# Towards a Running Time Analysis of the (1+1)-EA for OneMax and LeadingOnes Under General Bit-Wise Noise

Chao Bian[1], Chao Qian[1(✉)], and Ke Tang[2]

[1] Anhui Province Key Lab of Big Data Analysis and Application,
University of Science and Technology of China, Hefei 230027, China
biancht@mail.ustc.edu.cn, chaoqian@ustc.edu.cn
[2] Shenzhen Key Lab of Computational Intelligence,
Southern University of Science and Technology, Shenzhen 518055, China
tangk3@sustc.edu.cn

**Abstract.** Running time analysis of evolutionary algorithms (EAs) under noisy environments has recently received much attention, which can help us understand the behavior of EAs in practice where the fitness evaluation is often subject to noise. One of the mainly investigated noise models is bit-wise noise, which is characterized by a pair $(p, q)$ of parameters. However, previous analyses usually fix $p$ or $q$, which makes our understanding on bit-wise noise incomplete. In this paper, we analyze the running time of the (1+1)-EA solving OneMax and LeadingOnes under general bit-wise noise. Our results largely extend the known ones in specific cases of bit-wise noise, and disclose that $p$ and $pq$ together decide the running time to be polynomial or super-polynomial.

## 1 Introduction

Evolutionary algorithms (EAs) have been widely applied to solve real-world optimization problems, where the fitness evaluation of a solution is often disturbed by noise. Although they have shown good empirical performances in noisy optimization [8], it is also important to understand the impact of noise from a theoretical viewpoint. As an essential theoretical aspect of EAs, the running time analysis has received much attention in the last two decades, and has achieved many promising results [1,9]. However, most of them focus on noise-free optimization, where the fitness evaluation is exact. The theoretical understanding of EAs is still largely incomplete for the noisy case.

Previous running time analyses in noisy evolutionary optimization mainly considered two kinds of noise models, posterior and prior. The posterior noise

This work was supported by the Ministry of Science and Technology of China (2017YFC0804003), the NSFC (61603367, 61672478), the YESS (2016QNRC001), the Science and Technology Innovation Committee Foundation of Shenzhen (ZDSYS201703031748284), and the Royal Society Newton Advanced Fellowship (NA150123).

A. Auger et al. (Eds.): PPSN 2018, LNCS 11102, pp. 165–177, 2018.
https://doi.org/10.1007/978-3-319-99259-4_14

comes from the variation on the fitness of a solution, e.g., additive noise adds a value randomly drawn from some distribution. There was a sequence of papers [2,4,5,10] mainly showing that some specific EAs (e.g., compact genetic algorithm and ant colony optimization) can well handle additive noise, and using populations can bring robustness to additive noise.

For the prior noise coming from the variation on a solution, Droste [3] first considered a specific model, one-bit noise, which flips a randomly chosen bit of a binary solution before evaluation with probability $p$. He analyzed the (1+1)-EA solving the OneMax problem under one-bit noise, and proved that the maximal value of $p$ allowing a polynomial running time is $O(\log n/n)$. Gießen and Kötzing [6] considered another problem LeadingOnes, and proved that the expected running time is polynomial if $p \leq 1/(6en^2)$ and exponential if $p = 1/2$. For these two problems, Qian et al. [11,12,14] theoretically studied the effectiveness of two noise-handling strategies, threshold selection and resampling, and proved that under some large values of $p$, both of them can reduce the running time of the (1+1)-EA from exponential to polynomial.

Gießen and Kötzing [6] also studied another prior noise model, which flips each bit of a solution independently with probability $q$ before evaluation. They proved that for the (1+1)-EA on OneMax, the maximal $q$ allowing a polynomial running time is $O(\log n/n^2)$. By combining this model with one-bit noise, Qian et al. [13] studied a general prior noise model, bit-wise noise, which is characterized by a pair $(p, q)$ of parameters. It happens with probability $p$, and independently flips each bit of a solution with probability $q$ before evaluation. Their derived results for the (1+1)-EA on OneMax and LeadingOnes are shown in the middle two columns of Table 1, which give the ranges of $p$ and $q$ for a polynomial running time upper bound as well as a super-polynomial lower bound.

**Table 1.** For the expected running time of the (1+1)-EA on OneMax and LeadingOnes under bit-wise noise $(p, q)$, the ranges of $p, q$ for a polynomial upper bound (the first line in each cell) and a super-polynomial lower bound (the second line) are shown below.

| (1+1)-EA | $(p, 1/n)$ | $(1, q)$ | $(p, q)$ |
|---|---|---|---|
| OneMax | $O(\log n/n)$<br>$\omega(\log n/n)$ [13] | $O(\log n/n^2)$<br>$\omega(\log n/n^2)$ [6] | $p = O(\log n/n) \vee pq = O(\log n/n^2)$<br>$p = \omega(\log n/n) \wedge pq = \omega(\log n/n^2)$ |
| LeadingOnes | $O(\log n/n^2)$<br>$\omega(\log n/n)$ [13] | $O(\log n/n^3)$<br>$\omega(\log n/n^2)$ [13] | $p = O(\log n/n^2) \vee pq = O(\log n/n^3)$<br>$p = \omega(\log n/n) \wedge pq = \omega(\log n/n^2)$ |

However, the analysis for bit-wise noise [13] only considered two specific cases: $(p, 1/n)$ (i.e., $q$ is fixed to $1/n$) and $(1, q)$ (i.e., $p$ is fixed to 1). Some fundamental theoretical issues are thus not addressed. For example, what is the deciding factor for the running time of the (1+1)-EA under bit-wise noise? Will the (1+1)-EA perform similarly for two scenarios $(p, q)$ and $(p', q')$ with $pq = p'q'$? In this paper, we analyze the running time of the (1+1)-EA solving OneMax and LeadingOnes under general bit-wise noise, i.e., without fixing $p$ or $q$. We derive the ranges of $p$ and $q$ for a polynomial upper bound and a super-polynomial lower bound, which are summarized in the last column of Table 1. It is easy to verify that our results cover previously known ones [6,13], as shown in the middle two columns of Table 1. The results show that $p$ and $pq$ together decide the running time to be polynomial or super-polynomial. We can also observe that the expected running time for two scenarios $(p, q)$ and $(p', q')$ with $pq = p'q'$ may be significantly different, e.g., for the (1+1)-EA on OneMax, the running time is polynomial if $p = q = \log n/n$, but super-polynomial if $p' = 1, q' = (\log n/n)^2$.

