \geq n$. Thus, we have

$$\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \leq \frac{i}{n}\left(1 + \frac{1}{n}\right)^{i-1} - \frac{1}{en} \cdot \sum_{j=1}^{n-i} \frac{1}{2}\left(1 - \frac{n-i-j+1}{\sqrt{2}n}\right)$$

$$= \frac{i}{n}\left(1+\frac{1}{n}\right)^{i-1} - \frac{n-i}{2en} \cdot \left(1 - \frac{n-i+1}{2\sqrt{2}n}\right) \leq -\frac{1}{2e}\left(1-\frac{1}{2\sqrt{2}}\right) + O\left(\frac{n^{\frac{1}{4}}}{n}\right),$$

where the last inequality is by $i < n^{\frac{1}{4}}$. Thus, $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] = -\Omega(1)$, which implies that condition 1 of Lemma 3 holds. As in the proof of Theorem 6, it is easy to verify that condition 2 of Lemma 3 holds with $\delta = 1$ and $r(l) = 2$. Thus, we can conclude that the expected running time is exponential. □

We then prove in Corollary 2 that using sampling can reduce the expected running time to be polynomial. The idea is that sampling with a large enough k can reduce the noise to be $\sigma^2 \leq 1/(12en^2)$, which allows a polynomial running time, as shown in the following lemma.

Lemma 6. *(Gießen and Kötzing, 2016) Suppose posterior noise, sampling from some distribution D with variance σ^2. Then we have that the (1+1)-EA optimizes LeadingOnes in $O(n^2)$ time if $\sigma^2 \leq 1/(12en^2)$.*

Corollary 2. *For the (1+1)-EA solving the LeadingOnes problem under additive Gaussian noise $\mathcal{N}(\theta, \sigma^2)$ with $\sigma^2 \geq n^2$ and $\sigma^2 \in O(poly(n))$, if using sampling with $k = \lceil 12en^2\sigma^2 \rceil$, the expected running time is polynomial.*

Proof. As in the proof of Corollary 1, sampling reduces the variance σ^2 of noise to be $\frac{\sigma^2}{k}$. Because $k = \lceil 12en^2\sigma^2 \rceil$, we have $\frac{\sigma^2}{k} \leq \frac{1}{12en^2}$. By Lemma 6, the expected iterations of the (1+1)-EA for finding the optimal solution is $O(n^2)$. Because the expected running time is $2k \times$ the expected iterations and $\sigma^2 \in O(poly(n))$, the expected running time is polynomial. □

Note that Akimoto et al. (2015) have investigated the running time of a general optimization algorithm solving a class of integer-valued functions under additive Gaussian noise. They have proved in Theorem 2.2 of (Akimoto et al., 2015) that with a high probability, the running time by using sampling is not larger than the noise-free running time by a logarithmic factor. Note that although their algorithm and problem setting covers the case we considered that the (1+1)-EA is used to solve OneMax and LeadingOnes, the employed sampling technique is different. A rounding function $\lfloor \hat{f} + 0.5 \rfloor$ is applied to the average fitness value $\hat{f}$ in (Akimoto et al., 2015), while we directly use the average value.

5 Comparing with Parent Populations and Threshold Selection

In the above two sections, we have proved that sampling is robust to both one-bit and additive Gaussian noise for the (1+1)-EA solving OneMax and LeadingOnes. Previous studies (Qian et al., 2015a; Gießen and Kötzing, 2016) have shown that populations and threshold selection can also bring robustness to noise. For solving the OneMax problem under one-bit noise with $p \in w(\log n/n)$ (i.e., the noise level is high), the expected running time of the (1+1)-EA is super-polynomial (Droste, 2004). Gießen and Kötzing (2016) proved that the (μ+1)-EA with $\mu \geq 12\log(15n)/p$ finds the optimal solution in time $O(\mu n \log n)$, and the (1+λ)-EA with $\lambda \geq \max\{12/p, 24\} n \log n$ needs $O((n^2 \log n + n^2\lambda)/p)$ time; Qian et al. (2015a) proved that the (1+1)-EA using threshold selection $\tau = 1$ needs time $O(n^2 \log n/p^2)$. That is, employing populations or threshold selection can reduce the running time to be polynomial. Thus, a natural question is whether sampling can be better than these two strategies? We give positive answers by analyzing two concrete examples.

