Analyzing Evolutionary Optimization in Noisy Environments

Chao Qian  
National Key Laboratory for Novel Software Technology, Nanjing University,  
Nanjing, 210023, China  
quianc@lamda.nju.edu.cn

Yang Yu  
National Key Laboratory for Novel Software Technology, Nanjing University,  
Nanjing, 210023, China  
yuy@lamda.nju.edu.cn

Zhi-Hua Zhou  
National Key Laboratory for Novel Software Technology, Nanjing University,  
Nanjing, 210023, China  
zhouzh@lamda.nju.edu.cn

doi:10.1162/EVCO_a_00170

Abstract

Many optimization tasks must be handled in noisy environments, where the exact evaluation of a solution cannot be obtained, only a noisy one. For optimization of noisy tasks, evolutionary algorithms (EAs), a type of stochastic metaheuristic search algorithm, have been widely and successfully applied. Previous work mainly focuses on the empirical study and design of EAs for optimization under noisy conditions, while the theoretical understandings are largely insufficient. In this study, we first investigate how noisy fitness can affect the running time of EAs. Two kinds of noise-helpful problems are identified, on which the EAs will run faster with the presence of noise, and thus the noise should not be handled. Second, on a representative noise-harmful problem in which the noise has a strong negative effect, we examine two commonly employed mechanisms dealing with noise in EAs: reevaluation and threshold selection. The analysis discloses that using these two strategies simultaneously is effective for the one-bit noise but ineffective for the asymmetric one-bit noise. Smooth threshold selection is then proposed, which can be proved to be an effective strategy to further improve the noise tolerance ability in the problem. We then complement the theoretical analysis by experiments on both synthetic problems as well as two combinatorial problems, the minimum spanning tree and the maximum matching. The experimental results agree with the theoretical findings and also show that the proposed smooth threshold selection can deal with the noise better.

Keywords

Noisy optimization, evolutionary algorithms, reevaluation, threshold selection, running time, computational complexity.

1 Introduction

Optimization tasks often encounter noisy environments. For example, in industrial design such as VLSI design (Guo et al., 2014), every prototype is evaluated by simulations; therefore, the result of the evaluation may not be perfect due to the simulation error. Also, with machine learning, a prediction model is evaluated only on a limited amount of data (Qian et al., 2015a); therefore, the estimated performance is shifted from the true performance. It is possible that noisy environments change the properties of an
optimization problem; thus traditional optimization techniques may have low efficacy. Meanwhile, evolutionary algorithms (EAs) (Bäck, 1996) have been widely and successfully adopted for noisy optimization tasks (Freitas, 2003; Ma et al., 2006; Chang and Chen, 2006).

EAs are a type of randomized metaheuristic optimization algorithm, inspired by natural phenomena including evolution of species, swarm cooperation, immune systems, and others. EAs typically involve a cycle of three stages: a reproduction stage that produces new solutions based on the currently maintained solutions; an evaluation stage that evaluates the newly generated solutions; and a selection stage that wipes out bad solutions. The concept of using EAs for noisy optimization is that the corresponding natural phenomena have been successfully processed in noisy natural environments, and hence the algorithmic simulations are also likely to be able to handle noise.

On one hand, it is believed that noise makes the optimization harder, and thus handling mechanisms have been proposed to reduce the negative effect of the noise (Fitzpatrick and Grefenstette, 1988; Beyer, 2000; Arnold and Beyer, 2003). Two representative strategies are reevaluation and threshold selection. According to the reevaluation strategy (Jin and Branke, 2005; Goh and Tan, 2007; Doerr et al., 2012a), whenever the fitness (also called the cost or objective value) of a solution is required, EAs make an independent evaluation of the solution regardless of whether the solution has been evaluated before, such that the fitness is smoothed. According to the threshold selection strategy (Markon et al., 2001; Bartz-Beielstein and Markon, 2002; Bartz-Beielstein, 2005a), in the selection stage EAs accept a newly generated solution only if its fitness is larger than the fitness of the old solution by at least a threshold value $\tau$, such that the risk of accepting a bad solution due to noise is reduced.

On the other hand, several empirical observations have shown cases where noise can have a positive impact on the performance of local search (Selman et al., 1994; Hoos and Stützle, 2000; 2005), which indicates that noise does not always have a negative impact.

As these previous studies are mainly empirical, theoretical analysis is needed for a better understanding of evolutionary optimization in noisy environments.

1.1 Related Work

Despite EAs’ wide and successful application, the theoretical analysis of EAs on noisy optimization is rare. Some theoretical results on EAs have emerged (e.g., Neumann and Witt, 2010; Auger and Doerr, 2011), but most of them focus on clean environments. In noisy environments, the optimization is more complex and more randomized; thus the theoretical analysis is difficult.

Only a few theoretical analyses for EAs on noisy optimization have been published. Gutjahr (2003; 2004) first analyzed the ant colony optimization (ACO) algorithm for stochastic combinatorial optimization and proved convergence under mild conditions. Droste (2004) gave a running time analysis of EAs in discrete noisy optimization for the first time. Droste analyzed the $(1+1)$-EA on the OneMax problem under one-bit noise and showed the maximal noise strength $\log(n)/n$ allowing a polynomial running time, where the noise strength is characterized by the noise probability in $[0, 1]$ and $n$ is the problem size. Sudholt and Thyssen (2012) analyzed the running time of a simple ACO for stochastic shortest path problems where edge weights are subject to noise, and showed the ability and limitation of the ACO under various noise models. For the difficulty faced by an ACO under a specific noise model, Doerr et al. (2012a) further showed that the reevaluation strategy can overcome it, that is, avoid being misled.
Table 1: The PNT with respect to one-bit noise of the (1 + 1)-EA using different noise-handling strategies on the OneMax problem.

<table>
<thead>
<tr>
<th>Noise Handling Strategies</th>
<th>PNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>single evaluation</td>
<td>$[0, 1 - \frac{1}{\Theta_1(poly(n))}]$</td>
</tr>
<tr>
<td>single-evaluation and $\tau &gt; 0$</td>
<td>$[0, 0]$</td>
</tr>
<tr>
<td>reevaluation</td>
<td>$[0, \Theta_1(\frac{\log n}{n})]$ (Droste, 2004)</td>
</tr>
<tr>
<td>reevaluation and $\tau = 1$</td>
<td>$[0, 1]$</td>
</tr>
<tr>
<td>reevaluation and $\tau = 2$</td>
<td>$[\frac{1}{\Theta_1(poly(n))}, 1 - \frac{1}{\Theta_1(poly(n))}]$</td>
</tr>
<tr>
<td>reevaluation and $\tau &gt; 2$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

by an exceptionally optimistic evaluation due to noise. Qian et al. (2014) investigated the effectiveness of sampling, a common strategy to reduce the effect of noise. They proved a sufficient condition under which sampling is useless (i.e., sampling increases the running time), and applied it to show that sampling is useless for the (1 + 1)-EA optimizing the OneMax and the Trap problems under additive Gaussian noise.

1.2 Our Contribution

In this article, we study the effect of noise on EAs and investigate the noise-handling mechanisms when noise needs to be accounted for.

The effect of noise on the expected running time of EAs is investigated in Section 3. On deceptive and flat problems, we prove that noise can simplify the optimization (i.e., decrease the expected running time) for EAs. The analysis results support that for some difficult problems, handling the noise is not necessary.

In Section 4.1 the OneMax problem is proved to be negatively affected by noise, and in Section 4.2 two commonly employed noise-handling mechanisms are examined: the reevaluation and the threshold selection strategies. With the (1 + 1)-EA under one-bit noise, the noise-handling mechanisms are evaluated by the polynomial noise tolerance (PNT), which is the range of the noise strength such that the expected running time of the algorithm is polynomial. The wider the PNT is, the better a noise-handling mechanism is. For example, the one-bit noise strength (and thus the PNT) is characterized by the noise probability $p_n \in [0, 1]$. The configurations of the (1 + 1)-EA that we analyzed include without any noise-handling strategy (single-evaluation), single-evaluation with threshold selection (single-evaluation with a value of $\tau$), and reevaluation with threshold selection (reevaluation with a value of $\tau$). Their PNTs are presented in Table 1, where the PNT of the (1 + 1)-EA with reevaluation (but no threshold selection) is directly derived from Droste (2004). The comparison shows the following:

- Reevaluation alone makes the PNT much worse than single-evaluation.
- Threshold selection must be combined with reevaluation; otherwise, the EA could not tolerate any noise strength larger than 0; meanwhile, reevaluation can also be better if used with threshold selection.
- Reevaluation with threshold selection (threshold $= 1$) can improve upon that of single-evaluation.
In Section 4.3 we disclose a weakness of the noise-handling mechanisms: when used with the (1 + 1)-EA solving the OneMax problem under asymmetric one-bit noise, all of them are ineffective (i.e., need exponential running time) when the noise probability reaches 1. The reason for the ineffectiveness of reevaluation with threshold selection is that it has too large a probability of accepting false progress caused by the noise when the threshold $\tau \leq 1$ and too small a probability of accepting true progress when $\tau \geq 2$. Setting $\tau$ between 1 and 2 is useless because of the minimum fitness gap 1 (i.e., a value of $\tau \in (1, 2)$ is equivalent to $\tau = 2$). We then introduce a modification into a threshold selection strategy to turn the original hard threshold into a smooth threshold, which allows a fractional threshold to be effective. We prove that with the smooth threshold selection strategy the PNT can be $[0, 1]$, that is, the (1 + 1)-EA is always a polynomial algorithm on the problem regardless of the noise probability.

Finally, in Section 5, we describe our experiments to verify and complement the theoretical results. We show using two problem classes, the Jump problem, which is a synthetic problem, and the minimum spanning tree problem, which is a common combinatorial problem. We show that the badness of the noise is negatively correlated with the hardness of the problem, which was previously not noticed. Therefore, when the problem is quite hard, the noise can be helpful and thus handling the noise is not necessary. Then we verify that smooth threshold selection can better handle the noise by experiments on the maximum matching problem. Section 6 concludes the article.

2 Preliminaries

2.1 Noisy Optimization

A general optimization problem can be represented as $\arg \max_x f(x)$, where the objective $f$ is also called fitness in the context of evolutionary computation. In real-world optimization tasks, the fitness evaluation for a solution is usually disturbed by noise, and consequently we cannot obtain the exact fitness value but only a noisy one. Let $f^N(x)$ and $f(x)$ denote the noisy and true fitness of a solution $x$, respectively. In this study, we use the following three widely investigated noise models:

- **Additive.** $f^N(x) = f(x) + \delta$, where $\delta$ is uniformly selected from $[\delta_1, \delta_2]$ at random.
- **Multiplicative.** $f^N(x) = f(x) \cdot \delta$, where $\delta$ is uniformly randomly selected from $[\delta_1, \delta_2]$.
- **One-bit.** $f^N(x) = f(x)$ with probability $(1 - p_n) (p_n \in [0, 1])$; otherwise, $f^N(x) = f(x')$, where $x'$ is generated by flipping a uniformly randomly chosen bit of $x \in \{0, 1\}^n$. This noise is for problems where solutions are represented in binary strings.

Additive and multiplicative noise have often been used to analyze the effect of noise (Beyer, 2000; Jin and Branke, 2005). One-bit noise is specifically used for optimizing pseudo-Boolean problems over $[0, 1]^n$ and has been investigated in the first work for analyzing the running time of EAs in noisy optimization (Droste, 2004) and used to understand the role of noise in stochastic local search (Selman et al., 1994; Hoos and Stützle, 1999; Mengshoel, 2008).

Besides these kinds of noise we also consider a variant of one-bit noise called asymmetric one-bit noise (see Definition 1). Inspired by the asymmetric mutation operator...
Analyzing Evolutionary Optimization in Noisy Environments

(Jansen and Sudholt, 2010), asymmetric one-bit noise flips a specific bit position with probability depending on the number of bit positions that take the same value. For the flipping of asymmetric one-bit noise on a solution \( x \in \{0, 1\}^n \), 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 \( \frac{1}{n} \). For both one-bit and asymmetric one-bit noise, \( p_n \) controls the noise strength. In this article, we assume that the parameters of the environment \((p_n, \delta_1, \delta_2)\) do not change over time.

**Definition 1 (Asymmetric One-Bit Noise):** Given a fitness function \( f \) and a solution \( x \in \{0, 1\}^n \), asymmetric one-bit noise with a parameter \( p_n \in [0, 1] \) leads to a noisy fitness value \( f^N(x) \) as \( f^N(x) = f(x) \) with probability \((1 - p_n)\), 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 \( \begin{cases} \text{all bits of } x, & \text{if } |x|_0 = 0 \text{ or } n; \\
0 \text{ bits of } x, & \text{with probability } 1/2; \\
1 \text{ bits of } x, & \text{with probability } 1/2. \\
\end{cases} \)

It is possible that a large noise could make an optimization problem extremely hard for particular algorithms. We are thus interested in the noise strength under which an algorithm could be tolerant to have a polynomial running time. The noise strength can be measured by adjustable parameters, for instance, \( \delta_1, \delta_2 \) for additive and multiplicative noise, and \( p_n \) for one-bit noise. We denote \( g_\theta(f) \) as a type of noisy fitness that disturbs the original fitness function \( f \) by noise with parameter \( \theta \) (where \( \theta \) can be a tuple, e.g., \( \theta = (\delta_1, \delta_2) \) for additive noise), and define the PNT in Definition 2, which characterizes the maximum range of the noise parameter for allowing a polynomial expected running time. Note that the PNT is \( \emptyset \) if the algorithm never has a polynomial expected running time for any noise strength. We study the PNT of EAs in order to analyze the effectiveness of noise-handling strategies.

**Definition 2 (Polynomial Noise Tolerance):** For an algorithm \( A \) running on a problem \( f \) with a type of noise \( g_\theta \), let \( ERT(A; g_\theta(f)) \) be the expected running time of \( A \) on \( f \) with noise strength represented by the parameter \( \theta \) (where \( \theta \) can be a tuple, e.g., \( \theta = (\delta_1, \delta_2) \) for additive noise), and define the PNT in Definition 2, which characterizes the maximum range of the noise parameter for allowing a polynomial expected running time. Then, the polynomial noise tolerance of \( A \) on \( f \) with the type of noise \( g_\theta \) is the range of the noise strength in which the expected running time is polynomial to the problem size \( n \), that is,

\[
PNT(A; f, g_\theta) = \{ \theta \mid ERT(A; g_\theta(f)) = \text{poly}(n) \}.
\]

### 2.2 Evolutionary Algorithms

Evolutionary algorithms (Bäck, 1996) are a type of population-based metaheuristic optimization algorithm. Although many variants exist, the common procedure for EAs can be described as follows:

1. Generate an initial set of solutions (called a population).
2. Reproduce new solutions from the current population.
3. Evaluate the newly generated solutions.
4. Update the population by removing the bad solutions.
5. Repeat steps 2–5 until a specific criterion is met.
The (1 + 1)-EA, as in Algorithm 1, is a simple EA for maximizing pseudo-Boolean problems over \{0, 1\}^n, which reflects the common structure of EAs. It maintains only one solution and repeatedly improves the current solution by using bitwise mutation (i.e., step 3 of Algorithm 1). It has been widely used for the running time analysis of EAs (e.g., by He and Yao, 2001; Droste et al., 2002).