The rest of this paper is organized as follows. Section 2 introduces some preliminaries. Sections 3 and 4 present the running time analysis on OneMax and LeadingOnes, respectively. Section 5 concludes the paper.

## 2 Preliminaries

In this section, we first introduce the optimization problems, noise models and EAs studied in this paper, respectively, then present the analysis tools that we use throughout this paper.

### 2.1 OneMax and LeadingOnes

In this paper, we consider two pseudo-Boolean problems OneMax and LeadingOnes, which are widely used in EAs' theoretical analyses [1,9]. The former aims to maximize the number of 1-bits of a solution, and the latter aims to maximize the number of consecutive 1-bits counting from the left of a solution. Their optimal solutions are both $11\ldots1$ (briefly denoted as $1^n$).

**Definition 1 (OneMax).** *The OneMax Problem of size $n$ is to find an $n$ bits binary string $x^*$ that maximizes $f(x) = \sum_{i=1}^{n} x_i$.*

**Definition 2 (LeadingOnes).** *The LeadingOnes Problem of size $n$ is to find an $n$ bits binary string $x^*$ that maximizes $f(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$.*

### 2.2 Bit-Wise Noise

There are mainly two kinds of noise models: prior and posterior [6,8]. The prior noise comes from the variation on a solution, while the posterior noise comes from the variation on the fitness of a solution. Previous theoretical analyses involving prior noise [2,3,6,11,12] often focused on one-bit noise, which flips a random bit of a solution before evaluation with probability $p$. Qian et al. [13] recently considered a natural extension of one-bit noise, i.e., the bit-wise noise model. As presented in Definition 3, it happens with probability $p$, and independently flips each bit of a solution with probability $q$ before evaluation. However, the analyses in [13] considered bit-wise noise with one parameter fixed, i.e., $p = 1$ or $q = 1/n$. In this paper, we will analyze the general bit-wise noise model.

**Definition 3 (Bit-wise Noise** [13]**).** *Given $p, q \in [0,1]$, let $f^{\mathrm{n}}(x)$ and $f(x)$ denote the noisy and true fitness of a solution $x \in \{0,1\}^n$, respectively, then*

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

*where $x'$ is generated by independently flipping each bit of $x$ with probability $q$.*