5.1 Sampling vs. Parent Populations

To compare sampling with parent populations, we consider the OneMax problem under additive Gaussian noise $\mathcal{N}(\theta, \sigma^2)$. Recently, Friedrich et al. (2015) have proved that the (μ+1)-EA needs super-polynomial time when $\theta = 0$ and $\sigma^2 \geq n^3$, as shown in Theorem 8. In Corollary 1, we have proved that the (1+1)-EA using sampling with $k = \lceil \frac{n\sigma^2}{\log n} \rceil$ can solve the problem in polynomial time when $\sigma^2 \geq 1$ and $\sigma^2 \in O(poly(n))$. Thus, the comparison between Corollary 1 and Theorem 8 directly shows that sampling can be more tolerant of noise than parent populations.

Theorem 8. *(Friedrich et al., 2015) For the (μ+1)-EA solving the OneMax problem under additive Gaussian noise $\mathcal{N}(0, \sigma^2)$, if $\sigma^2 \geq n^3$ and μ is bounded from above by a polynomial in n, the expected running time is super-polynomial.*

5.2 Sampling vs. Threshold Selection

When proving the robustness of threshold selection to one-bit noise, Qian et al. (2015a) also showed its limitation by proving that the (1+1)-EA using threshold selection needs at least exponential time for solving the OneMax problem under asymmetric one-bit noise with $p = 1$, as shown in Theorem 9. To show that sampling can be better than threshold selection, we thus consider the asymmetric one-bit noise model here. We prove in Theorem 10 that the (1+1)-EA using sampling with $k = \lceil 24 \log n \rceil$ can solve OneMax in time $O(n \log^2 n)$. The proof is finished by using Lemma 2, and is similar to that of Theorem 3 except that the probabilities $p_{i,i+d}$ are different due to the difference on the noise and the value of k. The detailed proofs are shown in the supplementary material due to space limitations. The comparison between Theorems 9 and 10 shows that sampling can be more tolerant of noise than threshold selection.

Theorem 9. *(Qian et al., 2015a) For the (1+1)-EA solving the OneMax problem under asymmetric one-bit noise with $p = 1$, if using threshold selection $\tau \geq 0$, the expected running time is at least exponential.*

Theorem 10. *For the (1+1)-EA solving the OneMax problem with $n \geq 7$ under asymmetric one-bit noise with $p = 1$, if using sampling with $k = \lceil 24 \log n \rceil$, the expected running time is $O(n \log^2 n)$.*

6 The Ineffectiveness of Sampling

In the previous sections, we have shown that sampling is an effective strategy to cope with noise. Then, a natural question is that if sampling can be always effective. We give a negative answer by analyzing the (1+1)-EA solving the Trap problem under additive Gaussian noise. We first prove in Lemma 7 that noise does not bring a negative impact in this case, which means that the expected running time for finding an optimal solution under noise is not larger than that without noise. The proof is by applying additive drift analysis (He and Yao, 2001).

Lemma 7. *For the (1+1)-EA solving the Trap problem with $C \geq n + 1$ under additive Gaussian noise $\mathcal{N}(\theta, \sigma^2)$ with $\sigma^2 \leq n^2/(8 \log n)$, when $n \geq 8$, noise does not bring a negative impact, i.e., the expected running time is not larger than that without noise.*

Proof. The two EAs with and without noise are different only on whether the fitness evaluation is disturbed by noise, thus, they have the same number of fitness evaluations in each iteration. Then, comparing their expected running time is equivalent to comparing their expected iterations. Let Markov chains $\{\xi_t\}_{t=0}^{+\infty}$ and $\{\xi'_t\}_{t=0}^{+\infty}$ model the (1+1)-EA with additive Gaussian noise and without noise for maximizing the Trap problem, respectively. Let $\mathbb{E}[x]$ and $\mathbb{E}'[x]$ denote their EFHT when starting from x, respectively. Thus, our goal is to show that $\mathbb{E}[x] \leq \mathbb{E}'[x]$.

We use Lemma 1 to derive an upper bound on $\mathbb{E}[x]$. Let the distance function be $\forall x \in \mathcal{X} = \{0,1\}^n : V(x) = \mathbb{E}'[x]$. Then, it is easy to verify that $V(x \in \mathcal{X}^* = \{1^n\}) = 0$ and $V(x \notin \mathcal{X}^*) > 0$. From Lemma 2 in (Qian et al., 2012), we know that $\mathbb{E}'[x]$ only depends on the number of 0-bits of x (denoted by i), and it increases with i. Let $\mathbb{E}'[i]$ denote $\mathbb{E}'[x]$ with $|x|_0 = i$, then we have $\mathbb{E}'[0] = 0 < \mathbb{E}'[1] < \ldots < \mathbb{E}'[n]$.