Algorithm 1 (1+1)-EA

Given pseudo-Boolean function f with solution length n, the algorithm consists of the following steps:
1. \( x := \text{randomly selected from } \{0, 1\}^n \)
2. Repeat until the termination condition is met
3. \( x' := \text{flip each bit of } x \text{ independently with probability } p \)
4. \( \text{if } f(x') \geq f(x) \)
5. \( x := x' \)

where \( p \in (0, 0.5) \) is the mutation probability.

The (1 + λ)-EA, as in Algorithm 2, applies an offspring population size \( \lambda \). In each iteration, it first generates \( \lambda \) offspring solutions by independently mutating the current solution \( \lambda \) times, and then selects the best from the current solution and the offspring solutions as the next solution. It has been used to disclose the effect of offspring population size by running time analysis (Jansen et al., 2005; Neumann and Wegener, 2007). Note that the (1 + 1)-EA is a special case of the (1 + λ)-EA with \( \lambda = 1 \).

The running time of EAs is usually defined as the number of fitness evaluations (i.e., computing \( f(\cdot) \)) until an optimal solution is found for the first time, since the fitness evaluation is often the computational process with the highest cost of the algorithm (He and Yao, 2001; Yu and Zhou, 2008).

Algorithm 2 (1+λ)-EA

Given pseudo-Boolean function f with solution length n, the algorithm consists of the following steps:
1. \( x := \text{randomly selected from } \{0, 1\}^n \)
2. Repeat until the termination condition is met
3. \( i := 1 \)
4. Repeat until \( i > \lambda \)
5. \( x_i := \text{flip each bit of } x \text{ independently with probability } p \)
6. \( i := i + 1 \)
7. \( \text{if } \max\{f(x_1), \ldots, f(x_\lambda)\} \geq f(x) \)
8. \( x = \arg \max_{x' \in \{x_1, \ldots, x_\lambda\}} f(x') \)

where \( p \in (0, 0.5) \) is the mutation probability.

The (1 + 1)-EA, as in Algorithm 1, is a simple EA for maximizing pseudo-Boolean problems over \{0, 1\}^n, which reflects the common structure of EAs. It maintains only one solution and repeatedly improves the current solution by using bitwise mutation (i.e., step 3 of Algorithm 1). It has been widely used for the running time analysis of EAs (e.g., by He and Yao, 2001; Droste et al., 2002).

The (1 + λ)-EA, as in Algorithm 2, applies an offspring population size \( \lambda \). In each iteration, it first generates \( \lambda \) offspring solutions by independently mutating the current solution \( \lambda \) times, and then selects the best from the current solution and the offspring solutions as the next solution. It has been used to disclose the effect of offspring population size by running time analysis (Jansen et al., 2005; Neumann and Wegener, 2007). Note that the (1 + 1)-EA is a special case of the (1 + λ)-EA with \( \lambda = 1 \).

The running time of EAs is usually defined as the number of fitness evaluations (i.e., computing \( f(\cdot) \)) until an optimal solution is found for the first time, since the fitness evaluation is often the computational process with the highest cost of the algorithm (He and Yao, 2001; Yu and Zhou, 2008).

2.3 Markov Chain Modeling

We analyze EAs by modeling them as Markov chains. Here, we give some preliminaries.

EAs often generate solutions only based on their currently maintained solutions; thus they can be modeled and analyzed as Markov chains (e.g., He and Yao, 2001; Yu and Zhou, 2008; Yu et al., 2015). A Markov chain \( \{\xi_t\}_{t=0}^{\infty} \) modeling an EA is constructed by taking the EA’s population space \( \mathcal{X} \) as the chain’s state space, \( \xi_t \in \mathcal{X} \). Let \( \mathcal{X}^* \subset \mathcal{X} \) denote the set of all optimal populations that 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 $X^*$ can be analyzed by studying the corresponding Markov chain with the optimal state space $X^*$. Note that we consider the discrete state space (i.e., $X$ is discrete) in this article.

A Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ ($\xi_t \in X$) is a random process, where $\forall t \geq 0$, $\xi_{t+1}$ depends only on $\xi_t$. A Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ is said to be homogeneous if

$$\forall t \geq 0, \forall x, y \in X : P(\xi_{t+1} = y | \xi_t = x) = P(\xi_1 = y | \xi_0 = x). \quad (1)$$

In this article, we always denote $X$ and $X^*$ as the state space and the optimal state space of a Markov chain, respectively.

Given a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ and $\xi_t = x$, we define the first hitting time (FHT) of the chain as a random variable $\tau$ such that $\tau = \min\{t | \xi_{t+1} \in X^*, t \geq 0\}$. That is, $\tau$ is the number of steps needed to reach the optimal state space for the first time starting from $\xi_t = x$. The mathematical expectation of $\tau$, $E[\tau | \xi_t = x] = \sum_{t=0}^{+\infty} t P(\tau = t)$, is called the expected first hitting time (EFHT) of this chain starting from $\xi_t = x$. If $\xi_0$ is drawn from a distribution $\pi_0$, $E[\tau | \xi_0 \sim \pi_0] = \sum_{x \in X} \pi_0(x) E[\tau | \xi_0 = x]$ is the expected first hitting time of the Markov chain over the initial distribution $\pi_0$.

For the corresponding EA, the running time is the number of calls to the fitness function until meeting an optimal solution for the first time. Thus, the expected running time starting from $\xi_0$ and that starting from $\xi_0 \sim \pi_0$ are respectively equal to

$$N_1 + N_2 \cdot E[\tau | \xi_0] \quad \text{and} \quad N_1 + N_2 \cdot E[\tau | \xi_0 \sim \pi_0], \quad (2)$$

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, $N_1 = 1$ and $N_2 = 1$; for the $(1+\lambda)$-EA, $N_1 = 1$ and $N_2 = \lambda$. Note that, when involving the expected running time of an EA in a problem in this article, if the initial population is not specified, it is the expected running time starting from a uniform initial distribution $\pi_u$, that is, $N_1 + N_2 \cdot E[\tau | \xi_0 \sim \pi_u] = N_1 + N_2 \cdot \sum_{x \in X} \frac{1}{|X|} E[\tau | \xi_0 = x]$.

The following two lemmas on the EFHT of Markov chains (Freidlin, 1996) are used in the article.

**Lemma 1:** Given a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$, we have

$$\forall x \in X^* : E[\|\tau | \xi_t = x\|] = 0;$$

$$\forall x \notin X^* : E[\|\tau | \xi_t = x\|] = 1 + \sum_{y \in X} P(\xi_{t+1} = y | \xi_t = x) E[\|\tau | \xi_{t+1} = y\|].$$

**Lemma 2:** Given a homogeneous Markov chain $\{\xi_t\}_{t=0}^{+\infty}$, it holds

$$\forall t_1, t_2 \geq 0, x \in X : E[\|\tau | \xi_{t_1} = x\|] = E[\|\tau | \xi_{t_2} = x\|].$$

Drift analysis is a commonly used tool for analyzing the EFHT of Markov chains. It was first introduced to the running time analysis of EAs by He and Yao (2001; 2004). Later, it became a popular tool in this field, and advanced variants have been proposed (e.g., Doerr et al., 2012b; Doerr and Goldberg, 2013). In this article, we use the additive version (Lemma 3). To use it, a function $V(x)$ ($x \in X$) has to be constructed to measure the distance of a state $x$ to the optimal state space $X^*$. The distance function $V(x)$ satisfies that $V(x \in X^*) = 0$ and $V(x \notin X^*) > 0$. Then, we need to investigate the progress on the distance to $X^*$ in each step, $E[|V(\xi_t) - V(\xi_{t+1})|]$. An upper (lower) bound of the EFHT can be derived through dividing the initial distance by a lower (upper) bound of the progress.
Lemma 3 (Additive Drift Analysis) (He and Yao, 2001; 2004): Given a Markov chain \( \{\xi_t\}_{t=0}^{+\infty} \) and a distance function \( V(x) \), if it satisfies that for any \( t \geq 0 \) and any \( \xi_t \) with \( V(\xi_t) > 0 \),
\[
0 < c_l \leq \mathbb{E}[V(\xi_t) - V(\xi_{t+1})] \leq c_u,
\]
then the EFHT of this chain satisfies that
\[
V(\xi_0)/c_u \leq \mathbb{E}[\tau]\xi_0\| \leq V(\xi_0)/c_l,
\]
where \( c_l, c_u \) do not depend on \( \xi_t \) and \( t \).

The simplified drift theorem (Oliveto and Witt, 2011; 2012) as presented in Lemma 4 was proposed to prove exponential lower bounds on the FHT of Markov chains, where \( X_t \) is usually represented by a mapping of \( \xi_t \). It requires two conditions: a constant negative drift and exponentially decaying probabilities of jumping toward or away from the goal state. To relax the requirement of a constant negative drift, advanced variants have been proposed, for instance, the simplified drift theorem with self-loops (Rowe and Sudholt, 2014) and the simplified drift theorem with scaling (Oliveto and Witt, 2014; 2015). In this article, we use the original version (Lemma 4).

Lemma 4 (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. \( \mathbb{E}[X_t - X_{t+1} | a < X_t < b] \leq -\epsilon; \)
2. \( P(|X_{t+1} - X_t| \geq j | X_t > a) \leq \frac{r(l)}{(l + \delta)^l} \) for \( j \in \mathbb{N}_0 \).

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

2.4 Pseudo-Boolean Functions

The pseudo-Boolean function class in Definition 3 is a large function class that only requires the solution space to be \( \{0, 1\}^n \) and the objective space to be \( \mathbb{R} \). Many well-known NP-hard problems (e.g., the vertex cover problem and the 0-1 knapsack problem) belong to this class. Diverse pseudo-Boolean problems with different structures and difficulties have been used to disclose properties of EAs (e.g., Droste et al., 1998; 2002; He and Yao, 2001). We consider only maximization problems in this article. In the following, let \( x_i \) denote the \( i \)th bit of a solution \( x \in \{0, 1\}^n \).

Definition 3 (Pseudo-Boolean Function): A function in the pseudo-Boolean function class has the form: \( f : \{0, 1\}^n \rightarrow \mathbb{R} \).

The Trap problem in Definition 4 is a special instance in this class, in which the aim is to maximize the number of 0 bits of a solution except for the global optimum 11...1 (briefly denoted as 1\(^n\)). Its optimal function value is 2\(^n\), and the function value for any nonoptimal solution is not larger than 0. It has been used in the theoretical studies of EAs, and the expected running time of the \((1+1)\)-EA with mutation probability \( \frac{1}{n} \) has been proved to be \( \Theta(n^n) \) (Droste et al., 2002). It has also been recognized as the hardest instance in the pseudo-Boolean function class with a unique global optimum for the \((1+1)\)-EA (Qian et al., 2012), that is, the expected running time of the \((1+1)\)-EA on the Trap problem is the largest among the class.
DEFINITION 4 (TRAP PROBLEM): Trap problem of size $n$ is to solve the problem

$$\arg \max_{x \in \{0, 1\}^n} \left( f(x) = 3n \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i \right).$$

The Peak problem in Definition 5 has the same fitness for all solutions except for the global optimum $1^n$. It has been shown that for solving this problem, the $(1+1)$-EA with mutation probability $\frac{1}{n}$ needs $2^\Omega(n)$ running time with an overwhelming probability (Oliveto and Witt, 2011).

DEFINITION 5 (PEAK PROBLEM): Peak problem of size $n$ is to solve the problem

$$\arg \max_{x \in \{0, 1\}^n} \left( f(x) = \prod_{i=1}^{n} x_i \right).$$

The OneMax problem in Definition 6 aims to maximize the number of 1 bits of a solution. Its optimal solution is $1^n$ with the function value $n$. The running time of EAs has been well studied on the OneMax problem (He and Yao, 2001; Droste et al., 2002; Sudholt, 2013); particularly, the expected running time of the $(1+1)$-EA with mutation probability $\frac{1}{n}$ is $\Theta(n \log n)$ (Droste et al., 2002). It has also been recognized as the easiest instance in the pseudo-Boolean function class with a unique global optimum for the $(1+1)$-EA (Qian et al., 2012).

DEFINITION 6 (ONEMAX PROBLEM): OneMax problem of size $n$ is to solve the problem

$$\arg \max_{x \in \{0, 1\}^n} \left( f(x) = \sum_{i=1}^{n} x_i \right).$$

3 On the Effect of Noisy Fitness

In this section, we provide two types of problems in which the noise can make the optimization easier for EAs. By easier, we mean that the EA with noise needs less expected running time than that without noise to find the optimal solution.

We analyze EAs by modeling them as Markov chains. Here, we first give some properties of Markov chains, which are used in the following analysis. We define a partition of the state space of a homogeneous Markov chain based on the EFHT in Definition 7, and then define a jumping probability of a chain from one state to one state space in Definition 8. It is easy to see that $X_0$ in Definition 7 is just $X^*$, since $\mathbb{E}[\tau | \xi_0 \in X^*] = 0$.

DEFINITION 7 (EFHT PARTITION): For a homogeneous Markov chain $\{\xi_t\}_{t=0}^{\infty}$, the EFHT partition is a partition of $X$ into nonempty subspaces $\{X_0, X_1, \ldots, X_m\}$ such that

1. $\forall x, y \in X_i$, $\mathbb{E}[\tau | \xi_0 = x] = \mathbb{E}[\tau | \xi_0 = y]$;

2. $\mathbb{E}[\tau | \xi_0 \in X_0] < \mathbb{E}[\tau | \xi_0 \in X_1] < \cdots < \mathbb{E}[\tau | \xi_0 \in X_m]$.

Note that the EFHT partition is different from the fitness partition used in the fitness-level method (Wegener, 2002; Sudholt, 2013) for EAs’ running time analysis, since the solutions with the same fitness can have different EFHTs, and the EFHT order can be either consistent (e.g., the $(1+\lambda)$-EA on the Trap problem, as in Lemma 6) or inconsistent (e.g., the $(1+\lambda)$-EA on the OneMax problem, as in Lemma 10) with the fitness order.

DEFINITION 8: For a Markov chain $\{\xi_t\}_{t=0}^{\infty}$, $P_t(x, X') = \sum_{y \in X'} P(\xi_{t+1} = y | \xi_t = x)$ is the probability of jumping from state $x$ to state space $X' \subseteq X$ in one step at time $t$. 

Evolutionary Computation Volume 26, Number 1
Lemma 5 compares the EFHT of two Markov chains. It intuitively means that if one chain always has a larger probability of jumping into good states (i.e., $X_j$ with small $j$ values), it needs less time for reaching the optimal state space.