**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 some termination condition is met
3:     $x' :=$ flip each bit of $x$ with probability $1/n$.
4:     if $f^{\mathrm{n}}(x') \geq f^{\mathrm{n}}(x)$ then $x := x'$.
```

### 2.3 (1+1)-EA

This paper considers the (1+1)-EA, which is a benchmark EA widely used in theoretical analyses. For noisy optimization, only a noisy fitness value $f^{\mathrm{n}}(x)$ instead of the exact one $f(x)$ can be accessed, and thus the condition in line 4 of Algorithm 1 is "if $f^{\mathrm{n}}(x') \geq f^{\mathrm{n}}(x)$" instead of "if $f(x') \geq f(x)$". Note that the reevaluation strategy is used as in [3,6,11,12]. That is, besides evaluating $f^{\mathrm{n}}(x')$, $f^{\mathrm{n}}(x)$ will be reevaluated in each iteration of the (1+1)-EA. The running time 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 [3,6,11,12].

### 2.4 Analysis Tools

The process of the (1+1)-EA solving OneMax or LeadingOnes under noise can be modeled as a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$. We only need to take the solution space $\{0,1\}^n$ as the chain's state space (i.e., $\xi_t \in \mathcal{X} = \{0,1\}^n$), and take the optimal solution $1^n$ as the chain's target state (i.e., $\mathcal{X}^* = \{1^n\}$). Given a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ and $\xi_{\hat{t}} = x$, we define its *first hitting time* as $\tau = \min\{t \mid \xi_{\hat{t}+t} \in \mathcal{X}^*, t \geq 0\}$. The mathematical expectation of $\tau$, $\mathbb{E}[\![\tau \mid \xi_{\hat{t}} = x]\!] = \sum_{i=0}^{+\infty} i \cdot \mathrm{P}(\tau = i \mid \xi_{\hat{t}} = x)$, is called the *expected first hitting time* (EFHT) starting from $\xi_{\hat{t}} = x$. If $\xi_0$ is drawn from a distribution $\pi_0$, $\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!] = \sum_{x \in \mathcal{X}} \pi_0(x) \mathbb{E}[\![\tau \mid \xi_0 = x]\!]$ is called the EFHT of the chain over the initial distribution $\pi_0$. Thus, the expected running time of the (1+1)-EA starting from $\xi_0 \sim \pi_0$ is $1 + 2 \cdot \mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!]$, where the term 1 corresponds to evaluating the initial solution, and the factor 2 corresponds to evaluating the offspring solution $x'$ and reevaluating the parent solution $x$ in each iteration. Note that we consider the expected running time of the (1+1)-EA starting from a uniform initial distribution in this paper.

In the following, we give two drift theorems that will be used to derive upper and lower bounds on the EFHT of Markov chains in the paper.

**Theorem 1 (Additive Drift [7]).** *Given a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ and a distance function $V(x)$ with $V(x \in \mathcal{X}^*) = 0$ and $V(x \notin \mathcal{X}^*) > 0$, if for any $t \geq 0$ and any $\xi_t$ 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 $\mathbb{E}[\![\tau \mid \xi_0]\!] \leq V(\xi_0)/c$.*

**Theorem 2 (Negative Drift with Self-loops [15]).** *Let $X_t$, $t \geq 0$, be real-valued random variables describing a stochastic process. 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 $t \geq 0$:*

(1) $\forall a < i < b : \mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \leq -\epsilon \cdot \mathrm{P}(X_{t+1} \neq i \mid X_t = i)$,

(2) $\forall i > a, j \in \mathbb{N}^+ : \mathrm{P}(|X_{t+1} - X_t| \geq j \mid X_t = i) \leq \dfrac{r(l)}{(1+\delta)^j} \cdot \mathrm{P}(X_{t+1} \neq i \mid X_t = i)$.

*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 $\mathrm{P}(T \leq 2^{cl/r(l)}) = 2^{-\Omega(l/r(l))}$.*

## 3 The OneMax Problem

In this section, we analyze the expected running time of the (1+1)-EA on OneMax under bit-wise noise $(p,q)$. We prove in Theorems 3 and 4 that the expected running time is polynomial when $p = O(\log n/n) \vee pq = O(\log n/n^2)$; otherwise, it is super-polynomial. The results generalize that in [6,13], which only considered the case where $p = 1$ or $q = 1/n$.

Theorem 3 is proved by applying Lemma 1, which gives an upper bound on the running time of the (1+1)-EA solving noisy OneMax. Let $x^k$ denote any solution with $k$ 1-bits, and $f^{\mathrm{n}}(x^k)$ denote its noisy objective value, which is a random variable. Lemma 1 intuitively means that if the probability of recognizing the true better solution by noisy evaluation is large enough (i.e., Eq. (1)), the running time can be upper bounded. Note that in their original theorem (i.e., Theorem 5 in [6]), it requires that Eq. (1) holds with only $j = k$, but their proof actually also requires the property, i.e., $\forall j < k < n$, $\mathrm{P}(f^{\mathrm{n}}(x^k) < f^{\mathrm{n}}(x^{k+1})) \leq \mathrm{P}(f^{\mathrm{n}}(x^j) < f^{\mathrm{n}}(x^{k+1}))$. We have combined these two conditions in Lemma 1 by requiring Eq. (1) to hold with any $j \leq k$ instead of only $j = k$.

**Lemma 1 ([6]).** *If there is a constant $0 < c \leq \frac{1}{15}$ and some $2 < l \leq \frac{n}{2}$ such that*

$$\begin{aligned} &\forall j \leq k < n : \mathrm{P}(f^{\mathrm{n}}(x^j) < f^{\mathrm{n}}(x^{k+1})) \geq 1 - l/n; \\ &\forall j \leq k < n - l : \mathrm{P}(f^{\mathrm{n}}(x^j) < f^{\mathrm{n}}(x^{k+1})) \geq 1 - c(n-k)/n, \end{aligned} \tag{1}$$