We are then to analyze the drift $\mathbb{E}[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]$ for any x with $|x|_0 \geq 1$. Let i (where $1 \leq i \leq n$) denote the number of 0-bits of the current solution x. Let $mut_{i,j}$ be the probability that the offspring solution x' generated by mutating x has j number of 0-bits. For the two chains $\{\xi_t\}_{t=0}^{+\infty}$ and $\{\xi'_t\}_{t=0}^{+\infty}$, let $p_{i,j}$ and $p'_{i,j}$ be the probabilities that the next solution after mutation and selection has j number of 0-bits, respectively. For the (1+1)-EA solving Trap without noise (i.e., $\{\xi'_t\}_{t=0}^{+\infty}$), only the offspring solution with more 0-bits or the optimal solution 1^n will be accepted, thus

$$p'_{i,0} = mut_{i,0}, \quad \forall 0 < j < i : p'_{i,j} = 0, \quad p'_{i,i} = \sum_{j=1}^{i} mut_{i,j}, \quad \forall j > i : p'_{i,j} = mut_{i,j}.$$

For the (1+1)-EA solving Trap under noise (i.e., $\{\xi_t\}_{t=0}^{+\infty}$), due to the fitness evaluation disturbed by noise, the offspring solution with more 0-bits and the optimal solution may be rejected, while the offspring solution with less 0-bits may be accepted. Thus,

$$p_{i,0} \leq mut_{i,0}, \qquad \forall 0 < j < i : p_{i,j} \geq 0, \qquad \forall j > i : p_{i,j} \leq mut_{i,j}.$$

We then have

$$\begin{aligned} &\mathbb{E}[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x] = V(x) - \mathbb{E}[V(\xi_{t+1}) \mid \xi_t = x] \qquad (9)\\ &= \mathbb{E}'[i] - \sum_{j=0}^{n} p_{i,j}\mathbb{E}'[j] = 1 + \sum_{j=0}^{n} p'_{i,j}\mathbb{E}'[j] - \sum_{j=0}^{n} p_{i,j}\mathbb{E}'[j] \\ &= 1 - (p'_{i,0} - p_{i,0})\mathbb{E}'[i] + \sum_{j=1}^{i-1} p_{i,j}(\mathbb{E}'[i] - \mathbb{E}'[j]) + \sum_{j=i+1}^{n} (p'_{i,j} - p_{i,j})(\mathbb{E}'[j] - \mathbb{E}'[i]), \end{aligned}$$

where the last equality can be derived by using the above calculation of $p'_{i,j}$ and $p_{i,j}$.

Let $\delta_1, \delta_2 \sim \mathcal{N}(\theta, \sigma^2)$, $\delta \sim \mathcal{N}(0, 2\sigma^2)$ and $\delta' \sim \mathcal{N}(0,1)$. Let j denote the number of 0-bits of the offspring solution x' generated by mutation. For $j \geq 1 \wedge j \neq i$, we have

$$\begin{aligned} p_{i,j} &= mut_{i,j} \cdot P(f^n(x') \geq f^n(x)) = mut_{i,j} \cdot P(-(n-j) + \delta_1 \geq -(n-i) + \delta_2) \\ &= mut_{i,j} \cdot P(\delta \geq i - j) = mut_{i,j} \cdot P(\delta' \geq (i-j)/(\sqrt{2}\sigma)). \end{aligned}$$

For $j = 0$ (i.e., the offspring solution x' is optimal), we have

$$\begin{aligned} p_{i,0} &= mut_{i,0} \cdot P(f^n(1^n) \geq f^n(x)) = mut_{i,0} \cdot P(C - n + \delta_1 \geq -(n-i) + \delta_2) \\ &= mut_{i,0} \cdot P(\delta \leq C - i) \geq mut_{i,0} \cdot P(\delta' \leq (n+1-i)/(\sqrt{2}\sigma)), \end{aligned}$$

where the inequality is by $C \geq n+1$.