**LEMMA 5:** Given a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ and a homogeneous Markov chain $\{\xi'_t\}_{t=0}^{+\infty}$ with the same state space $\mathcal{X}$ and the same optimal space $\mathcal{X}^*$, let $\{X_0, X_1, \ldots, X_m\}$ denote the EFHT partition of $\{\xi_t\}_{t=0}^{+\infty}$. If for all $t \geq 0$, $x \in \mathcal{X} - X_{0}$, and for all integers $i \in [0, m - 1]$, $V = 0$, we prove one direction of the inequality, and the other can be proved similarly. We use Lemma 3 to derive a bound on $\mathbb{E}[\tau|\xi_0 = x]$, and the same optimal space with $\sum_{i=0}^{m} P_i \cdot E_i = \sum_{i=0}^{m} Q_i \cdot E_i$. Then, we investigate $\mathbb{E}[V(\xi_t) - V(\xi_{t+1})|\xi_t = x]$ for any $x$ with $V(x) > 0$.

\[
\sum_{j=0}^{i} P^i(x, X_j) \geq (\leq) \sum_{j=0}^{i} P^i(x, X_j),
\]

then for all $x \in \mathcal{X}$, $\mathbb{E}[\tau|\xi_0 = x] \leq (\geq) \mathbb{E}[\tau'|\xi'_0 = x]$.

To prove Lemma 5, we need the following lemma, which is proved by using the property of majorization and Schur concavity.

**LEMMA 6:** Let $m$ ($m \geq 1$) be an integer. If it satisfies that

1. $0 \leq E_0 < E_1 < \cdots < E_m$;
2. $\forall 0 \leq i \leq m$, $P_i, Q_i \geq 0$, $\sum_{i=0}^{m} P_i = \sum_{i=0}^{m} Q_i = 1$;
3. $\forall 0 \leq k \leq m - 1$, $\sum_{i=0}^{k} P_i \leq \sum_{i=0}^{k} Q_i$,

then it holds that

$$
\sum_{i=0}^{m} P_i \cdot E_i \geq \sum_{i=0}^{m} Q_i \cdot E_i.
$$

**PROOF:** Let $f(x_0, \ldots, x_m) = \sum_{i=0}^{m} E_i x_i$. Because of condition 1 that $E_i$ is increasing, $f$ is Schur-concave (Marshall et al., 2011, Theorem A.3). Conditions 2 and 3 imply that the vector $(Q_0, \ldots, Q_m)$ majorizes $(P_0, \ldots, P_m)$. Thus, we have $f(P_0, \ldots, P_m) \geq f(Q_0, \ldots, Q_m)$, which proves the lemma. \(\square\)

**PROOF OF LEMMA 5:** We prove one direction of the inequality, and the other can be proved similarly. We use Lemma 3 to derive a bound on $\mathbb{E}[\tau|\xi_0]$ based on which this lemma holds.

To use Lemma 3 to analyze $\mathbb{E}[\tau|\xi_0]$, we first construct a distance function $V(x)$ as

$$
\forall x \in \mathcal{X}, V(x) = \mathbb{E}[\tau|\xi_0 = x],
$$

which satisfies that $V(x \in \mathcal{X}^*) = 0$ and $V(x \notin \mathcal{X}^*) > 0$ by Lemma 1. Then, we investigate $\mathbb{E}[V(\xi_t) - V(\xi_{t+1})|\xi_t = x]$ for any $x$ with $V(x) > 0$.

\[
\mathbb{E}[V(\xi_t) - V(\xi_{t+1})|\xi_t = x] = V(x) - \mathbb{E}[V(\xi_{t+1})|\xi_t = x]
\]
\[
= V(x) - \sum_{y \in \mathcal{X}} P(\xi_{t+1} = y|\xi_t = x)V(y)
\]
\[
= 1 + \sum_{y \in \mathcal{X}} P(\xi_t = y|\xi'_0 = x)\mathbb{E}[\tau|\xi'_1 = y] - \sum_{y \in \mathcal{X}} P(\xi_{t+1} = y|\xi_t = x)\mathbb{E}[\tau'|\xi'_0 = y]
\]
(by Eq. (4) and Lemma 1)
Analyzing Evolutionary Optimization in Noisy Environments

\[ = 1 + \sum_{y \in \mathcal{X}} P(\xi_{i+1} = y | \xi_i = x) \mathbb{E}[\tau^\prime | \xi_0 = y] - \sum_{y \in \mathcal{X}} P(\xi_{i+1} = y | \xi_i = x) \mathbb{E}[\tau^\prime | \xi_0 = y] \]

(by Eq. (1) and Lemma 2, since \( \{\xi_i\}_{i=0}^{+\infty} \) is homogeneous)

\[ = 1 + \sum_{j=0}^{m} \left(P^j(x, \mathcal{X}_j) - P^j(x, \mathcal{X}_j)\right) \mathbb{E}[\tau^\prime | \xi_0 \in \mathcal{X}_j] \]  

(by Definitions 7 and 8).

Since \( \sum_{j=0}^{m} P^j(x, \mathcal{X}_j) = \sum_{j=0}^{m} P^j(x, \mathcal{X}_j) = 1 \), \( \mathbb{E}[\tau^\prime | \xi_0 \in \mathcal{X}_j] \) increases with \( j \) and Eq. (3) holds, by Lemma 6, we have

\[ \sum_{j=0}^{m} P^j(x, \mathcal{X}_j) \mathbb{E}[\tau^\prime | \xi_0 \in \mathcal{X}_j] \geq \sum_{j=0}^{m} P^j(x, \mathcal{X}_j) \mathbb{E}[\tau^\prime | \xi_0 \in \mathcal{X}_j]. \]

Thus, we have, for all \( t \geq 0 \), all \( x \in \mathcal{X}^* \), \( \mathbb{E}[V(\xi_t) - V(\xi_{t+1}) | \xi_t = x] \geq 1. \)

By Lemma 3, we get for all \( x \in \mathcal{X} \), \( \mathbb{E}[\tau | \xi_0 = x] \leq V(x) = \mathbb{E}[\tau | \xi_0 = x]. \)  

\[ \square \]

3.1 On Deceptive Problems

Most practical EAs employ time-invariant operators; thus we can model an EA without noise by a homogeneous Markov chain, while for an EA with noise, since noise may change over time, we can just model it by a Markov chain. In the following analysis, we always denote them respectively by \( \{\xi_i\}_{i=0}^{+\infty} \) and \( \{\xi_i\}_{i=0}^{+\infty} \), and denote the EFHT partition of \( \{\xi_i\}_{i=0}^{+\infty} \) by \( \{\mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_m\} \).

An evolutionary process can be characterized by variation (i.e., producing new solutions) and selection (i.e., weeding out bad solutions). Denote the state spaces before and after variation by \( \mathcal{X} \) and \( \mathcal{X}_{\text{var}} \) respectively, and then the variation process is a mapping \( \mathcal{X} \rightarrow \mathcal{X}_{\text{var}} \) and the selection process is a mapping \( \mathcal{X} \times \mathcal{X}_{\text{var}} \rightarrow \mathcal{X} \) (e.g., for the \((1+\lambda)\)-EA on any pseudo-Boolean problem, \( \mathcal{X} = \{0, 1\}^n \) and \( \mathcal{X}_{\text{var}} = \{\{x_1, \ldots, x_\lambda\} | x_i \in \{0, 1\}^n\} \)). Note that \( \mathcal{X} \) is just the state space of the Markov chain. Let \( P_{\text{var}}(x, x') \) denotes the state transition probability by the variation process. Let \( S^* \) denote the optimal solution set. The considered solution set (e.g., population) may be a multiset. For two multisets \( y \subseteq x \), we mean that \( \forall y \in y : x \in x \).

**Definition 9 (Deceptive Markov Chain):** A homogeneous Markov chain \( \{\xi_i\}_{i=0}^{+\infty} \) modeling an EA optimizing a problem without noise is deceptive if for any \( x \in \mathcal{X}_k \) (\( k \geq 1 \)),

\[ \forall 1 \leq j < k - 1 : P^j(x, \mathcal{X}_j) = 0; \]

\[ \forall k + 1 \leq i \leq m : \sum_{j=i}^{m} P^j(x, \mathcal{X}_j) \geq \sum_{y \in \mathcal{X} \setminus S^*} P_{\text{var}}(x, y). \]

**Theorem 1:** For an EA \( A \) optimizing a problem \( f \), which can be modeled by a deceptive Markov chain, if

\[ \forall x \notin \mathcal{X}_0 : P^j(x, \mathcal{X}_j) = \sum_{x' \setminus S^* \neq \emptyset} P_{\text{var}}(x, x'), \]

then noise makes \( f \) easier for \( A \).

The theorem intuitively means that if an evolutionary process is deceptive and the optimal solution is always accepted once generated in the noisy evolutionary process, then noise will be helpful.

Evolutionary Computation  Volume 26, Number 1 11
Proof: The two EAs with and without noise are different only on whether the fitness evaluation is disturbed by noise; thus they must have the same values on $N_1$ and $N_2$ for their running time Eq. (2). Then, comparing their expected running time is equivalent to comparing the EFHTs of their corresponding Markov chains.

In one step of the evolutionary process, denote the states before and after variation by $x \in \mathcal{X}$ and $x' \in \mathcal{X}_{\text{var}}$ respectively, and denote the state after selection by $y \in \mathcal{X}$. Because the selection process does not produce new solutions, it must satisfy that $y \subseteq x \cup x'$. Assume that $x \in \mathcal{X}_k$ ($k \geq 1$). For $(\xi_i)_{i=0}^{+\infty}$ (i.e., without noise), we have

$$P^i_{\xi}(x, \mathcal{X}_0) \leq \sum_{x' \in \mathcal{X} \setminus \mathcal{X}_0} \sum_{j \in \mathcal{X}_0 \setminus \{x', x\}} P_{\text{var}}(x, x'). \quad (7)$$

For $(\xi_i)_{i=0}^{+\infty}$ (i.e., with noise), the condition Eq. (6) makes that once an optimal solution is generated, it will be always accepted. Thus, we have

$$\forall k + 1 \leq i \leq m : \sum_{j=i}^{m} P^i_{\xi}(x, \mathcal{X}_j) \leq \sum_{j \in \mathcal{X}_0 \setminus \{x', x\}} P_{\text{var}}(x, x'). \quad (8)$$

By combining Eqs. (5)–(8), we have

$$\forall 1 \leq i \leq m : \sum_{j=i}^{m} P^i_{\xi}(x, \mathcal{X}_j) \leq \sum_{j \in \mathcal{X}_0 \setminus \{x', x\}} P_{\text{var}}(x, x').$$

Since $\sum_{j=0}^{m} P^i_{\xi}(x, \mathcal{X}_j) = \sum_{j=0}^{m} P^i_{\xi}(x, \mathcal{X}_j) = 1$, this inequality is equivalent to

$$\forall 0 \leq i \leq m - 1 : \sum_{j=0}^{i} P^i_{\xi}(x, \mathcal{X}_j) \geq \sum_{j=0}^{i} P^i_{\xi}(x, \mathcal{X}_j),$$

which implies that the condition Eq. (3) of Lemma 5 holds. Thus, by Lemma 5, we get $\forall x \in \mathcal{X}$, $\mathbb{E}[\tau|\xi_0 = x] \leq \mathbb{E}[\tau|\xi_0 = y]$, i.e., noise makes $f$ easier for $A$. \hfill $\square$

Then, we give a concrete deceptive evolutionary process, that is, the $(1+\lambda)$-EA optimizing the Trap problem. For the Trap problem given in Definition 4, it is to maximize the number of 0 bits except for the optimal solution 1$^n$. It is not hard to see that the EFHT $\mathbb{E}[\tau|\xi_0 = x]$ only depends on $|x_0|$ (i.e., the number of 0 bits). We denote $\mathbb{E}_1(j)$ as $\mathbb{E}[\tau|\xi_0 = x]$ with $|x_0| = j$. The order of $\mathbb{E}_1(j)$ is shown in Lemma 6.

**Lemma 6:** For any mutation probability $0 < p < 0.5$, it holds that $\mathbb{E}_1(0) < \mathbb{E}_1(1) < \mathbb{E}_1(2) < \cdots < \mathbb{E}_1(n)$.

For proving Lemma 6, we need the following two lemmas. Lemma 8 (Witt, 2013) says that it is more likely that the offspring generated by mutating a parent solution with fewer 0 bits has a smaller number of 0 bits. Note that we consider $|\cdot|_0$ instead of $|\cdot|_1$ in their original lemma. It still holds because of symmetry. We have also restricted $p < 0.5$ instead of $p \leq 0.5$, which leads to the strict inequality in the conclusion. Lemma 9 is very similar to Lemma 6, except that the inequalities in condition 3 and the conclusion hold strictly.

**Lemma 8** (Witt, 2013): Let $x, y \in \{0, 1\}^n$ be two search points satisfying $|x_0| < |y_0|$. Denote by $x'$ and $y'$ the random strings obtained by flipping each bit of $x$ and $y$ independently with probability $p$, respectively. If $p < 0.5$, then for any $0 \leq j \leq n - 1$,

$$P(|x'|_0 \leq j) > P(|y'|_0 \leq j).$$
**Lemma 9**: Let \( m (m \geq 1) \) be an integer. If it satisfies that

1. \( 0 \leq E_0 < E_1 < \cdots < E_m; \)
2. \( \forall 0 \leq i \leq m, P_i, Q_i \geq 0, \sum_{i=0}^{m} P_i = \sum_{i=0}^{m} Q_i = 1; \)
3. \( \forall 0 \leq k \leq m - 1, \sum_{i=0}^{k} P_i < \sum_{i=0}^{k} Q_i, \)

then it holds that \( \sum_{i=0}^{m} P_i \cdot E_i > \sum_{i=0}^{m} Q_i \cdot E_i. \)

**Proof**: Let \( R_i = P_i \) for \( 0 \leq i \leq m - 2, R_{m-1} = \sum_{i=0}^{m-1} Q_i - \sum_{i=0}^{m-2} P_i \) and \( R_m = Q_m \). Then, it is easy to see that the two vectors \((R_0, \ldots, R_m)\) and \((Q_0, \ldots, Q_m)\) satisfy conditions 2 and 3 of Lemma 6. Furthermore, condition 1 of Lemma 6 that \( E_i \) is increasing holds. Thus, by Lemma 6, we have \( \sum_{i=0}^{m} R_i \cdot E_i \geq \sum_{i=0}^{m} Q_i \cdot E_i. \)

Then, we compare \( \sum_{i=0}^{m} P_i \cdot E_i \) with \( \sum_{i=0}^{m} R_i \cdot E_i. \)

\[
\sum_{i=0}^{m} P_i \cdot E_i - \sum_{i=0}^{m} R_i \cdot E_i = \left( P_{m-1} - \left( \sum_{i=0}^{m-1} Q_i - \sum_{i=0}^{m-2} P_i \right) \right) E_{m-1} + (P_m - Q_m)E_m
\]

\[
= \left( \sum_{i=0}^{m-1} P_i - \sum_{i=0}^{m-1} Q_i \right) (E_{m-1} - E_m) > 0.
\]

Thus, we have \( \sum_{i=0}^{m} P_i \cdot E_i > \sum_{i=0}^{m} R_i \cdot E_i \geq \sum_{i=0}^{m} Q_i \cdot E_i, \) that is, the lemma holds. □

**Proof of Lemma 6**: First, \( \mathbb{E}_1(0) < \mathbb{E}_1(1) \) trivially holds, because \( \mathbb{E}_1(0) = 0 \) and \( \mathbb{E}_1(1) > 0. \)

Then, we prove \( \forall 0 < j < n : \mathbb{E}_1(j) < \mathbb{E}_1(j + 1) \) inductively on \( j \).