*then the (1+1)-EA optimizes $f$ in expectation in $O(n \log n) + n2^{O(l)}$ iterations.*

**Theorem 3.** *For the (1+1)-EA on OneMax under bit-wise noise $(p,q)$, the expected running time is polynomial if $p = O(\log n/n) \vee pq = O(\log n/n^2)$.*

*Proof.* We use Lemma 1 to prove it. We analyze the probability $\mathrm{P}(f^{\mathrm{n}}(x^j) \geq f^{\mathrm{n}}(x^{k+1}))$ for any $j \leq k < n$. We consider two cases.

(1) $p = O(\log n/n)$. For some positive constant $c_1$, assume that $p \le c_1 \log n/n$. It is easy to see that $f^{\rm n}(x^j) \ge f^{\rm n}(x^{k+1})$ implies that the fitness evaluation of $x^j$ or $x^{k+1}$ is affected by noise, whose probability is at most $2p$. Thus, we have $\mathrm{P}(f^{\rm n}(x^j) \ge f^{\rm n}(x^{k+1})) \le 2p \le 2c_1 \log n/n$.

(2) $pq = O(\log n/n^2)$. For some positive constant $c_2$, assume that $pq \le c_2 \log n/n^2$. Note that $f^{\rm n}(x^j) \ge f^{\rm n}(x^{k+1})$ implies that at least one bit of $x^j$ or $x^{k+1}$ is flipped by noise, whose probability is at most $2p(1-(1-q)^n) \le 2pqn \le 2c_2 \log n/n$.

Combining the above two cases leads to $\mathrm{P}(f^{\rm n}(x^j) \ge f^{\rm n}(x^{k+1})) \le \frac{2\max\{c_1,c_2\}\log n}{n}$. Let $l = 30\max\{c_1,c_2\}\log n$ and $c = 1/15$. Then, $\mathrm{P}(f^{\rm n}(x^j) \ge f^{\rm n}(x^{k+1})) \le c \cdot \frac{l}{n}$, and it is easy to verify that the condition of Lemma 1 holds. Thus, by Lemma 1, the expected number of iterations is $O(n\log n) + n2^{O(\log n)}$, which implies that the expected running time is polynomial. □

Theorem 4 is proved by applying Lemma 2, which gives a lower bound on the running time of the (1+1)-EA solving noisy OneMax when the probability of making a right comparison due to noise is not large enough (i.e., Eq. (2)).

**Lemma 2** ([6]). *If there is a constant $c \ge 16$ and some $l \le n/4$ such that*

$$\forall n-l \le k < n : \mathrm{P}(f^{\rm n}(x^k) < f^{\rm n}(x^{k+1})) \le 1 - c(n-k)/n, \tag{2}$$

*then the (1+1)-EA optimizes $f$ in $2^{\Omega(l)}$ iterations with a high probability.*

**Theorem 4.** *For the (1+1)-EA on OneMax under bit-wise noise $(p,q)$, the expected running time is super-polynomial if $p = \omega(\log n/n) \wedge pq = \omega(\log n/n^2)$.*

*Proof.* We use Lemma 2 to prove it. We are to show that $\forall\, 3n/4 \le k < n$, $\mathrm{P}(f^{\rm n}(x^k) \ge f^{\rm n}(x^{k+1})) = \omega(\log n/n)$ by considering two cases of $p$.

(1) $p = \omega(\log n/n) \cap o(1)$. To make $f^{\rm n}(x^k) \ge f^{\rm n}(x^{k+1})$, it is sufficient that $f^{\rm n}(x^k) = k$ and $f^{\rm n}(x^{k+1}) \le k$. The former event happens with probability at least $1-p$, since it is sufficient that the noise doesn't happen. Thus, we have

$$\mathrm{P}(f^{\rm n}(x^k) \ge f^{\rm n}(x^{k+1})) \ge (1-p) \cdot \mathrm{P}(f^{\rm n}(x^{k+1}) \le k). \tag{3}$$

Then, we analyze $\mathrm{P}(f^{\rm n}(x^{k+1}) \le k)$ by further considering two subcases.

(1a) $q \le 1/n$. To make $f^{\rm n}(x^{k+1}) \le k$, it is sufficient that exactly one 1-bit is flipped by noise. Thus,

$$\mathrm{P}(f^{\rm n}(x^{k+1}) \le k) \ge p \cdot (k+1)q(1-q)^{n-1} \ge \omega(\log n/n) \cdot (1/e) = \omega(\log n/n),$$

where the second inequality is by $pq = \omega(\log n/n^2)$, $k \ge 3n/4$ and $q \le 1/n$.

(1b) $q > 1/n$. Let $Y$ denote a random variable such that $\mathrm{P}(Y=0) = 1-q$ and $\mathrm{P}(Y=1) = q$. Let $Y_1, Y_2, ..., Y_n$ denote random variables which are independent and have the same distribution as $Y$. Then, under the condition that the noise happens in evaluating $x^{k+1}$, we have

$$f^{\rm n}(x^{k+1}) = \sum_{j=1}^{k+1}(1-Y_j) + \sum_{j=k+2}^{n} Y_j = k+1 - \sum_{j=1}^{2k-n+2} Y_j - \sum_{j=2k-n+3}^{k+1} Y_j + \sum_{j=k+2}^{n} Y_j.$$

Note that $\sum_{j=k+2}^{n} Y_j - \sum_{j=2k-n+3}^{k+1} Y_j$ is the difference between the sum of the same number of $Y_j$, thus $\mathrm{P}(\sum_{j=k+2}^{n} Y_j - \sum_{j=2k-n+3}^{k+1} Y_j \leq 0) \geq 1/2$ due to symmetry. Then, we have

$$\mathrm{P}(f^{\mathrm{n}}(x^{k+1}) \leq k) \geq \mathrm{P}(\text{the noise happens}) \cdot \mathrm{P}\left(k+1-\sum_{j=1}^{2k-n+2} Y_j \leq k\right) \cdot \frac{1}{2}$$
$$= \frac{p}{2} \cdot (1-(1-q)^{2k+2-n}) \geq \frac{p}{2} \cdot (1-(1-q)^{n/2}) \geq \frac{p}{2} \cdot (1-(1/e)^{1/2}) = \omega(\log n/n),$$

where the second inequality is by $k \geq 3n/4$ and the last is by $q > 1/n$.

Combining subcases (1a) and (1b) leads to $\mathrm{P}(f^{\mathrm{n}}(x^{k+1}) \leq k) = \omega(\log n/n)$. Thus, according to Eq. (3) and $p = o(1)$, it holds that for $3n/4 \leq k < n$, $\mathrm{P}(f^{\mathrm{n}}(x^k) \geq f^{\mathrm{n}}(x^{k+1})) = \omega(\log n/n)$.

(2) $p = \Omega(1)$. Since $pq = \omega(\log n/n^2)$, it must hold that $q = \omega(\log n/n^2)$. We consider three subcases.