We then derive a lower bound on $\mathbb{E}'[i+1] - \mathbb{E}'[i]$ for $i \geq 1$. In (Qian et al., 2012), it has been proved that $\mathbb{E}'[i+1] > \mathbb{E}'[i]$. In their proof of Lemma 2, they derived that

$$\mathbb{E}'[i+1] - \mathbb{E}'[i] = P_0 \cdot (1-2p)\mathbb{E}'[i+1] + \left(\sum\nolimits_{j=1}^{i-1} P_j + (1-p)P_i + pP_0\right)$$

$$\cdot \left(\mathbb{E}'[i+1]-\mathbb{E}'[i]\right)+(1-2p)\cdot\left(\sum\nolimits_{j=i+1}^{n-1}P_j(\mathbb{E}'[j+1]-\mathbb{E}'[j])\right),$$

where P_j $(0 \le j \le n-1)$ denotes the probability that the number of 0-bits changes to be j after bit-wise mutation on a Boolean string $z \in \{0,1\}^{n-1}$ with $|z|_0 = i$. The proof idea is to utilize a common mutation on a Boolean string $z \in \{0,1\}^{n-1}$ with $|z|_0 = i$ for the expansion of $\mathbb{E}'[i+1]$ and $\mathbb{E}'[i]$, because the bit-wise mutation on a solution with $i+1$ 0-bits can be equivalently divided into mutation on one 0-bit and mutation on z, and the mutation on a solution with i 0-bits can be divided into mutation on one 1-bit and mutation on z. Note that they considered a general setting for the mutation probability of the (1+1)-EA, i.e., each bit is flipped independently with a probability of $p \in (0, 0.5)$. Since $\mathbb{E}'[i+1] > \mathbb{E}'[i]$ for any $i \ge 0$, we get

$$\mathbb{E}'[i+1]-\mathbb{E}'[i] \ge P_0\cdot(1-2p)\mathbb{E}'[i+1] = \frac{1}{n^i}\left(1-\frac{1}{n}\right)^{n-1-i}\cdot\left(1-\frac{2}{n}\right)\mathbb{E}'[i+1],$$

where the equality is obtained by replacing p with $\frac{1}{n}$ and using $P_0 = \frac{1}{n^i}(1-\frac{1}{n})^{n-1-i}$.

Based on the above analysis, we can simplify Eq. (9). For $i \ge 2$, we have

$$\begin{aligned}
&\mathbb{E}[V(\xi_t)-V(\xi_{t+1}) \mid \xi_t = x] \ge 1-(p'_{i,0}-p_{i,0})\mathbb{E}'[i]+p_{i,i-1}(\mathbb{E}'[i]-\mathbb{E}'[i-1])\\
&\ge 1-mut_{i,0}\cdot P\left(\delta' \ge \frac{n+1-i}{\sqrt{2}\sigma}\right)\cdot\mathbb{E}'[i]\\
&\quad + mut_{i,i-1}\cdot P\left(\delta' \ge \frac{1}{\sqrt{2}\sigma}\right)\cdot\frac{1}{n^{i-1}}\left(1-\frac{1}{n}\right)^{n-i}\cdot\left(1-\frac{2}{n}\right)\mathbb{E}'[i]\\
&\ge 1-\frac{1}{n^i}\left(1-\frac{1}{n}\right)^{n-i}\cdot P\left(\delta' \ge \frac{n+1-i}{\sqrt{2}\sigma}\right)\cdot\mathbb{E}'[i]\\
&\quad + \frac{i}{n}\left(1-\frac{1}{n}\right)^{n-1}\cdot P\left(\delta' \ge \frac{1}{\sqrt{2}\sigma}\right)\cdot\frac{1}{n^{i-1}}\left(1-\frac{1}{n}\right)^{n-i}\cdot\left(1-\frac{2}{n}\right)\mathbb{E}'[i].
\end{aligned}$$

For $i = 1$, we have

$$\begin{aligned}
&\mathbb{E}[V(\xi_t)-V(\xi_{t+1}) \mid \xi_t = x] \ge 1-(p'_{1,0}-p_{1,0})\mathbb{E}'[1]+(p'_{1,2}-p_{1,2})(\mathbb{E}'[2]-\mathbb{E}'[1])\\
&\ge 1-\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}\cdot P\left(\delta' \ge \frac{n}{\sqrt{2}\sigma}\right)\cdot\mathbb{E}'[1]\\
&\quad + \frac{n-1}{n}\left(1-\frac{1}{n}\right)^{n-1}\cdot P\left(\delta' \ge \frac{1}{\sqrt{2}\sigma}\right)\cdot\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-2}\cdot\left(1-\frac{2}{n}\right)\mathbb{E}'[1].
\end{aligned}$$

Thus, in order to show that $\mathbb{E}[V(\xi_t)-V(\xi_{t+1}) \mid \xi_t = x] \ge 1$, we need to prove that, for $1 \le i \le n$, the following inequality holds:

$$i\left(1-\frac{1}{n}\right)^{n-1}\left(1-\frac{2}{n}\right)\cdot P\left(\delta' \ge \frac{1}{\sqrt{2}\sigma}\right) \ge P\left(\delta' \ge \frac{n+1-i}{\sqrt{2}\sigma}\right). \tag{10}$$