1. **Initialization** is to prove \( \mathbb{E}_1(n - 1) < \mathbb{E}_1(n) \). For \( \mathbb{E}_1(n) \), because the next solution can be only \( 1^n \) or \( 0^n \), we have \( \mathbb{E}_1(n) = 1 + (1 - (1 - p^n)^\lambda)\mathbb{E}_1(0) + (1 - p^n)^\lambda\mathbb{E}_1(n), \) we have \( \mathbb{E}_1(n) = 1/(1 - (1 - p^n)^\lambda). \) For \( \mathbb{E}_1(n - 1) \), because the next solution can be \( 1^n, 0^n \) or a solution with \( n - 1 \) number of \( 0 \) bits, we have \( \mathbb{E}_1(n - 1) = 1 + (1 - (1 - p^{n-1}(1 - p))^\lambda)\mathbb{E}_1(0) + P \cdot \mathbb{E}_1(n) + (1 - p^{n-1}(1 - p))^\lambda - P)\mathbb{E}_1(n - 1) \), where \( P \) denotes the probability that the next solution is \( 0^n \). Then, \( \mathbb{E}_1(n - 1) = (1 + P\mathbb{E}_1(n))/(1 - (1 - p^{n-1}(1 - p))^\lambda + P). \) Thus, we have

\[
\frac{\mathbb{E}_1(n - 1)}{\mathbb{E}_1(n)} = \frac{1 - (1 - p^n)^\lambda + P}{1 - (1 - p^{n-1}(1 - p))^\lambda + P} < 1,
\]

where the inequality is by \( 0 < p < 0.5 \).

2. **Inductive hypothesis** assumes that

\[
\forall K < j \leq n - 1 (K \geq 1) : \mathbb{E}_1(j) < \mathbb{E}_1(j + 1).
\]

Then, we consider \( j = K. \) Let \( x \) and \( y \) be a solution with \( K + 1 \) number of \( 0 \) bits and that with \( K \) number of \( 0 \) bits, respectively. Let \( a \) and \( b \) denote the number of \( 0 \) bits of the offspring solutions \( \text{mut}(x) \) and \( \text{mut}(y) \), respectively. That is, \( a = |\text{mut}(x)|_0 \) and \( b = |\text{mut}(y)|_0 \). For the \( \lambda \) independent mutations on \( x \) and \( y \), we use \( a_1, \ldots, a_\lambda \) and \( b_1, \ldots, b_\lambda \), respectively. Note that, \( a_1, \ldots, a_\lambda \) are independently and identically distributed (i.i.d.), and \( b_1, \ldots, b_\lambda \) are also i.i.d. Let \( p_j = P(a_i \leq j) \) and \( q_j = P(b_i \leq j) \).

Then, from Lemma 8, we have \( \forall 0 \leq j \leq n - 1 : p_j < q_j. \)

For \( \mathbb{E}_1(K + 1) \), let \( P_0 \) and \( P_1 (1 \leq i \leq n) \) be the probability that for the \( \lambda \) offspring solutions, the least number of \( 0 \) bits is 0 (i.e., \( P_0 = P(\min\{a_1, \ldots, a_\lambda\} = 0) \)), and the largest number of \( 0 \) bits is \( i \), while the least number of \( 0 \) bits is larger than 0 (i.e., \( P_1 =

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/evco_a_00170 by guest on 23 May 2021
C. Qian, Y. Yu, and Z.-H. Zhou

\[ P(\max\{a_1, \ldots, a_k\} = i \land \min\{a_1, \ldots, a_k\} > 0) \], respectively. By considering the mutation and selection behavior of the \((1+\lambda)\)-EA on the Trap problem, we have

\[
\mathbb{E}_1(K + 1) = 1 + P_0\mathbb{E}_1(0) + \sum_{i=1}^{K+1} P_i\mathbb{E}_1(K + 1) + \sum_{i=K+2}^{n} P_i\mathbb{E}_1(i).
\]

For \( \mathbb{E}_1(K) \), let \( Q_0 = P(\min\{b_1, \ldots, b_k\} = 0) \) and \( Q_i = P(\max\{b_1, \ldots, b_k\} = i \land \min\{b_1, \ldots, b_k\} > 0) \). Then, we have

\[
\mathbb{E}_1(K) = 1 + Q_0\mathbb{E}_1(0) + \sum_{i=1}^{K} Q_i\mathbb{E}_1(K) + \sum_{i=K+1}^{n} Q_i\mathbb{E}_1(i).
\]

For comparing \( \mathbb{E}_1(K + 1) \) with \( \mathbb{E}_1(K) \), we need to show that

\[
\forall 0 \leq j \leq n - 1: \sum_{i=0}^{j} P_i < \sum_{i=0}^{j} Q_i. \tag{9}
\]

For \( \sum_{i=0}^{j} P_i \), we have

\[
\sum_{i=0}^{j} P_i = P(\min\{a_1, \ldots, a_k\} = 0) + P(\max\{a_1, \ldots, a_k\} \leq j \land \min\{a_1, \ldots, a_k\} > 0)
\]

\[
= P(a_1 = 0 \lor \ldots \lor a_k = 0) + P(0 < a_1 \leq j \land \ldots \land 0 < a_k \leq j)
\]

\[
= 1 - (1 - p_0)^k + (p_j - p_0)^j < 1 - (1 - p_0)^k + (q_j - p_0)^j \quad \text{(by } p_j < q_j\text{)}.
\]

For \( \sum_{i=0}^{j} Q_i \), we similarly have \( \sum_{i=0}^{j} Q_i = 1 - (1 - q_0)^k + (q_j - q_0)^j \). Thus,

\[
\sum_{i=0}^{j} Q_i - \sum_{i=0}^{j} P_i > (1 - p_0)^k - (1 - q_0)^k + (q_j - q_0)^j - (q_j - p_0)^j
\]

\[
= ((1 - q_0 + q_0 - p_0)^k - (1 - q_0)^k) - ((q_j - q_0 + q_0 - p_0)^k - (q_j - q_0)^k)
\]

\[
= f(q_j - q_0) - f(q_j - q_0).
\]

where the last equality is by letting \( f(x) = (x + q_0 - p_0)^k - x^k \).

Since \( q_0 > p_0 \), it is easy to verify that \( f(x) \) is increasing. Then, we have \( f(1 - q_0) > f(q_j - q_0) \) by \( q_j < 1 \). Thus, the Eq. (9) holds.

By subtracting \( \mathbb{E}_1(K) \) from \( \mathbb{E}_1(K + 1) \), we get

\[
\mathbb{E}_1(K + 1) - \mathbb{E}_1(K) = \left( P_0\mathbb{E}_1(0) + \sum_{i=1}^{K+1} P_i\mathbb{E}_1(K + 1) + \sum_{i=K+2}^{n} P_i\mathbb{E}_1(i) - Q_0\mathbb{E}_1(0)\right)
\]

\[
- \sum_{i=1}^{K+1} Q_i\mathbb{E}_1(K + 1) + \sum_{i=K+1}^{n} Q_i(\mathbb{E}_1(K + 1) - \mathbb{E}_1(K))
\]

\[
> \sum_{i=1}^{K} Q_i(\mathbb{E}_1(K + 1) - \mathbb{E}_1(K))
\]

where the inequality is by applying Lemma 9 to the formula in \(( \cdot )\). The three conditions of Lemma 9 can be easily verified, because \( \mathbb{E}_1(0) = 0 < \mathbb{E}_1(K + 1) < \cdots < \mathbb{E}_1(n) \) by
inductive hypothesis; \( \sum_{i=0}^{n} P_i = \sum_{i=0}^{n} Q_i = 1 \); and Eq. (9) holds. Because \( \sum_{i=1}^{K} Q_i < 1 \), we have \( E_1(K + 1) > E_1(K) \).

(3) Conclusion. According to steps (1) and (2), the lemma holds. \( \square \)

**Theorem 2:** Either additive noise with \( \delta_2 - \delta_1 < 2n \) or multiplicative noise with \( \delta_2 > \delta_1 > 0 \) makes the Trap problem easier for the \((1 + \lambda)-\text{EA}\) with mutation probability less than 0.5.

**Proof:** First, we are to show that the \((1 + \lambda)-\text{EA}\) optimizing the Trap problem can be modeled by a deceptive Markov chain. By Lemma 6, the EFHT partition of \( \{ \xi_i \}_{i=0}^{+\infty} \) is \( \mathcal{X}_i = \{ x \in \{0, 1 \}^n \mid |x|_0 = i \} (0 \leq i \leq n \) and \( m \) in Definition 7 is equal to \( n \) here.

For any \( x \in \mathcal{X}_k (k \geq 1) \), we denote \( P(0) \) and \( P(j) (1 \leq j \leq n) \) as the probability that for the \( \lambda \) offspring solutions \( x_1, \ldots, x_\lambda \) generated by bitwise mutation on \( x, \min\{|x|_1, \ldots, |x|_\lambda\} = 0 \) (i.e., the least number of 0 bits is 0), and \( \min\{|x|_1, \ldots, |x|_\lambda\} > 0 \) \( \wedge \max\{|x|_1, \ldots, |x|_\lambda\} = j \) (i.e., the largest number of 0 bits is \( j \), while the least number of 0 bits is larger than 0), respectively. For \( \{ \xi_i \}_{i=0}^{+\infty} \), because only the optimal solution or the solution with the largest number of 0 bits among the parent solution and \( \lambda \) offspring solutions will be accepted, we have

\[
\forall 1 \leq j \leq k - 1: P^x_\xi(x, \mathcal{X}_j) = 0; \quad \forall k + 1 \leq j \leq n : P^x_\xi(x, \mathcal{X}_j) = P(j).
\]

This implies that Eq. (5) holds.

Then, we are to show that the condition of Theorem 1 (i.e., Eq. (6)) holds. For \( \{ \xi_i \}_{i=0}^{+\infty} \) with additive noise, since \( \delta_2 - \delta_1 < 2n \), we have

\[
f^N(1^n) \geq f(1^n) + \delta_1 > 2n + \delta_2 - 2n = \delta_2; \quad \forall y \neq 1^n, f^N(y) \leq f(y) + \delta_2 \leq \delta_2.
\]

For multiplicative noise, since \( \delta_2 > \delta_1 > 0 \), then \( f^N(1^n) > 0 \) and \( \forall y \neq 1^n, f^N(y) \leq 0 \). Thus, for these two noises, we have \( \forall y \neq 1^n, f^N(1^n) > f^N(y) \), which implies that if the optimal solution \( 1^n \) is generated, it will always be accepted. Thus, we have \( \mathcal{X}_0 = \{ 1^n \}, P^x_\xi(x, \mathcal{X}_0) = P(0) \). This implies that Eq. (6) holds.

Thus, by Theorem 1, we get that the Trap problem becomes easier for the \((1 + \lambda)-\text{EA}\) under these two types of noise. \( \square \)

### 3.2 On Flat Problems

Besides deceptive problems, we show that noise can also make flat problems easier for EAs. We take the Peak problem given in Definition 5 as the representative problem, which has the same fitness for all solutions except for the optimal solution \( 1^n \). When using EAs to solve it, it provides no information for the search direction; thus it is hard for EAs. We analyze the \((1 + 1)-\text{EA}^*\) optimizing the Peak problem. The \((1 + 1)-\text{EA}^*\) is the same as the \((1 + 1)-\text{EA}\) except that it employs the strict selection strategy. That is, step 4 of Algorithm 1 changes to be “if \( f(x') > f(x) \).” The expected running time of the \((1 + 1)-\text{EA}^*\) with mutation probability \( \frac{1}{n} \) on the Peak problem has been proved to be lower-bounded by \( e^\pi \ln(n/2) \) (Droste et al., 2002).

**Theorem 3:** One-bit noise with \( p_n \in (0, 1) \) being a constant makes the Peak problem easier for the \((1 + 1)-\text{EA}^*\) with mutation probability \( \frac{1}{n^2} \), when starting from an initial solution \( x \) with \( |x|_0 > \frac{1 + p_n}{p_n(1 - p_n)} \).

**Proof:** Let \( \{ \xi_i \}_{i=0}^{+\infty} \) and \( \{ \xi'_i \}_{i=0}^{+\infty} \) model the \((1 + 1)-\text{EA}^*\) with one-bit noise and without noise for maximizing the Peak problem, respectively. It is not hard to see that both the EFHT \( E[\mathcal{T}|\xi_0 = x] \) and \( E[\mathcal{T}|\xi'_0 = x] \) only depend on \( |x|_0 \). We denote \( E(i) \) and \( E'(i) \) as \( E[\mathcal{T}|\xi_0 = x] \) and \( E[\mathcal{T}|\xi'_0 = x] \) with \( |x|_0 = i \), respectively.
For \( \{\xi_t\}_{t=0}^{+\infty} \) (i.e., without noise) starting from a solution \( x \) with \( |x|_0 = i > 0 \), in one step, any nonoptimal offspring solution has the same fitness as the parent and then will be rejected because of the strict selection strategy; only the optimal solution can be accepted, which happens with probability \( \frac{1}{n} (1 - \frac{1}{n})^{n-i} \). Thus, we have

\[
\mathbb{E}'(i) = 1 + \frac{1}{n^i} \left( 1 - \frac{1}{n} \right)^{n-i} \mathbb{E}'(0) + \left( 1 - \frac{1}{n^i} \left( 1 - \frac{1}{n} \right)^{n-i} \right) \mathbb{E}(i),
\]

which leads to \( \mathbb{E}'(i) = n^i \left( \frac{n^i}{n} \right)^{n-i} \).

For \( \{\xi_t\}_{t=0}^{+\infty} \) (i.e., with one-bit noise), we assume using reevaluation, which reevaluates \( f(x) \) and evaluates \( f(x') \) in each iteration of Algorithm 1. When starting from \( x \) with \( |x|_0 = i \), if the generated offspring \( x' \) is the optimal solution \( 1^n \), it will be accepted with probability \( (1 - p_n)(1 - p_{n-1}) \) because only no bit flip for noise on \( x' \) and no 0 bit flip for noise on \( x \) can make \( f^N(x') > f^N(x) \); otherwise, \( x \) will keep \( |x|_0 = 1 \), because \( f^N(x') \leq f^N(x) \) for any \( x' \) with \( |x'|_0 \geq 2 \). Thus, we have

\[
\mathbb{E}(1) = 1 + \frac{1}{n} \left( 1 - \frac{1}{n} \right)^{n-1} (1 - p_n) \left( 1 - \frac{p_n}{n} \right) \mathbb{E}(0)
\]

\[
+ \left( 1 - \frac{1}{n} \left( 1 - \frac{1}{n} \right)^{n-1} (1 - p_n) \left( 1 - \frac{p_n}{n} \right) \right) \mathbb{E}(1),
\]

which leads to \( \mathbb{E}(1) = n \left( \frac{n^i}{n} \right)^{n-i} \frac{1}{(1-p_n)(1-\frac{n}{n})} \).