(2a) $q = \omega(\log n/n^2) \cap O(1/n)$. We have $\mathrm{P}(f^{\mathrm{n}}(x^k) \geq f^{\mathrm{n}}(x^{k+1})) \geq \mathrm{P}(f^{\mathrm{n}}(x^k) = k) \cdot \mathrm{P}(f^{\mathrm{n}}(x^{k+1}) = k)$. To make $f^{\mathrm{n}}(x^k) = k$, it is sufficient that the noise happens but no bit is flipped by noise. To make $f^{\mathrm{n}}(x^{k+1}) = k$, it is sufficient that exactly one 1-bit is flipped by noise. Thus,

$$\mathrm{P}(f^{\mathrm{n}}(x^k) \geq f^{\mathrm{n}}(x^{k+1})) \geq p(1-q)^n \cdot p(k+1)q(1-q)^{n-1} = \omega(\log n/n),$$

where the equality holds since $p = \Omega(1)$, $q = \omega(\log n/n^2) \cap O(1/n)$ and $k \geq 3n/4$.

(2b) $q = \omega(1/n) \cap O(\log n/n)$. We conduct the following analysis under the condition that both noise happens in evaluating $x^k$ and $x^{k+1}$, whose probability is $p^2 = \Omega(1)$. We divide $x^k$ into two parts: $y^k$ and $z^k$, where $y^k$ is a string with $(\log n - 1)$ 1-bits and one 0-bit, and $z^k$ is a string with $(k - \log n + 1)$ 1-bits and $(n-k-1)$ 0-bits. We also divide $x^{k+1}$ into two parts: $y^{k+1}$ and $z^{k+1}$, where $y^{k+1}$ is a string with $(\log n)$ 1-bits, and $z^{k+1}$ is a string with $(k+1-\log n)$ 1-bits and $(n-k-1)$ 0-bits. Let $mut(x)$ denote the string generated by flipping each bit of $x$ with probability $q$, and let $|x|_1$ denote the number of 1-bits of a string $x$. Thus, we have $f^{\mathrm{n}}(x^k) = |mut(y^k)|_1 + |mut(z^k)|_1$ and $f^{\mathrm{n}}(x^{k+1}) = |mut(y^{k+1})|_1 + |mut(z^{k+1})|_1$. To make $f^{\mathrm{n}}(x^k) \geq f^{\mathrm{n}}(x^{k+1})$, it is sufficient that $|mut(y^k)|_1 \geq \log n - 1$, $|mut(y^{k+1})|_1 = \log n - 1$ and $|mut(z^k)|_1 \geq |mut(z^{k+1})|_1$. Note that $\mathrm{P}(|mut(y^k)|_1 \geq \log n - 1) \geq (1-q)^{\log n - 1}$ since it is sufficient that all the 1-bits of $y^k$ are not flipped; $\mathrm{P}(|mut(y^{k+1})|_1 = \log n - 1) = \log n \cdot q(1-q)^{\log n - 1}$ since exactly one 1-bit needs to be flipped. For $z^k$ and $z^{k+1}$, they are two strings with the same number of 1-bits and 0-bits, and thus $\mathrm{P}(|mut(z^k)|_1 \geq |mut(z^{k+1})|_1) \geq 1/2$ due to symmetry. Then, we get

$$\mathrm{P}(f^{\mathrm{n}}(x^k) \geq f^{\mathrm{n}}(x^{k+1})) \geq p^2 \cdot (1-q)^{\log n - 1} \cdot \log n \cdot q(1-q)^{\log n - 1} \cdot (1/2)$$
$$\geq \omega(\log n/n) \cdot (1 - 2\log n \cdot q) \geq \omega(\log n/n),$$

where the second inequality is by $q = \omega(1/n)$ and Bernoulli's inequality, and the last is by $q = O(\log n/n)$.

(2c) $q = \omega(\log n/n)$. The analysis is similar to that of case (2b), except the division of $x^k$ and $x^{k+1}$. Here, $y^k$ is just a 0-bit, $y^{k+1}$ is just a 1-bit, and $z^k, z^{k+1}$ are two strings with $k$ 1-bits and $(n-k-1)$ 0-bits. We similarly get

$$\mathrm{P}(f^{\mathrm{n}}(x^k) \geq f^{\mathrm{n}}(x^{k+1})) \geq p^2 \cdot q \cdot (1/2) = \omega(\log n/n).$$

Combining cases (1) and (2) shows that $\forall\, 3n/4 \leq k < n$, $\mathrm{P}(f^{\mathrm{n}}(x^k) \geq f^{\mathrm{n}}(x^{k+1})) = \omega(\log n/n)$. We set the parameters in Lemma 2 as $c = 16$ and $l = b \log n$, where $b$ is any positive constant. Thus, for any $n - l \leq k < n$, $\mathrm{P}(f^{\mathrm{n}}(x^k) < f^{\mathrm{n}}(x^{k+1})) = 1 - \omega(\log n/n) \leq 1 - c(n-k)/n$. By Lemma 2, the expected number of iterations is $2^{\Omega(l)} = n^{\Omega(b)}$. Since $b$ can be any positive constant, the expected running time is super-polynomial. □

## 4 The LeadingOnes Problem

In this section, we consider the (1+1)-EA solving LeadingOnes under bit-wise noise $(p, q)$. We prove in Theorems 5 and 6 that the expected running time is polynomial if $p = O(\log n/n^2) \vee pq = O(\log n/n^3)$, and super-polynomial if $p = \omega(\log n/n) \wedge pq = \omega(\log n/n^2)$. The results generalize that in [13], where $p$ is fixed to 1 or $q$ is fixed to $1/n$.

Theorem 5 is proved by applying the additive drift theorem (i.e., Theorem 1). We will use $\mathrm{LO}(x)$ to denote the number of leading 1-bits of a solution $x$.

**Theorem 5.** *For the (1+1)-EA on LeadingOnes under bit-wise noise $(p, q)$, the expected running time is polynomial if $p = O(\log n/n^2) \vee pq = O(\log n/n^3)$.*

*Proof.* We use Theorem 1 to prove it. For some positive constant $b$, suppose that $p \leq b \log n/n^2$ or $pq \leq b \log n/n^3$. We construct a distance function as follows:

$$V(x) = (1 + c/n)^n - (1 + c/n)^{\mathrm{LO}(x)},$$

where $c = 12b \log n + 1$. It is easy to verify that $V(x) = 0$ iff $x \in \mathcal{X}^* = \{1^n\}$.

Then, we investigate $\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!]$ for any $x$ with $\mathrm{LO}(x) = i < n$. Let $\mathrm{P}_{\mathrm{mut}}(x, x')$ denote the probability that $x'$ is generated from $x$ by mutation, and let $\mathrm{P}_{\mathrm{acc}}(x, x')$ denote the probability that the offspring solution $x'$ is accepted by comparing with $x$, i.e., $\mathrm{P}_{\mathrm{acc}}(x, x') = \mathrm{P}(f^{\mathrm{n}}(x') \geq f^{\mathrm{n}}(x))$. We divide the drift into two parts: positive $\mathrm{E}^+$ and negative $\mathrm{E}^-$. That is,

$$\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] = \mathrm{E}^+ - \mathrm{E}^-,$$

where