Using the inequality for the standard normal distribution $\frac{1}{\sqrt{2\pi}}\frac{t}{t^2+1}e^{-\frac{t^2}{2}} < P(\delta' \ge t) < \frac{1}{\sqrt{2\pi}}\frac{1}{t}e^{-\frac{t^2}{2}}$ for $t > 0$ (Cook, 2009), we get

$$\frac{P\left(\delta' \ge \frac{1}{\sqrt{2}\sigma}\right)}{P\left(\delta' \ge \frac{n+1-i}{\sqrt{2}\sigma}\right)} > \frac{\frac{1}{\sqrt{2\pi}}\frac{\frac{1}{\sqrt{2}\sigma}}{\frac{1}{2\sigma^2}+1}e^{-\frac{1}{4\sigma^2}}}{\frac{1}{\sqrt{2\pi}}\frac{\sqrt{2}\sigma}{n+1-i}e^{-\frac{(n+1-i)^2}{4\sigma^2}}} = \frac{n+1-i}{1+2\sigma^2}e^{\frac{(n+1-i)^2-1}{4\sigma^2}}.$$

Thus, we only need to show that

$$\frac{i}{e}\left(1-\frac{2}{n}\right)\frac{n+1-i}{1+2\sigma^2}e^{\frac{(n+1-i)^2-1}{4\sigma^2}} \geq 1.$$

For $i=1$, we have

$$\frac{1}{e}\left(1-\frac{2}{n}\right)\frac{n}{1+2\sigma^2}e^{\frac{n^2-1}{4\sigma^2}} = \frac{1}{e}\frac{n-2}{1+2\sigma^2}e^{\frac{n^2-1}{4\sigma^2}} \geq \frac{1}{e}\frac{n}{2(1+2\sigma^2)}e^{\frac{n^2}{8\sigma^2}}$$
$$\geq \frac{1}{e}\frac{n}{2(1+n^2/(4\log n))}n \geq \frac{1}{e}\cdot\frac{3}{2}\log n \geq 1,$$

where the first inequality holds with $n \geq 4$, the second inequality is by $\sigma^2 \leq n^2/(8\log n)$, the third inequality is obtained by using $1+n^2/(4\log n) \leq n^2/(3\log n)$ for $n \geq 5$, and the last one holds with $n \geq 7$.
For $i=2$, we can similarly get, for $n \geq 6$,

$$\frac{2}{e}\left(1-\frac{2}{n}\right)\frac{n-1}{1+2\sigma^2}e^{\frac{(n-1)^2-1}{4\sigma^2}} \geq \frac{2}{e}\frac{n}{2(1+2\sigma^2)}e^{\frac{n^2}{8\sigma^2}} \geq \frac{2}{e}\log n \geq 1.$$

For $i=3$, we have, for $n \geq 8$,

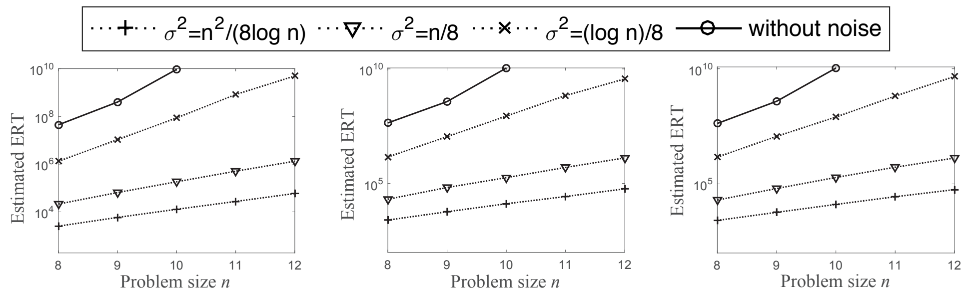
$$\frac{3}{e}\left(1-\frac{2}{n}\right)\frac{n-2}{1+2\sigma^2}e^{\frac{(n-2)^2-1}{4\sigma^2}} \geq \frac{3}{e}\frac{n}{2(1+2\sigma^2)}e^{\frac{n^2}{8\sigma^2}} \geq \frac{3}{e}\log n \geq 1.$$

For $i \geq 4$, Eq. (10) trivially holds, since $\frac{i}{e}(1-\frac{2}{n}) \geq 1$ for $n \geq 7$.