When starting from \( x \) with \( |x|_0 = i \geq 2 \), if the offspring \( x' \) is \( 1^n \), it will be accepted with probability \( (1 - p_n) \) because only no bit flip for noise on \( x' \) can make \( f^N(x') > f^N(x) \); if \( |x'|_0 = 1 \), it will be accepted with probability \( p_{n-1} \) because only flipping the unique 0 bit for noise on \( x' \) can make \( f^N(x') > f^N(x) \); otherwise, \( x \) keeps \( |x|_0 = i \) because \( f^N(x') = f^N(x) \) for any \( x' \) with \( |x'|_0 \geq 2 \). Let \( \text{mut}_{i-1} \) be the probability of mutating \( |x|_0 = i \) to \( |x|_0 = 1 \) by bitwise mutation with \( p = \frac{1}{n} \). Then, we have, for \( i \geq 2 \),

\[
\mathbb{E}(i) = 1 + \frac{(n-1)^{n-i}}{n^n} (1 - p_n) \mathbb{E}(0) + \text{mut}_{i-1} \left( \frac{p_n}{n} \right) \mathbb{E}(1)
\]

\[
+ \left( 1 - \frac{(n-1)^{n-i}}{n^n} (1 - p_n) - \text{mut}_{i-1} \left( \frac{p_n}{n} \right) \right) \mathbb{E}(i),
\]

which leads to \( \mathbb{E}(i) = \frac{1 + \text{mut}_{i-1} \left( \frac{p_n}{n} \right) \mathbb{E}(1)}{1 - \frac{1}{n^i} \left( 1 - \frac{1}{n} \right)^{n-i} (1 - p_n) - \text{mut}_{i-1} \left( \frac{p_n}{n} \right)} \).

From Eq. (2), we know that the expected running times without noise and with one-bit noise are \( 1 + \mathbb{E}'(i) \) and \( 1 + 2 \mathbb{E}(i) \), respectively. To prove that one-bit noise can be helpful, we need to show that there exists \( i \geq 1 \) such that \( 2 \mathbb{E}(i) < \mathbb{E}'(i) \). Obviously, \( i = 1 \) is impossible because \( \mathbb{E}(1) > \mathbb{E}'(1) \). Then, for larger \( i \) with \( i > \frac{1 + p_n}{p_n (1 - \frac{1}{n})} \),

\[
(2 \mathbb{E}(i) - \mathbb{E}'(i)) \cdot \left( \frac{1}{n^i} \left( 1 - \frac{1}{n} \right)^{n-i} (1 - p_n) + \text{mut}_{i-1} \left( \frac{p_n}{n} \right) \right)
\]

\[
= 1 + p_n - \text{mut}_{i-1} \left( \frac{p_n}{n} \right) \left( \frac{1}{n^{i-1}} \left( 1 - \frac{1}{n} \right)^{n-i+1} + \frac{p_n}{n} n^{i-1} \right) \left( 1 - \frac{1}{n} \right)^{n-i} \left( 1 - \frac{1}{n} \right) < 0,
\]
where the first inequality is because \( \mu_{t \rightarrow 1} \geq \frac{1}{n^t} (1 - \frac{1}{n})^{n-i+1} \) and \( \mathbb{E}(1) \leq n^i (\frac{n}{n-i})^{n-i} \) for large enough \( n \) and \( p_n \) being constant, and the last inequality is by \( i > \frac{1+p_n}{p_n(1-\frac{1}{n})} \).

This is equivalent to \( 2\mathbb{E}(i) - \mathbb{E}'(i) < 0 \), which implies that noise is helpful when starting from an initial solution \( x \) with \( |x|_0 > \frac{1+p_n}{p_n(1-\frac{1}{n})} \).

This theorem implies that the Peak problem becomes easier under noise when starting from an initial solution \( x \) with a large number of 0 bits. From the analysis, we can see that the reason for requiring a large \( |x|_0 \) is to make \( \mu_{t \rightarrow 1} \) much larger than \( \mu_{t \rightarrow 0} \), which means that the negative effect of rejecting the optimal solution by noise can be compensated by the positive effect of accepting the solution \( x \) with \( |x|_0 = 1 \).

For the \((1 + 1)\)-EA solving the Peak problem, any offspring solution will be accepted because its fitness is always not less than the fitness of the parent solution; thus the solution \( x \) in the evolutionary process almost performs a random walk over \( \{0, 1\}^n \). In this case, we can intuitively find a similar effect of one-bit noise as that found in the \((1 + 1)\)-EA solving the Peak problem. Here, we assume that the single-evaluation strategy is used. Under one-bit noise, for any nonoptimal parent solution \( x \), if \( |x|_0 \geq 2 \), then \( f^N(x) = 0 \) and any offspring will be accepted; if \( |x|_0 = 1 \) and \( f^N(x) = 0 \), then any offspring will be accepted; if \( |x|_0 = 1 \) and \( f^N(x) = 1 \), any offspring \( x' \) with \( |x'|_0 \geq 2 \) will be rejected because \( f^N(x') = 0 < f^N(x) \), and the optimal solution \( |x'|_0 = 0 \) will be rejected with probability \( p_n \). Compared with the transition behavior without noise, noise only has an effect when \( |x|_0 = 1 \) and \( f^N(x) = 1 \): the negative effect of rejecting the optimal solution, which has the probability \( \frac{1}{n}(1 - \frac{1}{n})^{n-1}p_n \), and the positive effect of rejecting \( |x'|_0 \geq 2 \), which has the probability at least \( \frac{n-1}{n}(1 - \frac{1}{n})^{n-1} \). Obviously, the negative effect can be compensated by the positive effect, which implies that one-bit noise is helpful. Thus, we have the following conjecture. The rigorous analysis is not easy, and we leave it to future work. We instead verify it in the experiment section.

**Conjecture 1**: One-bit noise makes the Peak problem easier for the \((1 + 1)\)-EA with mutation probability \( \frac{1}{n} \).

## 4 On the Effect of Noise-Handling Strategies

In the previous section, we found that noise can make optimization easier for EAs when the problem presents some deceptiveness and flatness. Meanwhile, on other problems, noisy fitness evaluation can make an optimization harder for EAs. For example, Droste (2004) proved that the running time of the \((1 + 1)\)-EA on the OneMax problem can increase from polynomial to exponential in the presence of noise. Thus, in this section, we investigate how well different noise-handling strategies can perform when the noise is indeed harmful.

### 4.1 A Noise-Harmful Case

We consider the case that the \((1 + \lambda)\)-EA is used for optimizing the OneMax problem. Let \( \{s_i\}_{i=0}^{\infty} \) and \( \{s'_i\}_{i=0}^{\infty} \) model the \((1 + \lambda)\)-EA with and without noise for maximizing OneMax, respectively. It is not hard to see that the EFHT \( \mathbb{E}[\tau | s'_0 = x] \) only depends on \( |x|_0 \). We denote \( \mathbb{E}_2(j) \) as \( \mathbb{E}[\tau | s'_0 = x] \) with \( |x|_0 = j \). The order of \( \mathbb{E}_2(j) \) is shown in Lemma 10.

**Lemma 10**: For any mutation probability \( 0 < p < 0.5 \), it holds that \( \mathbb{E}_2(0) < \mathbb{E}_2(1) < \mathbb{E}_2(2) < \cdots < \mathbb{E}_2(n) \).
PROOF: We prove $\forall 0 \leq j < n : E_2(j) < E_2(j + 1)$ inductively on $j$.

1. **Initialization** is to prove $E_2(0) < E_2(1)$, which holds, since $E_2(1) > 0 = E_2(0)$.
2. **Inductive hypothesis** assumes that
   \[ \forall 0 \leq j < K (K \leq n - 1) : E_2(j) < E_2(j + 1). \]

Then, we consider $j = K$. We use the similar analysis method as in the proof of Lemma 6 to compare $E_2(K + 1)$ with $E_2(K)$.

For $E_2(K + 1)$, let $P_i (0 \leq i \leq n)$ be the probability that the least number of 0 bits for the $\lambda$ offspring solutions is $i$ (i.e., $P_i = P(\min\{a_1, \ldots, a_\lambda\} = i)$). By considering the mutation and selection behavior of the $(1 + \lambda)$-EA on the OneMax problem, we have

\[
E_2(K + 1) = \sum_{i=0}^{K} P_i E_2(i) + \sum_{i=K+1}^{n} P_i E_2(K + 1).
\]

For $E_2(K)$, let $Q_i = P(\min\{b_1, \ldots, b_\lambda\} = i)$. We have

\[
E_2(K) = \sum_{i=0}^{K-1} Q_i E_2(i) + \sum_{i=K}^{n} Q_i E_2(K).
\]

By subtracting $E_2(K)$ from $E_2(K + 1)$, we get

\[
E_2(K + 1) - E_2(K) = \sum_{i=K+1}^{n} P_i (E_2(K + 1) - E_2(K))
\]

\[
+ \left( \sum_{i=0}^{K-1} P_i E_2(i) + \sum_{i=K}^{n} P_i E_2(K) - \sum_{i=0}^{K-1} Q_i E_2(i) - \sum_{i=K}^{n} Q_i E_2(K) \right)
\]

\[
> \sum_{i=K+1}^{n} P_i (E_2(K + 1) - E_2(K)),
\]

where the inequality is by applying Lemma 9 to the formula in $(\cdot)$. The three conditions of Lemma 9 can be easily verified, because $E_2(0) < E_2(1) < \cdots < E_2(K)$ by inductive hypothesis; $\sum_{i=0}^{n} P_i = \sum_{i=0}^{n} Q_i = 1$; and the following inequality holds.

\[
\sum_{i=0}^{j} Q_i - \sum_{i=0}^{j} P_i = P(\min\{b_1, \ldots, b_\lambda\} \leq j) - P(\min\{a_1, \ldots, a_\lambda\} \leq j)
\]

\[
= P(b_1 \leq j \vee \cdots \vee b_\lambda \leq j) - P(a_1 \leq j \vee \cdots \vee a_\lambda \leq j)
\]

\[
= 1 - (1 - q_j)^\lambda - (1 - (1 - p_j)^\lambda) > 0 \quad (\text{by } p_j < q_j).
\]

Because $\sum_{i=K+1}^{n} P_i < 1$, we have $E_2(K + 1) > E_2(K)$.

3. **Conclusion.** According to steps (1) and (2), the lemma holds. \qed

**THEOREM 4:** Any noise makes the OneMax problem harder for the $(1 + \lambda)$-EA with mutation probability less than 0.5.

**PROOF:** We use Lemma 5 to prove it. By Lemma 10, the EFHT partition of $\{x_i\}_{i=0}^{+\infty}$ is $X_i = \{x \in \{0, 1\}^n | |x_0| = i\}$ $(0 \leq i \leq n)$.

For any nonoptimal solution $x \in X_i$ $(k > 0)$, we denote $P(j) \ (0 \leq j \leq n)$ as the probability that the least number of 0 bits for the $\lambda$ offspring solutions generated by bitwise
mutation on \( x \) is \( j \). For \( \{\xi_t\}_{t=0}^{+\infty} \), because the solution with the least number of 0 bits among the parent solution and \( \lambda \) offspring solutions will be accepted, we have

\[
\forall 0 \leq j \leq k - 1 : P^t_\xi(x, X_j) = P(j); \quad P^t_\xi(x, X_k) = \sum_{j=k}^n P(j).
\]

For \( \{\xi_t\}_{t=0}^{+\infty} \), because of the fitness evaluation disturbed by noise, the solution with the least number of 0 bits among the parent and \( \lambda \) offspring solutions may be rejected. Thus, we have

\[
0 \leq i \leq k - 1 : \sum_{j=0}^i P^t_\xi(x, X_j) \leq \sum_{j=0}^i P(j).
\]

Then, we get

\[
\forall 0 \leq i \leq n - 1 : \sum_{j=0}^i P^t_\xi(x, X_j) \leq \sum_{j=0}^i P^t_\xi(x, X_j),
\]

which implies that the condition Eq. (3) of Lemma 5 holds. Thus, we get \( \forall x \in X \), \( \mathbb{E}[\tau|\xi_0 = x] \geq \mathbb{E}[\tau'|\xi'_0 = x] \), namely, noise makes the OneMax problem harder for the \((1 + \lambda)\)-EA.

\[ \square \]

In the following sections we analyze the effect of different noise-handling strategies for the \((1 + 1)\)-EA (a specific case of the \((1 + \lambda)\)-EA), optimizing the OneMax problem to investigate their usefulness.

## 4.2 On Reevaluation and Threshold Selection Strategies

### 4.2.1 Reevaluation

There are naturally two fitness evaluation options for EAs (Arnold and Beyer, 2002; Jin and Branke, 2005; Goh and Tan, 2007):

- **Single-evaluation.** We evaluate a solution once, and use the evaluated fitness for this solution in the future.

- **Reevaluation.** We access the fitness of a solution by evaluation every time.

For example, for the \((1 + 1)\)-EA in Algorithm 1, if using reevaluation, both \( f(x') \) and \( f(x) \) will be calculated and recalculated in each iteration; if using single-evaluation, only \( f(x') \) will be calculated and the previous obtained fitness \( f(x) \) will be reused. Note that the analysis in the previous section without explicitly indicating the employed evaluation strategy assumes single-evaluation.

Sudholt and Thyssen (2012), for an ACO with single-evaluation solving stochastic shortest path problems, constructed an example graph to show that exponential running time is required for approximating real shortest paths. The difficulty is because once a path is luckily evaluated to have a relatively small length due to noise, it will always be preferred and make the ACO get stuck in an inferior solution. By using reevaluation instead of single-evaluation when evaluating the best-so-far path, the ACO can easily solve the example graph (Doerr et al., 2012a). Reevaluation has also been employed for EAs solving noisy multiobjective optimization problems (e.g., Buche et al., 2002; Park and Ryu, 2011; Fieldsend and Everson, 2015).

Intuitively, reevaluation can smooth noise and thus could be better for noisy optimizations, but it also increases the fitness evaluation cost and thus increases the running time. Its usefulness was not clear.
In this section we compare these two options for the (1+1)-EA, solving the OneMax problem under one-bit noise to show whether reevaluation is useful. Note that for one-bit noise, \( p_n \) controls the noise strength, that is, noise becomes stronger as \( p_n \) gets larger, and it is also the parameter of the PNT. In the following analysis, let \( \text{poly}(n) \) indicate any polynomial of \( n \).