$$\mathrm{E}^+ = \textstyle\sum_{x':\mathrm{LO}(x')>i} \mathrm{P}_{\mathrm{mut}}(x, x') \cdot \mathrm{P}_{\mathrm{acc}}(x, x') \cdot (V(x) - V(x')),$$
$$\mathrm{E}^- = \textstyle\sum_{x':\mathrm{LO}(x')<i} \mathrm{P}_{\mathrm{mut}}(x, x') \cdot \mathrm{P}_{\mathrm{acc}}(x, x') \cdot (V(x') - V(x)).$$

Note that $V(x) > V(x')$ iff $\mathrm{LO}(x') > \mathrm{LO}(x) = i$, since the distance function $V$ decreases with the number of leading 1-bits.

We first analyze the positive drift $\mathrm{E}^+$. For any $x'$ with $\mathrm{LO}(x') > i$,

$$V(x) - V(x') = (1 + c/n)^{\mathrm{LO}(x')} - (1 + c/n)^i \geq (1 + c/n)^i \cdot c/n. \qquad (4)$$

To generate $x'$ with $\mathrm{LO}(x') > i$ by mutating $x$, it needs to flip the $(i+1)$-th bit (which must be 0) of $x$ and keep the $i$ leading 1-bits unchanged. Thus, we have

$$\textstyle\sum_{x':\mathrm{LO}(x')>i} \mathrm{P}_{\mathrm{mut}}(x, x') = \mathrm{P}(\mathrm{LO}(x') > i) = (1-1/n)^i\,(1/n) \geq 1/en. \tag{5}$$

To analyze $\mathrm{P}_{\mathrm{acc}}(x, x')$ for any $x'$ with $\mathrm{LO}(x') > i$, we consider the opposite event that $x'$ is rejected, i.e., $f^{\mathrm{n}}(x) > f^{\mathrm{n}}(x')$, which implies that $f^{\mathrm{n}}(x) \geq i+1$ or $f^{\mathrm{n}}(x') \leq i-1$. By the union bound, $\mathrm{P}(f^{\mathrm{n}}(x) > f^{\mathrm{n}}(x')) \leq \mathrm{P}(f^{\mathrm{n}}(x) \geq i+1) + \mathrm{P}(f^{\mathrm{n}}(x') \leq i-1) = pq(1-q)^i + p(1-(1-q)^i) = p - p(1-q)^{i+1}$, where the first equality is because $f^{\mathrm{n}}(x) \geq i+1$ iff the $(i+1)$-th bit of $x$ is flipped by noise while the $i$ leading 1-bits are not flipped; $f^{\mathrm{n}}(x') \leq i-1$ iff at least one of the $i$ leading 1-bits of $x'$ is flipped by noise. Then, we get

$$\begin{aligned}\mathrm{P}_{\mathrm{acc}}(x, x') &= 1 - \mathrm{P}(f^{\mathrm{n}}(x) > f^{\mathrm{n}}(x')) \geq 1 - p + p(1-q)^{i+1} \\ &\geq \max\{1-p, 1-pq(i+1)\} \geq 1 - b\log n/n^2 \geq 1/2,\end{aligned} \tag{6}$$

where the second inequality is by Bernoulli's inequality, the third inequality is by $p \leq b\log n/n^2$ or $pq \leq b\log n/n^3$, and the last holds with sufficiently large $n$. By applying Eqs. (4), (5) and (6) to $\mathrm{E}^+$, we get

$$\mathrm{E}^+ \geq (1/en)\cdot(1/2)\cdot(1+c/n)^i\cdot(c/n) \geq (1+c/n)^i\cdot(c/6n^2).$$

We then analyze the negative drift $\mathrm{E}^-$. For any $x'$ with $\mathrm{LO}(x') < i$, we have

$$V(x') - V(x) = (1+c/n)^i - (1+c/n)^{\mathrm{LO}(x')} \leq (1+c/n)^i - 1. \tag{7}$$

To analyze $\mathrm{P}_{\mathrm{acc}}(x, x')$ for any $x'$ with $\mathrm{LO}(x') < i$, we consider $f^{\mathrm{n}}(x) \leq f^{\mathrm{n}}(x')$, which implies that at least one bit of $x$ or $x'$ is flipped by noise. By the union bound, we have $\mathrm{P}_{\mathrm{acc}}(x, x') \leq 2p\,(1-(1-q)^n)$. Note that $1-(1-q)^n \leq \min\{qn, 1\}$ and $p \leq b\log n/n^2$ or $pq \leq b\log n/n^3$, we have

$$\mathrm{P}_{\mathrm{acc}}(x, x') \leq 2p\cdot\min\{nq, 1\} = \min\{2npq, 2p\} \leq 2b\log n/n^2. \tag{8}$$

By applying Eqs. (7) and (8) to $\mathrm{E}^-$, we get

$$\mathrm{E}^- \leq \textstyle\sum_{x':\mathrm{LO}(x')<i} \mathrm{P}_{\mathrm{mut}}(x, x')\cdot(2b\log n/n^2)\cdot(1+c/n)^i \leq (1+c/n)^i\,(2b\log n/n^2).$$

By subtracting $\mathrm{E}^-$ from $\mathrm{E}^+$, we get

$$\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] \geq (1+c/n)^i\left(c/6n^2 - 2b\log n/n^2\right) \geq 1/6n^2,$$

where the last inequality is by $c = 12b\log n + 1$. Note that $V(x) \leq (1+\frac{c}{n})^n \leq e^c = en^{12b}$. By Theorem 1, we have $\mathbb{E}[\![\tau \mid \xi_0]\!] \leq 6n^2 \cdot en^{12b} = O\left(n^{12b+2}\right)$, thus the expected running time is polynomial. □

Next, we use the negative drift with self-loops theorem (i.e., Theorem 2) to prove a super-polynomial lower bound for $p = \omega(\log n/n) \wedge pq = \omega(\log n/n^2)$.

**Theorem 6.** *For the (1+1)-EA on LeadingOnes under bit-wise noise $(p, q)$, the expected running time is super-polynomial if $p = \omega(\log n/n) \wedge pq = \omega(\log n/n^2)$.*

*Proof.* We use Theorem 2 to prove it. Let $X_t = |x|_0$ be the number of 0-bits of the solution $x$ after $t$ iterations of the (1+1)-EA. Let $c$ by any positive constant. We consider the interval $[0, c\log n]$, i.e., $a = 0$ and $b = c\log n$ in Theorem 2.

Then, we analyze $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!]$ for $1 \le i < c\log n$. As in the proof of Theorem 5, we also divide the drift into positive $\mathrm{E}^+$ and negative $\mathrm{E}^-$:

$$\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] = \mathrm{E}^+ - \mathrm{E}^-,$$

where

$$\mathrm{E}^+ = \textstyle\sum_{x':|x'|_0<i} \mathrm{P}_{\mathrm{mut}}(x, x') \cdot \mathrm{P}_{\mathrm{acc}}(x, x') \cdot (i - |x'|_0),$$
$$\mathrm{E}^- = \textstyle\sum_{x':|x'|_0>i} \mathrm{P}_{\mathrm{mut}}(x, x') \cdot \mathrm{P}_{\mathrm{acc}}(x, x') \cdot (|x'|_0 - i).$$

Note that we still use $\mathrm{P}_{\mathrm{mut}}(x, x')$ and $\mathrm{P}_{\mathrm{acc}}(x, x')$ to denote the probability that the offspring solution $x'$ is generated and accepted, respectively.

To analyze $\mathrm{E}^+$, we use a trivial upper bound 1 for $\mathrm{P}_{\mathrm{acc}}(x, x')$. Then, we have