Thus, we can conclude that when $n \geq 8$, $\mathbb{E}[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x] \geq 1$, which implies that the condition of Lemma 1 holds with $c=1$. We then get that $\forall x \in \mathcal{X}, \mathbb{E}[x] \leq V(x) = \mathbb{E}'[x]$, and thus the lemma holds. □

We have also conducted experiments to verify the theoretical results. We compare the estimated expected running time (ERT) of the (1+1)-EA without noise and with additive Gaussian noise. For the parameter C of the Trap problem, we test $n+1$, n^2 and 2^n, respectively; for additive Gaussian noise, we set $\theta=0$, and the variance σ^2 is set as $(\log n)/8$, $n/8$ and $n^2/(8\log n)$, respectively. We test the problem size n from 8 to 12. For each value of C and n, the ERT of the (1+1)-EA is estimated by the average number of fitness evaluations of 1000 independent runs. The results are plotted in Figure 3. We can observe that the curves with noise are always under the curve without noise, which is consistent with our theoretical result. Furthermore, we can make a clear observation that the curve with larger noise variance is lower, which implies that the positive impact of noise increases with the level of noise. Note that the curves are almost the same for different C values. This is because the value of C only affects the acceptance probability of the optimal solution (which is always at least $1/2$), and thus the behavior of the (1+1)-EA is almost the same for different C values.

For the (1+1)-EA solving the Trap problem under additive Gaussian noise, noise may bring a positive impact and sampling will reduce the noise level (i.e., the variance of noise is reduced by a factor of k). Under the assumption that the positive impact of noise increases with the level of noise (observed in Figure 3), we can conclude that sampling will decrease the positive impact and then increase the running time, as shown in Conjecture 1. When the noise level is very large, i.e., $\sigma^2 \to +\infty$, the noise will completely obscure the underlying deception of the Trap problem, and the (1+1)-EA behaves similar to that on the Needle problem (i.e., a random walk), the running

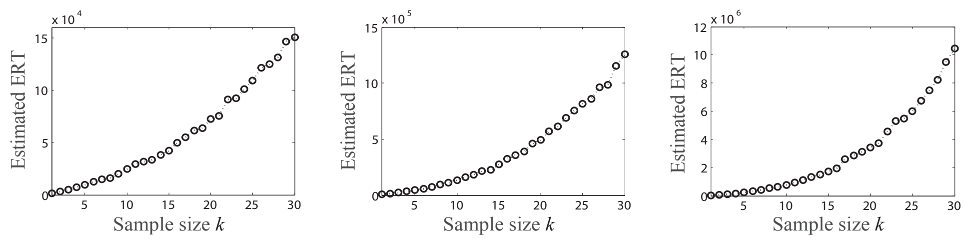


(a) $C = n + 1$ (b) $C = n^2$ (c) $C = 2^n$

Figure 3: Estimated ERT comparison for the (1+1)-EA solving the Trap problem with or without noise, where a base 10 logarithmic scale is used for the y-axis.

time of which is $2^{\Omega(n)}$ (Oliveto and Witt, 2011). As the sample size k increases, the noise level σ^2/k decreases. When the noise tends to disappear, i.e., $\sigma^2/k \to 0$, the (1+1)-EA on Trap will behave close to that without noise, the expected running time of which is $\Theta(n^n)$ (Droste et al., 2002). A rigorous analysis will be studied in the future, but we verify the conjecture by experiments here. For additive Gaussian noise, we set $\theta = 0$ and $\sigma = 10$. The parameter C of the Trap problem is set to be $n + 1$. The results for the problem size $n = 8, 10, 12$ are plotted in Figure 4. We can observe that the ERT always increases with k, which is consistent with our conjecture.

Conjecture 1. *For the (1+1)-EA solving the Trap problem under additive Gaussian noise, sampling is ineffective, i.e., using sampling will increase the running time.*



(a) $n = 8$ (b) $n = 10$ (c) $n = 12$

Figure 4: Estimated ERT for the (1+1)-EA using sampling on the Trap problem under additive Gaussian noise $\mathcal{N}(\theta, \sigma^2)$ with $\theta = 0$ and $\sigma = 10$.