**Theorem 5:** For the (1+1)-EA with mutation probability \( \frac{1}{n} \) solving the OneMax problem under one-bit noise, if using single-evaluation, the PNT is \([0, 1 - 1/\Theta(\text{poly}(n))]\).

The theorem is straightforwardly derived from the following two lemmas. Lemma 11 tells us the expected running time upper bound \( O(n^2 + n/(1 - p_n)) \), which implies that the expected running time is polynomial if \( \frac{1}{1-p_n} \in O(\text{poly}(n)) \), i.e., \( p_n \in 1 - O(\text{poly}(n)) \).

Lemma 12 tells us the lower bound \( \Omega(n \log n + p_n/(1 - p_n)) \), which implies that the running time is superpolynomial if \( \frac{1}{1-p_n} \in \omega(\text{poly}(n)) \), i.e., \( p_n \in 1 - \frac{1}{\omega(\text{poly}(n))} \). By combining these results, we get that the maximum noise strength allowing polynomial expected running time is \( 1 - \frac{1}{o(\text{poly}(n))} \), i.e., the PNT is \([0, 1 - 1/\Theta(\text{poly}(n))]\).

**Lemma 11:** For the (1+1)-EA using single-evaluation with mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, the expected running time is upper-bounded by \( O(n^2 + n/(1 - p_n)) \).

**Proof:** Let \( L \) denote the noisy fitness value \( f^N(x) \) of the current solution \( x \). Because the (1+1)-EA does not accept a solution with a smaller fitness (step 4 of Algorithm 1) and does not reevaluate the fitness of the current solution \( x \), \( 0 \leq L \leq n \) will never decrease. By applying the fitness level technique (Wegener, 2002; Sudholt, 2013), we first analyze the expected steps until \( L \) increases when starting from \( L = i \) (denoted by \( \mathbb{E}[i] \)) and then sum them up to get an upper bound \( \sum_{i=0}^{n-1} \mathbb{E}[i] \) for the expected steps until \( L \) reaches the maximum value \( n \). For \( \mathbb{E}[i] \), we analyze the probability \( P \) that \( L \) increases in two steps when \( L = i \), then \( \mathbb{E}[i] = 2 \cdot \frac{1}{n} \). Note that one-bit noise can make \( L \) be \( \lfloor x_1 \rfloor - 1, \lfloor x_1 \rfloor \) or \( \lfloor x_1 \rfloor + 1 \), where \( \lfloor x_1 \rfloor = \sum_{i=1}^n x_i \) is the number of 1 bits. When analyzing the noisy fitness \( f^N(x') \) of the offspring \( x' \) in each step, we need to first consider bitwise mutation on \( x \) and then one random bit flip for noise.

When \( 0 < L < n - 1 \), \( \lfloor x_1 \rfloor = L - 1 \), or \( L = 1 \).

1. For \( \lfloor x_1 \rfloor = L - 1 \), \( P \geq \frac{n-n-1}{n} p_n \frac{n-n}{n} + \frac{n-n}{n} (1 - \frac{1}{n})^{n-1} (1 - p_n) \frac{n-n}{n} (1 - \frac{1}{n})^{n-1} (1 - p_n) \), since it is sufficient to flip one 0 bit for mutation and one 0 bit for noise in the first step, or flip one 0 bit for mutation and no bit for noise in the first step and flip one 0 bit for mutation and no bit for noise in the second step.

2. For \( \lfloor x_1 \rfloor = L \), \( P \geq (1 - \frac{1}{n}) p_n \frac{n-n}{n} + \frac{n-n}{n} (1 - \frac{1}{n})^{n-1} (1 - p_n) \), since it is sufficient to flip no bit for mutation and one 0 bit for noise, or flip one 0 bit for mutation and no bit for noise in the first step.

3. For \( \lfloor x_1 \rfloor = L + 1 \), \( P \geq (1 - \frac{1}{n}) p_n \frac{n-n}{n} + \frac{n-n}{n} (1 - \frac{1}{n})^{n-1} (1 - p_n) \), since it is sufficient to flip no bit for mutation and no bit for noise, or flip one 0 bit for mutation and no bit for noise in the first step.

Thus, for these three cases, we have

\[
P \geq p_n \left(1 - \frac{1}{n}\right) \frac{n-1}{n} \frac{n-L-n}{n} + \left(1 - \frac{1}{n}\right) \frac{2(n-1)}{n} \frac{n-L-n}{n} \frac{n-L-n}{n} \frac{1-n}{n} \geq \frac{3(n-L)(n-L-1)}{4e^2n^2} \quad \text{(by } 1 - \frac{1}{n} \text{ is lower than } 1 \text{ and } 0 \leq p_n \leq 1 \text{).}
\]

When \( L = 0 \), \( \lfloor x_1 \rfloor = 0 \) or 1. By considering cases 2 and 3, we get the same lower bound for \( P \).
When \( L = n - 1 \) and the optimal solution \( 1^n \) has not been found, \( |x|_1 = n - 2 \) or \( n - 1 \). By considering cases 1 and 2, we get \( P \geq 3/(2e^2n^2) \).

Based on this analysis, we get that the expected steps until \( L = n \) are at most

\[
\sum_{i=0}^{n-1} \mathbb{E}[i] \leq 2 \cdot \left( \sum_{L=0}^{n-2} \frac{4e^2n^2}{3(n-L)(n-L-1)} + \frac{2e^2n^2}{3} \right) \in O(n^2).
\]

When \( L = n \), \( |x|_1 = n - 1 \) or \( n \). The equality \( |x|_1 = n \) means that the optimal solution has been found. Because we can get an upper bound for the expected running time of finding \( 1^n \), we can pessimistically assume that \( |x|_1 = n - 1 \). Starting from \( |x|_1 = n - 1 \) and \( L = n \) (i.e., the current solution has \( n - 1 \) one bits and the fitness is \( n \)), it will always keep in such a situation before finding \( 1^n \), and the optimal solution \( 1^n \) can be generated and accepted in one step only through flipping the unique 0 bit for mutation and no bit for noise, which happens with probability \( \frac{1}{n}(1 - \frac{1}{n})^{n-1}(1 - p_n) \geq \frac{1}{2}(1 - p_n) \). This implies that the expected steps for finding the optimal solution are at most \( \frac{1}{2}(1 - p_n) \).

Thus, the total expected running time is upper-bounded by \( O(n^2 + \frac{n}{1-p_n}) \).

**Lemma 12:** For the \((1+1)\)-EA using single-evaluation with mutation probability \( \frac{1}{n} \) on the One-Max problem under one-bit noise, the expected running time is lower-bounded by \( \Omega(n \log n + p_n/(1 - p_n)) \).

**Proof:** Assume that the number of 1 bits of the initial solution \( x \) is less than \( n - 1 \), that is, \( |x|_1 < n - 1 \). Let \( T \) denote the running time of finding the optimal solution \( 1^n \) when starting from \( x \). Denote \( A \) as the event that in the evolutionary process, any solution \( x' \) with \( |x'|_1 = n - 1 \) is never found. By the law of total expectation, we have

\[
\mathbb{E}[T] = \mathbb{E}[T|A] \cdot P(A) + \mathbb{E}[T|\bar{A}] \cdot P(\bar{A}).
\]

We are first to show that \( P(\bar{A}) \geq P(A) \). Let \( l : x = x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_{m-1} \rightarrow x_m = 1^n \) denote an evolutionary path from \( x \) to the optimal solution \( 1^n \), which satisfies that \( \forall i \leq m, \ |x|_1 \leq n - 2 \). Then, \( P(A) \) is the sum of the probabilities of all possible such \( l \). For any such \( l \), there must exist a corresponding set of paths \( S(l) = \{x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_{m-1} \rightarrow y_m \rightarrow \cdots \rightarrow 1^n \ | |y_m|_1 = n - 1 \} \), in which the first \( m - 1 \) solutions of any path are the same as that of \( l \) and the \( m \)th solution has \( n - 1 \) number of 1 bits. Let \( q \) denote the probability of the subpath \( x_1 \rightarrow \cdots \rightarrow x_{m-1} \), and let \( |x_{m-1}|_1 = n - j \leq n - 2 \). Then, \( P(l) = q \cdot \frac{1}{n} (1 - \frac{1}{n})^{n-j-1} \). The probability of mutating from \( x_{m-1} \) to \( y_m \) is at least \( j \cdot \frac{1}{n} (1 - \frac{1}{n})^{n-j+1} \), and the acceptance probability of \( y_m \) is at least \( 1 - p_n + p_n \frac{1}{n} \), which is reached when \( |x_{m-1}|_1 = n - 2 \) and \( f^N(x_{m-1}) = n - 1 \). Thus, we have

\[
P(S(l)) \geq q \cdot \frac{1}{n} (1 - \frac{1}{n})^{n-j+1} \cdot (1 - p_n + \frac{p_n}{n}) \geq q \cdot \frac{j}{n} \cdot \frac{1}{n} \cdot \frac{1}{n} \geq P(l).
\]

Moreover, for any two different paths \( l_1, l_2 \), it must hold that \( S(l_1) \cap S(l_2) = \emptyset \). Thus, \( P(\bar{A}) \geq P(A) \). Because \( P(\bar{A}) + P(A) = 1 \), we get \( P(\bar{A}) \geq 1/2 \). Then,

\[
\mathbb{E}[T] \geq \mathbb{E}[T|A] \cdot P(A) + \frac{1}{2} \mathbb{E}[T|\bar{A}].
\]

We are then to derive a lower bound on \( \mathbb{E}[T|\bar{A}] \). We further divide the running time \( T \) into two parts: the running time until finding a solution \( x' \) with \( |x'|_1 = n - 1 \) for the first time (denoted by \( T_1 \)), and the remaining running time for finding the optimal solution (denoted by \( T_2 \)). Thus, we have

\[
\mathbb{E}[T|\bar{A}] = \mathbb{E}[T_1|\bar{A}] + \mathbb{E}[T_2|\bar{A}].
\]
For $\mathbb{E}[T_2 | A]$, when finding a solution $x'$ with $|x'|_1 = n - 1$ for the first time, we consider the case that the fitness is evaluated as $n$, which happens with probability $p_n \frac{1}{2}$. If it happens, because of the single-evaluation strategy, the solution will always have $n - 1$ number of 1 bits and its fitness will always be $n$. From the upper-bound analysis in Lemma 11, we know that the probability of generating and accepting the optimal solution in one step such a situation is $\frac{1}{n} (1 - \frac{1}{n})^{n-1} (1 - p_n) \leq \frac{(1 - p_n)}{n}$. Thus,

$$\mathbb{E}[T_2 | A] \geq p_n \frac{1}{n} \cdot \frac{n}{1 - p_n} = \frac{p_n}{1 - p_n},$$

which implies that $\mathbb{E}[T | A] \geq \frac{p_n}{1 - p_n}$, and thus $\mathbb{E}[T] \geq \frac{p_n}{1 - p_n}$.

Because the initial solution is uniformly distributed over $\{0, 1\}^n$, we have $P(|x|_1 < n - 1) = 1 - \frac{n+1}{2^n}$. Thus, the expected running time of the whole process is lower-bounded by $(1 - \frac{n+1}{2^n}) \cdot \frac{p_n}{2(1-p_n)}$, i.e., $\Omega(\frac{p_n}{1 - p_n})$.

Note that when $1 - p_n \in \Omega(1)$, the derived lower bound $\Omega(\frac{p_n}{1 - p_n})$ would be quite loose. Thus, for filling up this gap, we are to derive another lower bound that does not depend on $p_n$. From Droste et al. (2002, Lemma 10), we know that the expected running time of the $(1 + 1)$-EA to optimize linear functions with positive weights is $\Omega(n \log n)$. Their proof idea is to analyze the expected running time until all the 0 bits of the initial solution have been flipped at least once, which is obviously a lower bound on the expected running time of finding the optimal solution $1^*$. Because noise will not affect this analysis process, we can directly apply their result to our setting and then get the lower bound $\Omega(n \log n)$.

By combining the derived two lower bounds, we get that the expected running time of the whole process is lower-bounded by $\Omega(n \log n + p_n/(1 - p_n))$. \qed

We then show the PNT using reevaluation in the following theorem, which can be straightforwardly derived from Lemma 13.

**Theorem 6:** For the $(1 + 1)$-EA with mutation probability $\frac{1}{n}$ solving the OneMax problem under one-bit noise, if using reevaluation, the PNT is $[0, \Theta(\frac{\log n}{n})]$.

**Lemma 13** (Droste, 2004): For the $(1 + 1)$-EA using reevaluation with mutation probability $\frac{1}{n}$ on the OneMax problem under one-bit noise, the expected running time is polynomial when $p_n \in O(\log(n)/n)$, and superpolynomial when $p_n \in \omega(\log(n)/n)$.

### 4.2.2 Threshold Selection

During the process of evolutionary optimization, most of the improvements in one generation are small. When using reevaluation, because of noisy fitness evaluation, a considerable portion of these improvements are not real, where a worse solution appears to have a better fitness and then survives to replace the true better solution which appears to have a worse fitness. This may mislead the search direction of EAs and slow down the efficiency of EAs or make the EAs get trapped in the local optimal solution (see Section 4.2.1). To deal with this problem, a selection strategy for EAs handling noise was proposed (Markon et al., 2001; Bartz-Beielstein, 2005a), namely, threshold selection, where 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, for the $(1 + 1)$-EA with threshold selection as in Algorithm 3, step 4 changes to be “if $f(x') \geq f(x) + \tau$” rather than “if $f(x') \geq f(x)$” in Algorithm 1. Such a strategy can reduce the risk of accepting a bad solution due to noise. Although the good local performance (i.e., the progress of one step) of EAs with threshold selection has
Analyzing Evolutionary Optimization in Noisy Environments

been shown on some problems (Markon et al., 2001; Bartz-Beielstein and Markon, 2002; Bartz-Beielstein, 2005b), its usefulness for the global performance (i.e., the running time until finding the optimal solution) of EAs under noise is not yet clear.

In this section we analyze the running time of the (1+1)-EA with threshold selection solving OneMax under one-bit noise to see whether threshold selection is useful. Note that the analysis here assumes reevaluation. This is because using single-evaluation and threshold selection simultaneously will lead to infinite expected running time for any noise strength \( p_n > 0 \), as shown in the following theorem.

**Theorem 7:** For the (1+1)-EA with mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, if using single-evaluation with threshold selection \( \tau > 0 \), the PNT is \( \{0\} \).

**Proof:** For the noise strength \( p_n > 0 \), it is easy to see that in the evolutionary process, there exists some positive probability that a solution \( x \) with \( |x|_1 = n - 1 \) is found and its fitness is evaluated as \( n \). Once this happens, it will always stay in such a situation because the fitness of the parent solution will never be re-evaluated; the fitness of the offspring solution is at most \( n \); and then any offspring solution will be rejected by the threshold selection strategy. This implies that the optimal solution \( 1^n \) will never be found. Thus, the expected running time is infinite for \( p_n > 0 \).