$$\mathrm{E}^+ \le \textstyle\sum_{x':|x'|_0<i} \mathrm{P}_{\mathrm{mut}}(x, x')(i - |x'|_0) \le i/n,$$

where the second inequality is directly from the proof of Theorem 4.2 in [13].

For the negative drift $\mathrm{E}^-$, we need to consider the increase of the number of 0-bits. We analyze the $n-i$ cases where only one 1-bit is flipped (i.e., $|x'|_0 = i+1$), which happens with probability $\frac{1}{n}(1 - \frac{1}{n})^{n-1}$. Assume that $\mathrm{LO}(x) = k \le n - i$. To analyze $\mathrm{P}_{\mathrm{acc}}(x, x') = \mathrm{P}(f^{\mathrm{n}}(x') \ge f^{\mathrm{n}}(x))$, we consider two cases.
(1) The $j$-th (where $1 \le j \le k$) leading 1-bit is flipped. To make $f^{\mathrm{n}}(x') \ge f^{\mathrm{n}}(x)$, we consider the $j$ cases where $f^{\mathrm{n}}(x) = l$ and $f^{\mathrm{n}}(x') \ge l$ for $0 \le l \le j-1$. Note that $\mathrm{P}(f^{\mathrm{n}}(x) = l) = p(1-q)^l q$ and $\mathrm{P}(f^{\mathrm{n}}(x') \ge l) = 1 - p + p(1-q)^l$. Thus,

$$\mathrm{P}_{\mathrm{acc}}(x, x') \ge \textstyle\sum_{l=0}^{j-1} p(1-q)^l q \cdot \left(1 - p + p(1-q)^l\right).$$

If $p = o(1)$, $1-p+p(1-q)^l \ge \Omega(1)$; otherwise, $1-p+p(1-q)^l \ge \Omega(1) \cdot (1-q)^l$. Thus,

$$\mathrm{P}_{\mathrm{acc}}(x, x') \ge \Omega(1) \cdot pq \textstyle\sum_{l=0}^{j-1} (1-q)^{2l} \ge \Omega(1) \cdot p \cdot \left(1 - (1-q)^{2j}\right).$$

(2) One of the $(n-i-k)$ non-leading 1-bits is flipped, i.e., $\mathrm{LO}(x') = \mathrm{LO}(x) = k$. To make $f^{\mathrm{n}}(x') \ge f^{\mathrm{n}}(x)$, we consider the $k+1$ cases where $f^{\mathrm{n}}(x) = l$ and $f^{\mathrm{n}}(x') \ge l$ for $0 \le l \le k$. Thus, we have

$$\begin{aligned}\mathrm{P}_{\mathrm{acc}}(x, x') \ge & \textstyle\sum_{l=0}^{k-1} p(1-q)^l q \cdot \left(1 - p + p(1-q)^l\right) \\ & + \left(1 - p + p(1-q)^{k+1}\right) \cdot \left(1 - p + p(1-q)^k\right).\end{aligned}$$

If $p = o(1)$, obviously $\mathrm{P}_{\mathrm{acc}}(x, x') = \Omega(1)$. If $p = \Omega(1)$, we can derive that $\mathrm{P}_{\mathrm{acc}}(x, x') \ge \Omega(1) \cdot \left(1-(1-q)^{2k}\right) + \left(\Omega(1)(1-q)^{k+1}\right)^2$, and then further consider two cases for $q$. If $q=\Omega(1)$, $\mathrm{P}_{\mathrm{acc}}(x,x')\geq\Omega(1)\cdot(1-(1-q)^{2k})\geq\Omega(1)\cdot(1-(1-q))=\Omega(1)$. If $q=o(1)$, $\mathrm{P}_{\mathrm{acc}}(x,x')\geq\Omega(1)(1-(1-q)^{2k})+\Omega(1)(1-q)^{2k}=\Omega(1)$. Thus,

$$\mathrm{P}_{\mathrm{acc}}(x,x') = \Omega(1).$$

Combining cases (1) and (2), we get

$$\mathrm{E}^- \geq (1/n)\,(1-1/n)^{n-1}\cdot\Big(\Omega(1)\cdot p\textstyle\sum_{j=1}^{k}(1-(1-q)^{2j}) + (n-i-k)\cdot\Omega(1)\Big).$$

If $(1-q)^{2j} < 1/2, 1-(1-q)^{2j} > 1/2$; otherwise, $1-(1-q)^{2j} = (1-q)^{2j}((1+\frac{q}{1-q})^{2j}-1) \geq (1-q)^{2j}\frac{2qj}{1-q} \geq (1-q)^{2j}\cdot 2qj \geq qj$ . Thus,

$$\mathrm{E}^- \geq \Omega(1/n)\cdot\Big(p\textstyle\sum_{j=1}^{k}\min\{1/2,qj\}+n-i-k\Big).$$

By subtracting $\mathrm{E}^-$ from $\mathrm{E}^+$, we get

$$\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] \leq i/n - \Omega(1/n)\cdot\Big(p\textstyle\sum_{j=1}^{k}\min\{1/2,qj\}+n-i-k\Big).$$

To investigate condition (1) of Theorem 2, we need to derive an upper bound on $\mathrm{P}(X_{t+1}\neq i \mid X_t = i)$. To make $X_{t+1}\neq i$, it is necessary that at least one bit of $x$ is flipped and $x'$ is accepted. We consider two cases: (1) at least one of the $k$ leading 1-bits of $x$ is flipped; (2) the $k$ leading 1-bits of $x$ are not flipped and at least one of the last $n-k$ bits is flipped. For case (1), $\mathrm{P}_{\mathrm{acc}}(x,x')\leq\min\{2npq,2p\}$ by Eq. (8). For case (2), $\sum_{x'}\mathrm{P}_{\mathrm{mut}}(x,x')\leq\frac{n-k}{n}$. Thus, we get

$$\mathrm{P}(X_{t+1}\neq i \mid X_t = i) \leq \min\{2npq,2p\} + (n-k)/n.$$

Now we compare $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!]$ with $\mathrm{P}(X_{t+1}\neq i \mid X_t = i)$.
(1) $k < n/2$. We have

$$\begin{aligned}\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] &\leq i/n - \Omega(1/n)\cdot(n-i-k)\\ &= -\Omega(1) \leq -\Omega(1)\cdot\mathrm{P}(X_{t+1}\neq i \mid X_t = i),\end{aligned}$$

where the equality is by $i < c\log n$ and $k < n/2$.
(2) $k \geq n/2$. We first investigate $\sum_{j=1}^{k}\min\{1/2,qj\}$. If $q = o(1/n)$, we have $\sum_{j=1}^{k}\min\{1/2,qj\} \geq qk^2/2 = \Omega(qn^2)$. If $q=\Omega(1/n)$, $\sum_{j=1}^{k}\min\{1/2,qj\} = \Omega(n)$. Thus, we have $\sum_{j=1}^{k}\min\{1/2,qj\}\geq\Omega(1)\cdot\min\{qn^2,n\}$. Then we get

$$\begin{aligned}\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] &\leq i/n - \Omega(1/n)\cdot\big(p\cdot\min\{qn^2,n\}+n-i-k\big)\\ &= i/n - \Omega(1)\cdot(\min\{pqn,p\} - i/n + (n-k)/n)\,.\end{aligned}$$

Note that $pq=\omega(\log n/n^2)$ and $p=\omega(\log n/n)$, thus $\min\{pqn,p\}=\omega(\log n/n)$. Furthermore, $i < c\log n$. Thus, we get

$$\begin{aligned}\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!] &\leq -\Omega(1)\cdot(\min\{pqn,p\} + (n-k)/n)\\ &\leq -\Omega(1)\cdot\mathrm{P}(X_{t+1}\neq i \mid X_t = i).\end{aligned}$$

Combining the above two cases implies that condition (1) of Theorem 2 holds.