7 Conclusion

EAs have been widely applied to solve noisy optimization problems, which are often encountered in real-world optimization tasks. Sampling is a popular method to reduce the negative effect of noise in evolutionary optimization. Previous studies mainly focused on empirical analysis and design of efficient sampling strategies, while the impact of sampling on the running time is unclear theoretically. In this paper, we study the effectiveness of sampling by rigorous running time analysis. Firstly, we show that sampling can speed up noisy evolutionary optimization exponentially. For the (1+1)-EA solving the OneMax and the LeadingOnes problems under one-bit or additive Gaussian noise, we prove that when the noise level is high, sampling can reduce the running time from exponential to be polynomial. The analysis also shows that a gap of one on the sample size k can lead to an exponential difference on the expected running time, which indicates that a careful selection of k is important for the effectiveness of sampling. Secondly, we prove that sampling can be better than using parent populations and threshold selection, two strategies that have been proven to be robust to noise. Finally, we also show that sampling can be ineffective when the noise does not have a negative impact.

8 Acknowledgements

We would like to thank Per Kristian Lehre for helpful discussions during the PPSN'14 conference, and thank the associate editor and anonymous reviewers for helpful comments and suggestions. This work was partially supported by the NSFC (61329302, 61333014, 61375061, 61603367, 61672478), the Joint Research Fund for Overseas Chinese, Hong Kong and Macao Scholars of the NSFC (61428302), the EPSRC (EP/K001523/1), the Jiangsu Science Foundation (BK20160066), the Royal Society Newton Advanced Fellowship (NA150123), the Fundamental Research Funds for the Central Universities (WK2150110002), the CCF-Tencent Open Research Fund, and the Collaborative Innovation Center of Novel Software Technology and Industrialization. Xin Yao was also supported by a Royal Society Wolfson Research Merit Award.

References

Aizawa, A. N. and Wah, B. W. (1994). Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2(2):97–122.

Akimoto, Y., Astete-Morales, S., and Teytaud, O. (2015). Analysis of runtime of optimization algorithms for noisy functions over discrete codomains. *Theoretical Computer Science*, 605:42–50.

Arnold, D. V. (2002). *Noisy Optimization with Evolution Strategies*. Kluwer, Norwell, MA.

Arnold, D. V. and Beyer, H.-G. (2006). A general noise model and its effects on evolution strategy performance. *IEEE Transactions on Evolutionary Computation*, 10(4):380–391.

Auger, A. and Doerr, B. (2011). *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific, Singapore.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK.

Bartz-Beielstein, T. (2005). Evolution strategies and threshold selection. In *Proceedings of the 2nd International Workshop on Hybrid Metaheuristics*, pages 104–115, Barcelona, Spain.

Beyer, H.-G. (2000). Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanics and Engineering*, 186(2):239–267.

Bianchi, L., Dorigo, M., Gambardella, L. M., and Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287.

Branke, J. and Schmidt, C. (2003). Selection in the presence of noise. In *Proceedings of the 5th ACM Conference on Genetic and Evolutionary Computation*, pages 766–777, Chicago, IL.

Branke, J. and Schmidt, C. (2004). Sequential sampling in noisy environments. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*, pages 202–211, Birmingham, UK.

Cantú-Paz, E. (2004). Adaptive sampling for noisy problems. In *Proceedings of the 6th ACM Conference on Genetic and Evolutionary Computation*, pages 947–958, Seattle, WA.

Cook, J. D. (2009). Upper and lower bounds for the normal distribution function. http://www.johndcook.com/normalbounds.pdf.

Dang, D.-C. and Lehre, P. K. (2015a). Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In *Proceedings of the 13th ACM Conference on Foundations of Genetic Algorithms*, pages 62–68, Aberystwyth, UK.

Dang, D.-C. and Lehre, P. K. (2015b). Runtime analysis of non-elitist populations: From classical optimisation to partial information. *Algorithmica*, 75(3):428–461.

Doerr, B. and Goldberg, L. A. (2013). Adaptive drift analysis. *Algorithmica*, 65(1):224–250.

Doerr, B., Hota, A., and Kötzing, T. (2012a). Ants easily solve stochastic shortest path problems. In *Proceedings of the 14th ACM Conference on Genetic and Evolutionary Computation*, pages 17–24, Philadelphia, PA.

Doerr, B., Johannsen, D., and Winzen, C. (2012b). Multiplicative drift analysis. *Algorithmica*, 64(4):673–697.

Droste, S. (2004). Analysis of the (1+1) EA for a noisy OneMax. In *Proceedings of the 6th ACM Conference on Genetic and Evolutionary Computation*, pages 1088–1099, Seattle, WA.

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

Feldmann, M. and Kötzing, T. (2013). Optimizing expected path lengths with ant colony optimization using fitness proportional update. In *Proceedings of the 12th International Workshop on Foundations of Genetic Algorithms*, pages 65–74, Adelaide, Australia.