For \( p_n = 0 \) (i.e., without noise), in the evolution process the number of 1 bits \( i \) of the solution will never decrease. When using threshold selection \( \tau = 1 \), \( i \) can increase in one step with probability at least \( \frac{n-i}{n} (1 - \frac{1}{n})^{n-1} \geq \frac{n-i}{en} \), since it is sufficient to flip one 0 bit and keep other bits unchanged. Thus, by using the fitness level method (Wegener, 2002; Sudholt, 2013), the expected running time is at most \( \sum_{i=0}^{n-1} \frac{e^n}{i} \), i.e., \( O(n \log n) \).

Then, we are to analyze the PNT of the (1+1)-EA using reevaluation and threshold selection on the OneMax problem for different threshold values \( \tau \). Note that the minimal fitness gap for the OneMax problem is 1. Thus, we first analyze \( \tau = 1 \).

**Theorem 8:** For the (1+1)-EA with mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, if using reevaluation with threshold selection \( \tau = 1 \), the PNT is \([0, 1] \).

The theorem can be directly derived from the following lemma, which implies the expected running time upper bound \( O(n \log n) \) for \( p_n \leq 1/(\sqrt{2}e) \) and \( O(n^2 \log n) \) for \( p_n > 1/(\sqrt{2}e) \).

**Lemma 14:** For the (1+1)-EA using reevaluation with threshold selection \( \tau = 1 \) and mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, the expected running time is upper-bounded by \( O(n^2 \log n / p_n^2) \) when \( p_n \in [0, 1] \), and specifically \( O(n \log n) \) when \( p_n \leq 1/(\sqrt{2}e) \).
PROOF: We use additive drift analysis (i.e., Lemma 3) to prove it. Let $H_i = \sum_{j=1}^{i} \frac{1}{j}$ denote the $i$th harmonic number, and $H_0 = 0$. We first construct a distance function $V(x)$ as $\forall x \in X = \{0, 1\}^n$, $V(x) = H_{|x|_0}$, 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 $V(x) = \{0\}$ and $V(x) = \{0\} > 0$.

Then, we investigate $E[V(\xi_i) - V(\xi_{i+1})] = x]$ for any $x$ with $V(x) > 0$ (i.e., $x \notin X^*)$. We denote the number of 0 bits of the current solution $x$ by $i$ ($1 \leq i \leq n$). Let $p_{i,i+d}$ be the probability that the next solution after bitwise mutation and selection has $i + d (-i \leq d \leq n - i)$ number of 0 bits. We then have

$$E[V(\xi_i) - V(\xi_{i+1})]|\xi_t = x] = H_i - \sum_{d=-i}^{n-i} p_{i,i+d} \cdot H_{i+d}. \quad (10)$$

Then, we analyze $p_{i,i+d}$ for $1 \leq i \leq n$. Let $P_d$ denote the probability that the offspring solution $x'$ by bitwise mutation on $x$ has $i + d (-i \leq d \leq n - i)$ number of 0 bits. Note that one-bit noise can change the true fitness of a solution by at most 1, i.e., $|f^N(x) - f(x)| \leq 1$.

(1) When $d \geq 2$, $f^N(x') \leq n - i - d + 1 \leq n - i - 1 \leq f^N(x)$. Because an offspring solution will be accepted only if $f^N(x') \geq f^N(x) + 1$, the offspring $x'$ will be discarded in this case, which implies that $V_d \geq 2 : p_{i,i+d} = 0$.

(2) When $d = 1$, the offspring solution $x'$ will be accepted only if $f^N(x') = n - i \land f^N(x) = n - i - 1$, the probability of which is $p_n^2 \cdot \frac{n^i}{n^{i+1}}$, since it needs to flip one 0 bit of $x'$ and flip one 1 bit of $x$. Thus, $p_{i,i+1} = P_1 \cdot (p_n^2 \cdot \frac{n^{i+1}}{n^{i+2}})$.

(3) When $d = -1$, if $f^N(x) = n - i - 1$, the probability of which is $p_n^2 \cdot \frac{n^{i+1}}{n^{i+2}}$, the offspring solution $x'$ will be accepted, since $f^N(x') \geq n - i + 1 = n - i + 1$, the probability of which is $(1 - p_n) \cdot (1 - p + p_n^{i+1})$, $x'$ will be accepted; if $f^N(x) = n - i + 1 \land f^N(x') = n - i + 2$, the probability of which is $p_n^2 \cdot \frac{n^{i+1}}{n^{i+2}}$, $x'$ will be accepted; otherwise, $x'$ will be discarded. Thus, $p_{i,i-1} = P_1 \cdot (p_n^2 \cdot \frac{n^{i+1}}{n^{i+2}} + (1 - p_n) \cdot (1 - p + p_n^{i+1}) + p_n^2 \cdot \frac{n^{i+1}}{n^{i+2}})$.

(4) When $d \leq -2$, it is easy to see that $p_{i,i+d} > 0$.

By applying these probabilities to Eq. (10), we have

$$E[V(\xi_i) - V(\xi_{i+1})]|\xi_t = x] \geq H_i - p_{i,i-1}H_{i-1} - p_{i,i+1}H_{i+1} - (1 - p_{i,i-1} - p_{i,i+1})H_i$$

$$= p_{i,i-1} \cdot \frac{1}{i} - p_{i,i+1} \cdot \frac{1}{i+1}$$

$$\geq P_1 \left( p_n \frac{n - i}{n} + p_n^2 \frac{i(i-1)}{n^2} \right) \frac{1}{i} - P_1 p_n^2 \frac{(i+1)(n-i)}{n^2} \frac{1}{i+1}. \quad (11)$$

We then bound the two mutation probabilities $P_{-1}$ and $P_1$. 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 $P_{-1} \geq \frac{1}{i} (1 - \frac{1}{n})^{n-1}$. 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

$$P_1 = \min_{n-i,i+1} \left( \sum_{k=1}^{n-i-i+1} \binom{n-i}{k} \left( \frac{i}{k-1} \right) \frac{1}{n^{2k-1}} \left( 1 - \frac{1}{n} \right)^{n-2k+1} \right)$$

$$\leq \frac{n-i}{n} \left( 1 - \frac{1}{n} \right)^{n-1} + \min_{n-i,i+1} \left( \sum_{k=2}^{n-i-i+1} \frac{1}{k!(k-1)} \frac{(n-i)^k}{n^k} \frac{(1-1/n)^{n-2k+1}}{n^{n-2k+1}} \right)$$
Analyzing Evolutionary Optimization in Noisy Environments

\[
\leq \frac{n - i}{n} \left(1 - \frac{1}{n}\right)^{n-1} + \frac{i}{n} \cdot \sum_{k=2}^{\min(n-i,i+1)} \frac{1}{k!(k-1)!} \left(1 - \frac{1}{n}\right)^{n-1}
\]

\[
\leq \frac{n - i}{n} \left(1 - \frac{1}{n}\right)^{n-1} + \frac{i}{n} \cdot \sum_{k=2}^{+\infty} \frac{1}{k!} \left(1 - \frac{1}{n}\right)^{n-1}
\]

\[
= \frac{n - i}{n} \left(1 - \frac{1}{n}\right)^{n-1} + (e - 2) \frac{i}{n} \left(1 - \frac{1}{n}\right)^{n-1}.
\]

By applying these two bounds of \(P_{-1}\) and \(P_1\) to Eq. (11), we have

\[
\mathbb{E}[V(\xi_t) - V(\xi_{t+1})|\xi_t = x] \geq \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} p_n^2 \left(\frac{n - i}{n} + i(i - 1)\right)
\]

\[
- \left(\frac{n - i}{n} + (e - 2) \frac{i}{n}\right) \left(1 - \frac{1}{n}\right)^{n-1} p_n^2 n - i
\]

\[
\geq (3 - e) \frac{i}{n^2} \left(1 - \frac{1}{n}\right)^n p_n^2
\]

\[
\geq \frac{3 - e}{2e} \frac{p_n^2}{n^2} \quad \text{(by} i \geq 1 \text{and} \left(1 - \frac{1}{n}\right)^n \geq \frac{1}{2e}\text{)}.
\]

Thus, by Lemma 3, we get, noting that \(V(x) \leq H_n < 1 + \log n\),

\[
\mathbb{E}[\tau|\xi_0] \leq \frac{2e}{3 - e} \frac{n^2}{p_n^2} V(\xi_0) \in O\left(\frac{n^2 \log n}{p_n^2}\right),
\]

i.e., the expected running time of the \((1+1)\)-EA with \(\tau = 1\) on the OneMax problem is upper-bounded by \(O(n^2 \log n/p_n^2)\).

For \(p_n \leq \frac{1}{\sqrt{2e}}\), we can derive a tighter upper bound \(O(n \log n)\) by applying proper bounds of the two probabilities \(p_{i,i-1}\) and \(p_{i,i+1}\) to Eq. (11). From cases 2 and 3 in the analysis of \(p_{i,i+d}\), we have

\[
p_{i,i-1} \geq P_{-1}(1 - p_n)^2 \geq \frac{i}{n} \left(1 - \frac{1}{n}\right)^{n-1} (1 - p_n)^2;
\]

\[
p_{i,i+1} = P_1 p_n^2 \left(\frac{i + 1}{n} - \frac{(i + 1)(n - i)^2}{n^3}\right) \leq \frac{(i + 1)(n - i)^2}{n^3} p_n^2,
\]

where the last inequality is by \(P_1 \leq \frac{n - i}{n}\), since it is necessary to flip at least one 1 bit. Then, Eq. (11) becomes

\[
\mathbb{E}[V(\xi_t) - V(\xi_{t+1})|\xi_t = x] \geq \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} (1 - p_n)^2 - \frac{(n - i)^2}{n^3} p_n^2
\]

\[
\geq \frac{1}{n} \left(1 - \frac{1}{n}\right)^2 - p_n^2 \geq 0.13 \cdot \frac{1}{n},
\]

where the last inequality is because \(\frac{1}{e}(1 - p_n)^2 - p_n^2\) decreases with \(p_n\) for \(p_n \leq \frac{1}{\sqrt{2e}}\). Thus, by Lemma 3, we have

\[
\mathbb{E}[\tau|\xi_0] \leq \frac{n}{0.13} V(\xi_0) \in O(n \log n).
\]
that is, the expected running time of the (1 + 1)-EA with \( \tau = 1 \) on the OneMax problem is upper-bounded by \( O(n \log n) \) for \( p_n \leq \frac{1}{2 \sqrt{e}} \).

Then, we analyze the effect of a relatively large threshold value \( \tau = 2 \) on the PNT.

**Theorem 9:** For the \((1 + 1)\)-EA with mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, if using reevaluation with threshold selection \( \tau = 2 \), the PNT is \([1/\Theta(\log(n)), 1 - 1/\Theta(\log(n))]\).

The theorem can be directly derived from the following two lemmas.

**Lemma 15:** For the \((1 + 1)\)-EA using reevaluation with threshold selection \( \tau = 2 \) and mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, the expected running time is upper-bounded by \( O(n \log n/(p_n (1 - p_n))) \).

**Proof:** Let \( i \) \((0 \leq i \leq n)\) denote the number of 0 bits of the current solution \( x \). Here, an offspring \( x' \) will be accepted only if \( f(x') - f(x) \geq 2 \). As in the proof of Lemma 14, we can derive \( \forall d \geq 1 : p_{i,i+d} = 0; \forall d \geq 2 : p_{i,i-d} > 0; \)

\[
p_{i,i-1} = P_{-1} \left( p_n \frac{n-i}{n} \left( 1 - p_n + p_n \frac{i-1}{n} \right) + (1 - p_n) \left( p_n \frac{i-1}{n} \right) \right).
\]

Thus, \( i \) never increases, and it decreases in one step with probability at least

\[
p_{i,i-1} \geq \frac{i}{n} \left( 1 - \frac{1}{n} \right)^{n-1} \left( 1 - p_n \right) \frac{p_n}{n} \left( 1 - \frac{1}{n} \right) + p_n \frac{2(n-i)(i-1)}{n^2} \geq \frac{1}{2e} (1 - p_n) p_n \frac{i}{n}.
\]

Then, the expected steps until \( i = 0 \) (i.e., the optimal solution is found) is at most

\[
\sum_{i=1}^{n} \frac{2en}{i(1-p_n)p_n} = O \left( \frac{n \log n}{p_n (1-p_n)} \right).
\]

**Lemma 16:** For the \((1 + 1)\)-EA using reevaluation with threshold selection \( \tau = 2 \) and mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, the expected running time is lower-bounded by \( \Omega(n \log n + n/(p_n (1 - p_n))) \).

**Proof:** The proof is very similar to that of Lemma 12 except the calculation of \( \mathbb{E}[T_2|A] \). Here we first use a different but simple idea to show that \( P(A) \geq 1/2 \). For any evolutionary path with the event \( A \) happening, it has to flip at least two bits in the last step for finding the optimal solution, because any solution \( x' \) with \( |x'|_1 = n - 1 \) is never found. Thus, \( P(A) \leq \left( \frac{\binom{n}{2}}{n^2} \right) \leq \frac{1}{2} \). Since \( P(\bar{A}) + P(A) = 1 \), we have \( P(\bar{A}) \geq 1/2 \).

Then, we analyze \( \mathbb{E}[T_2|\bar{A}] \), which is the expected running time for finding the optimal solution when starting from a solution \( x \) with \( |x|_1 = n - 1 \) (i.e., \( |x|_0 = 1 \)). From the upper-bound analysis in the proof of Lemma 15, we know that once a solution \( x \) with \( |x|_0 = 1 \) is found, it will always satisfy \( |x|_0 = 1 \) before finding the optimal solution, because \( \forall d \geq 1 : p_{i,i+d} = 0 \). Meanwhile, the optimal solution (i.e., \( |x|_0 = 0 \)) will be found in one step with probability \( p_{1,0} = \frac{1}{n} (1 - \frac{1}{n})^{n-1} p_n (1 - p_n) (1 - \frac{1}{n}) \leq \frac{p_n (1 - p_n)}{en} \). We then have \( \mathbb{E}[T_2|\bar{A}] \geq \frac{en}{p_n (1 - p_n)} \). Thus, the lemma holds.

For larger threshold values \( \tau > 2 \), we have the following.

**Theorem 10:** For the \((1 + 1)\)-EA with mutation probability \( \frac{1}{n} \) on the OneMax problem under one-bit noise, if using reevaluation with threshold selection \( \tau > 2 \), the PNT is \( \varnothing \).
PROOF: Let \( i = |x|_0 \) for the current solution \( x \). An offspring solution \( x' \) will be accepted only if \( f^N(x') - f^N(x) \geq \tau > 0 \). Then, we have

\[
\forall d \geq 1: p_{i,i+d} = 0; \quad p_{i,i-1} = \begin{cases} 
    p_{-1} \cdot \left( \frac{n-i}{n} \cdot \frac{i-1}{n} \right), & \text{if } \tau = 3, \\
    0, & \text{otherwise}.
\end{cases}
\]

In the evolutionary process, it is easy to see that there exists some positive probability that a solution \( x \) with \( |x|_0 = 1 \) is found (i.e., \( i = 1 \)). Once this happens, \( i = 1 \) will always hold because \( p_{1,0} = 0 \) and \( p_{1,1+d} = 0 \) for any \( d \geq 1 \). In such a situation, the optimal solution \( 1^n \) will never be found. Thus, the expected running time is infinite for any \( p_n \in [0, 1] \).