To investigate condition (2) of Theorem 2, we need to derive a lower bound on $\mathrm{P}(X_{t+1} \neq i \mid X_t = i)$. We consider the $n$ cases where only one bit is flipped. We can directly follow the analysis for $\mathrm{E}^-$ to derive that

$$\mathrm{P}(X_{t+1} \neq i \mid X_t = i) \geq \Omega(1/n) \cdot \big(p\textstyle\sum_{j=1}^{k} \min\{1/2, qj\} + n - k\big).$$

The only difference is that we also consider flipping only one 0-bit, whose analysis is the same as that for flipping only one non-leading 1-bit. To make $|X_{t+1} – X_t| \geq j$, it is necessary that at least $j$ bits of $x$ are flipped and $x'$ is accepted. We consider two cases: (1) at least one of the $k$ leading 1-bits of $x$ is flipped; (2) the $k$ leading 1-bits are not flipped. For case (1), $\sum_{x'} \mathrm{P}_{\mathrm{mut}}(x,x') \leq \frac{k}{n}\binom{n-1}{j-1}\frac{1}{n^{j-1}}$ and $\mathrm{P}_{\mathrm{acc}}(x,x') \leq \min\{2npq, 2p\}$ by Eq. (8). For case (2), $\sum_{x'} \mathrm{P}_{\mathrm{mut}}(x,x') \leq (1-\frac{1}{n})^k\binom{n-k}{j}\frac{1}{n^j}$. Thus,

$$\begin{aligned}\mathrm{P}(|X_{t+1} - X_t| \geq j \mid X_t = i) &\leq \frac{k}{n}\binom{n-1}{j-1}\frac{\min\{2npq, 2p\}}{n^{j-1}} + \left(1-\frac{1}{n}\right)^k \binom{n-k}{j}\frac{1}{n^j} \\ &\leq (k/n) \cdot \min\{2npq, 2p\} \cdot (4/2^j) + ((n-k)/n) \cdot (2/2^j).\end{aligned}$$

By following the way of comparing $\mathbb{E}[\![X_t - X_{t+1} \mid X_t = i]\!]$ with $\mathrm{P}(X_{t+1} \neq i \mid X_t = i)$ in the above analysis, we can derive that

$$\mathrm{P}(|X_{t+1} - X_t| \geq j \mid X_t = i) \leq (O(1)/2^j) \cdot \mathrm{P}(X_{t+1} \neq i \mid X_t = i),$$

i.e., condition (2) of Theorem 2 holds with $\delta = 1$ and $r(l) = O(1)$.

The parameter $l$ in Theorem 2 is $b - a = c \log n$. Thus, the expected running time is $2^{\Omega(c \log n)}$ (where $c > 0$ can be any constant), i.e., super-polynomial. □

## 5 Conclusion

In this paper, we analyze the running time of the (1+1)-EA solving OneMax and LeadingOnes under bit-wise noise $(p, q)$. We derive the ranges of $p, q$ for the running time being polynomial and super-polynomial, respectively. Our results complement previous analyses, which fix $p = 1$ or $q = 1/n$. Note that our analysis on LeadingOnes does not cover all the ranges of $p, q$. That is, the running time is not known for $p = \omega(\log n/n^2) \cap O(\log n/n) \wedge pq = \omega(\log n/n^3)$ and $p = \omega(\log n/n) \wedge pq = \omega(\log n/n^3) \cap O(\log n/n^2)$. This question has been partially addressed in the recent work [16]. We leave the full analysis as our future work.

## References

1. Auger, A., Doerr, B.: Theory of Randomized Search Heuristics: Foundations and Recent Developments. World Scientific, Singapore (2011)
2. Dang, D.C., Lehre, P.K.: Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In: FOGA 2015, Aberystwyth, UK, pp. 62–68 (2015)

3. Droste, S.: Analysis of the (1+1) EA for a noisy OneMax. In: Deb, K. (ed.) GECCO 2004. LNCS, vol. 3102, pp. 1088–1099. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24854-5_107
4. Friedrich, T., Kötzing, T., Krejca, M., Sutton, A.: Robustness of ant colony optimization to noise. Evol. Comput. **24**(2), 237–254 (2016)
5. Friedrich, T., Kötzing, T., Krejca, M., Sutton, A.: The compact genetic algorithm is efficient under extreme gaussian noise. IEEE Trans. Evol. Comput. **21**(3), 477–490 (2017)
6. Gießen, C., Kötzing, T.: Robustness of populations in stochastic environments. Algorithmica **75**(3), 462–489 (2016)
7. He, J., Yao, X.: Drift analysis and average time complexity of evolutionary algorithms. Artif. Intell. **127**(1), 57–85 (2001)
8. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments-a survey. IEEE Trans. Evol. Comput. **9**(3), 303–317 (2005)
9. Neumann, F., Witt, C.: Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16544-3
10. Prügel-Bennett, A., Rowe, J., Shapiro, J.: Run-time analysis of population-based evolutionary algorithm in noisy environments. In: FOGA 2015, Aberystwyth, UK, pp. 69–75 (2015)
11. Qian, C., Yu, Y., Tang, K., Jin, Y., Yao, X., Zhou, Z.H.: On the effectiveness of sampling for evolutionary optimization in noisy environments. Evol. Comput. **26**(2), 237–267 (2018)
12. Qian, C., Yu, Y., Zhou, Z.H.: Analyzing evolutionary optimization in noisy environments. Evol. Comput. **26**(1), 1–41 (2018)
13. Qian, C., Bian, C., Jiang, W., Tang, K.: Running time analysis of the (1+1)-EA for OneMax and LeadingOnes under bit-wise noise. In: GECCO 2017, Berlin, Germany, pp. 1399–1406 (2017)
14. Qian, C., Bian, C., Yu, Y., Tang, K., Yao, X.: Analysis of noisy evolutionary optimization when sampling fails. In: GECCO 2018, Kyoto, Japan, pp. 1507–1514 (2018)
15. Rowe, J.E., Sudholt, D.: The choice of the offspring population size in the (1,$\lambda$) evolutionary algorithm. Theor. Comput. Sci. **545**, 20–38 (2014)
16. Sudholt, D.: On the robustness of evolutionary algorithms to noise: Refined results and an example where noise helps. In: GECCO 2018, Kyoto, Japan, pp. 1523–1530 (2018)