Friedrich, T., Kötzing, T., Krejca, M., and Sutton, A. M. (2015). The benefit of sex in noisy evolutionary search. *arXiv:1502.02793*.

Friedrich, T., Kötzing, T., Krejca, M. S., and Sutton, A. M. (2016). Robustness of ant colony optimization to noise. *Evolutionary Computation*, 24(2):237–254.

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

Gutjahr, W. J. (2003). A converging ACO algorithm for stochastic combinatorial optimization. In *Proceedings of the 2nd International Symposium on Stochastic Algorithms: Foundations and Applications*, pages 10–25, Hatfield, UK.

Gutjahr, W. J. (2012). Runtime analysis of an evolutionary algorithm for stochastic multi-objective combinatorial optimization. *Evolutionary Computation*, 20(3):395–421.

He, J. and Yao, X. (2001). Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85.

He, J. and Yao, X. (2003). Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence*, 145(1):59–97.

He, J. and Yao, X. (2004). A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3(1):21–35.

Heidrich-Meisner, V. and Igel, C. (2009). Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th International Conference on Machine Learning*, pages 401–408, Montreal, Canada.

Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.

Markon, S., Arnold, D. V., Back, T., Bartz-Beielstein, T., and Beyer, H.-G. (2001). Thresholding-A selection operator for noisy ES. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pages 465–472, Seoul, Korea.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. MIT press, London, UK.

Neumann, F. and Witt, C. (2010). *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*. Springer-Verlag, Berlin, Germany.

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

Oliveto, P. S. and Witt, C. (2012). Erratum: Simplified drift analysis for proving lower bounds in evolutionary computation. *CORR abs/1211.7184*.

Paixão, T., Pérez Heredia, J., Sudholt, D., and Trubenová, B. (2015). First steps towards a runtime comparison of natural and artificial evolution. In *Proceedings of the 17th ACM Conference on Genetic and Evolutionary Computation*, pages 1455–1462, Madrid, Spain.

Prugel-Bennett, A., Rowe, J., and Shapiro, J. (2015). Run-time analysis of population-based evolutionary algorithm in noisy environments. In *Proceedings of the 13th ACM Conference on Foundations of Genetic Algorithms*, pages 69–75, Aberystwyth, UK.

Qian, C., Yu, Y., Jin, Y., and Zhou, Z.-H. (2014). On the effectiveness of sampling for evolutionary optimization in noisy environments. In *Proceedings of the 13th International Conference on Parallel Problem Solving from Nature*, pages 302–311, Ljubljana, Slovenia.

Qian, C., Yu, Y., and Zhou, Z.-H. (2012). On algorithm-dependent boundary case identification for problem classes. In *Proceedings of the 12th International Conference on Parallel Problem Solving from Nature*, pages 62–71, Taormina, Italy.

Qian, C., Yu, Y., and Zhou, Z.-H. (2015a). Analyzing evolutionary optimization in noisy environments. *Evolutionary Computation*, in press.

Qian, C., Yu, Y., and Zhou, Z.-H. (2015b). Variable solution structure can be helpful in evolutionary optimization. *Science China Information Sciences*, 58(11):1–17.

Rolet, P. and Teytaud, O. (2010). Bandit-based estimation of distribution algorithms for noisy optimization: Rigorous runtime analysis. In *Proceedings of the 4th International Conference on Learning and Intelligent Optimization*, pages 97–110, Venice, Italy.

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

Stagge, P. (1998). Averaging efficiently in the presence of noise. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 188–197, Amsterdam, The Netherlands.

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

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

Wegener, I. (2002). Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. In Ruhul A. Sarker, M. M. and Yao, X., editors, *Evolutionary Optimization*. Kluwer.

Yu, Y. and Qian, C. (2015). Running time analysis: Convergence-based analysis reduces to switch analysis. In *Proceedings of the 2015 IEEE Congress on Evolutionary Computation*, pages 2603–2610, Sendai, Japan.

Yu, Y., Qian, C., and Zhou, Z.-H. (2015). Switch analysis for running time analysis of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 19(6):777–792.

Yu, Y. and Zhou, Z.-H. (2008). A new approach to estimating the expected first hitting time of evolutionary algorithms. *Artificial Intelligence*, 172(15):1809–1832.

Zeng, X., Yuan, S., Huang, X., and Zou, Q. (2015). Identification of cytokine via an improved genetic algorithm. *Frontiers of Computer Science*, 9(4):643–651.