### 4.3 Smooth Threshold Selection

We have shown that for the \((1 + 1)\)-EA solving the OneMax problem under one-bit noise, the reevaluation with threshold selection \( \tau = 1 \) can improve the PNT to \([0, 1]\), which means that the expected running time of the \((1 + 1)\)-EA is always polynomial regardless of the noise strength. Under asymmetric one-bit noise, we prove that all the preceding strategies are not effective, however, when the noise probability \( p_n \) equals 1, as shown in Theorem 11.

**THEOREM 11:** For the \((1 + 1)\)-EA with mutation probability \( \frac{1}{n} \) on the OneMax problem under asymmetric one-bit noise, if using threshold selection \( \tau \geq 0 \) with either single-evaluation or reevaluation, the expected running time is at least exponential for \( p_n = 1 \).

**PROOF:** We analyze the expected running time for each strategy.

For single-evaluation with threshold selection \( \tau \geq 0 \), where single-evaluation with threshold selection \( \tau = 0 \) is equivalent to single-evaluation alone, there exists some positive probability that a solution \( x \) with \( |x|_0 = 1 \) and \( f^N(x) = n \) is found. Because the fitness is not reevaluated, \( f^N(1^n) = n - 1 \) due to \( p_n = 1 \); and \( f^N(x) \leq n - 1 \) for \( x \) with \( |x|_0 \geq 2 \), it will always stay in such a state. Thus, the expected running time for finding the optimal solution \( 1^n \) is infinite.

For reevaluation with threshold selection \( \tau = 0 \) (i.e., reevaluation alone), we use the simplified drift theorem (Lemma 4) to prove an exponential running time lower bound. Let \( X_t \) be the number of 0 bits of the solution after \( t \) iterations of the \((1 + 1)\)-EA. We consider the interval \([0, n^{1/4}]\), that is, the parameters \( a = 0 \) (i.e., the global optimum) and \( b = n^{1/4} \) in Lemma 4. Then, we analyze the drift \( \mathbb{E}[X_t - X_{t+1}|X_t = i] \) for \( 1 \leq i < n^{1/4} \). Let \( p_{i,i+d} \) denote the probability that the next solution after bitwise 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

\[
\mathbb{E}[X_t - X_{t+1}|X_t = i] = \sum_{d=1}^{i} d \cdot p_{i,i-d} - \sum_{d=1}^{n-i} d \cdot p_{i,i+d}.
\]

Let \( P_d \) denote the probability that the offspring solution generated by bitwise mutation has \( i+d \) number of 0 bits. Using the same analysis procedure for \( p_{i,i+d} \) as in the proof of Lemma 14 and noting that \( p_n = 1 \) and \( \tau = 0 \) here, we have

\[
\forall d \geq 3, p_{i,i+d} = 0; \quad p_{i,i+2} = P_2/4; \quad p_{i,i+1} = P_1/4; \quad p_{i,i-1} = 3P_{-1}/4; \quad \forall 2 \leq d \leq i, p_{i,i-d} = P_{-d}.
\]
Furthermore, $P_1 \geq \frac{n-i}{n}(1 - \frac{1}{n})^{n-1} \geq \frac{n-i}{en}$, since it is sufficient to flip one bit and keep other bits unchanged, and $P_{-d} \leq \left(\frac{1}{d}\right)^{n}$, since it is necessary to flip at least $d$ number of 0 bits. By applying these probabilities to Eq. (12) we have:

$$
\mathbb{E}[X_t - X_{t+1}|X_t = i] \leq \frac{3P_{-1}}{4} + \sum_{d=2}^{i} d \cdot P_{-d} - \frac{P_1}{4}
$$

$$
\leq \frac{3i}{4n} + \sum_{d=2}^{i} d \cdot \left(\frac{i}{d}\right) \frac{1}{n^d} - n - i \frac{1}{4en} = \frac{i}{n} \left(\left(1 + \frac{1}{n}\right)^{i-1} + \frac{1}{4e} - \frac{1}{4}\right) - \frac{1}{4e}
$$

$$
= -\frac{1}{4e} + O\left(\frac{n^{1/4}}{n}\right) \quad \text{(since } i < n^{1/4}).
$$

Thus, $\mathbb{E}[X_t - X_{t+1}|X_t = i] = -\Omega(1)$, which implies that condition 1 of Lemma 4 holds. For its 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 4 holds with $\delta = 1$ and $r(l) = 2$. 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. Because of 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.

For reevaluation with threshold selection $\tau = 1$, we use the same analysis as for $\tau = 0$. The only difference is the calculation of $p_{l,i,+d}$:

$$
\forall d \geq 2, \quad p_{l,i,+d} = 0; \quad p_{l,i+1} = P_{l}/4;
$$

$$
p_{l,i-1} \leq 3P_{-1}/4; \quad p_{l,i-2} \leq 3P_{-2}/4; \quad \forall 3 \leq d \leq i, \quad p_{l,i,-d} = P_{-d}.
$$

It is easy to verify that the analysis of the two conditions of Lemma 4 will not be affected. Thus, we derive the same result as for $\tau = 0$: the expected running time is exponential.

For reevaluation with threshold selection $\tau \geq 2$, we have

$$
\forall d \geq 1, \quad p_{l,i,+d} = 0; \quad \forall 2 \leq d \leq i, \quad p_{l,i,-d} > 0.
$$

For $p_{l,i-1}$, we need to consider two cases:

$$
\text{for } i \geq 2, \quad p_{l,i,-1} = P_{-1}/4; \quad \text{for } i = 1, \quad p_{l,1,0} = 0.
$$

There exists some positive probability that a solution $x$ with $|x_0| = 1$ is found. For $\tau \geq 2$, we have $\forall d \geq 1, p_{l,1,+d} = 0$ and $p_{l,1,0} = 0$. Thus, it will always stay in such a state (i.e., $|x_0| = 1$), which implies that the expected running time for finding the optimal solution $1^*$ is infinite. \hfill \square

Therefore, the reevaluation with threshold selection is ineffective in this case. When the threshold $\tau \leq 1$, it has too large probability of accepting false progress, which leads to a negative drift and thus the exponential running time. When $\tau \geq 2$, although the probability of accepting false progress is 0 (i.e., $\forall d \geq 1, p_{l,i,+d} = 0$), it has too small probability of accepting true progress (i.e., $p_{l,0} = 0$), which leads to the infinite running time. However, setting $\tau$ between 1 and 2 is useless, because the minimum fitness gap is 1, which makes a value of $\tau \in (1, 2)$ equivalent to $\tau = 2$. 


Analyzing Evolutionary Optimization in Noisy Environments

We propose the smooth threshold selection as in Definition 10, which modifies the original threshold selection by changing the hard threshold value to a smooth one. By smooth we mean that the offspring solution will be accepted with some probability when the fitness gap between the offspring and the parent is just the threshold. For example, (1) + (0.1)-smooth threshold selection accepts the offspring solution with probability 0.9 when the fitness gap is 1; this makes a fractional threshold 1.1 effective. Such a strategy of accepting new solutions probabilistically based on the fitness is similar to the acceptance strategy of simulated annealing (Kirkpatrick, 1984). We are to show that using smooth threshold selection with proper threshold values can improve the PNT to [0, 1] in this case.

**Definition 10 (Smooth Threshold Selection):** Let \( \delta \) be the gap between the fitness of the offspring solution \( x' \) and the parent solution \( x \), i.e., \( \delta = f(x') - f(x) \). Given a threshold \((A) + (B)\) with \(B \in [0, 1]\), the selection process will behave as follows:

1. If \( \delta < A \), \( x' \) will be rejected.
2. If \( \delta = A \), \( x' \) will be accepted with probability \( 1 - B \).
3. If \( \delta > A \), \( x' \) will be accepted.

In the following analysis, we view the evolutionary process as a random walk on a graph (Algorithm 4), which has often been used for analyzing randomized search heuristics (e.g., Giel and Wegener, 2003; Neumann and Witt, 2010). Lemma 17 gives an upper bound on the expected steps for a random walk to visit each vertex of a graph at least once.

**Lemma 17** (Aleliunas et al., 1979): Given an undirected connected graph \( G = (V, E) \), the expected number of steps until each vertex \( v \in V \) has been visited at least once for a random walk on \( G \) is upper-bounded by \( 2|E|(|V| - 1) \).

We first analyze a smooth threshold depending on the current search point.

**Theorem 12:** For the (1 + 1)-EA with mutation probability \( \frac{1}{n} \) on the OneMax problem under asymmetric one-bit noise, if using reevaluation with \((1) + \left(1 - \frac{|x|}{2n}\right)\)-smooth threshold selection, the PNT is \([0, 1]\).

**Proof:** Let \( i \) (0 ≤ \( i \) ≤ \( n \)) denote the number of 0 bits of the current solution \( x \). We first analyze \( p_{i,i+d} \) as that analyzed in the proof of Lemma 14. Note that there are two differences in the analyses, which are caused by different threshold and noise settings, respectively. Because of the threshold difference, the acceptance probability is different when the fitness gap between the offspring \( x' \) and the parent solution \( x \) is 1: \( x' \) will be accepted with probability \( \frac{|x|}{2n} \) here, while it will be always accepted in the proof of Lemma 14. Because of the noise difference, the probability of flipping a 0 or 1 bit is...
different: a random 0 or 1 bit will be flipped with an equal probability of $\frac{1}{2}$ here, while a uniformly randomly chosen bit will be flipped in the proof of Lemma 14.

Thus, we can similarly derive the value of $p_{i,i+d}$ for $1 \leq i \leq n - 1$. It is easy to see that $\forall d \geq 2: p_{i,i+d} = 0$, and $p_{i,i-d} > 0$. For $p_{i,i+1}$, the offspring solution $x'$ will be accepted only if $f^N(x') = n - i \land f^N(x) = n - i - 1$, and the acceptance probability is $\frac{i}{2en}$. The probability of $f^N(x') = n - i$ is at most $p_n$, since it needs to flip one 0 bit of $x'$ in noise; the probability of $f^N(x) = n - i - 1$ is $p_n \frac{1}{2}$, since it needs to flip one 1 bit of $x$.

Thus, $p_{i,i+1} \leq P_1\left(p_n \frac{1}{2} + p_n \cdot \frac{i}{2en}\right)$.

For $p_{i,i-1}$, we need to consider two cases:

(1) $2 \leq i \leq n - 1$. If $f^N(x) = n - i - 1$, the probability of which is $p_n \frac{1}{2}$, there are three cases for the offspring solution $x'$: if $f^N(x') = n - i$ (the probability is $p_n \frac{1}{2}$), the acceptance probability is $\frac{i}{2en}$, since $f^N(x') = f^N(x) + 1$; if $f^N(x') = n - i + 1$ or $f^N(x') = n - i + 2$ (the probability is $(1 - p_n) + p_n \frac{1}{2}$), the acceptance probability is $1$, since $f^N(x') > f^N(x) + 1$. If $f^N(x) = n - i$, the probability of which is $1 - p_n$, there are two cases for the acceptance of $x'$: if $f^N(x') = n - i + 1$ (the probability is $(1 - p_n)$), the acceptance probability is $\frac{i}{2en}$; if $f^N(x') = n - i + 2$ (the probability is $p_n \frac{1}{2}$), the acceptance probability is $1$. If $f^N(x) = n - i + 1$, the probability of which is $p_n \frac{1}{2}$, $x'$ will be accepted only if $f^N(x') = n - i + 2$ (the probability is $p_n \frac{1}{2}$), and the acceptance probability is $\frac{i}{2en}$.

Thus, we have

\[
p_{i,i-1} = P_1\left(p_n \frac{1}{2} \left(p_n \frac{1}{2} \cdot \frac{i}{2en} + (1 - p_n) + p_n \frac{1}{2}\right) + (1 - p_n) \left((1 - p_n) \cdot \frac{i}{2en} + p_n \frac{1}{2}\right) + p_n \frac{1}{2} \cdot p_n \frac{1}{2} \cdot \frac{i}{2en}\right) = \frac{1}{n}\left((1 - p_n)^{2-i} \cdot (p_n \frac{1}{2} \left((i - 2n + 1) + (1 - p_n)\right) + (1 - p_n)(1 - p_n) \frac{1}{2}\right).
\]

Our goal is to reach $i = 0$ (i.e., the global optimum). Starting from $i = 1$, $i$ will reach 0 in one step with probability

\[
p_{1,0} \geq \frac{1}{i} \cdot \frac{1}{2en} \cdot \left(\frac{p_n^2}{2} + (1 - p_n)^2\right) \geq \frac{1}{2en} \quad \text{(by } 0 \leq p_n \leq 1\text{)}.
\]

Thus, for reaching $i = 0$, we need to reach $i = 1$ for $O(n^2)$ times in expectation.

Then, we analyze the expected running time until $i = 1$. In this process, we can pessimistically assume that $i = 0$ will never be reached, because our final goal is to get the running time upper bound for reaching $i = 0$. For $2 \leq i \leq n - 1$, we have

\[
\frac{p_{i,i-1}}{p_{i,i+1}} \geq \frac{P_1 \cdot \left(p_n \frac{1}{2} \cdot p_n \frac{1}{2}\right)}{p_1 \cdot \left(p_n \frac{1}{2} \cdot p_n \frac{1}{2}\right) \cdot \frac{i}{2en}} \geq \frac{(1 - p_n)^{2-i} \cdot (p_n \frac{1}{2} \cdot p_n \frac{1}{2})}{n - i} \geq \frac{n}{n - i} > 1.
\]

Again, we can pessimistically assume that $p_{i,i-1} = p_{i,i+1}$ and $\forall d \geq 2, p_{i,i-d} = 0$, because we are to get the upper bound on the expected running time until $i = 1$. Then, we can view the evolutionary process for reaching $i = 1$ as a random walk on the path $\{1, 2, \ldots, n - 1, n\}$. We call a step that jumps to the neighbor state a relevant step. Thus,
by Lemma 17, it needs at most $2(n - 1)^2$ expected relevant steps to reach $i = 1$. Because the probability of a relevant step is at least

$$p_{i,i-1} \geq \frac{i}{en} \cdot \frac{i}{2en} \left((1 - p_n)^2 + p_n^2 \frac{1}{2}\right) \geq \frac{4}{2e^2n^2} \cdot \frac{1}{3},$$

the expected running time for a relevant step is $O(n^2)$. Then, the expected running time for reaching $i = 1$ is $O(n^4)$.

Thus, the expected running time of the whole optimization process is $O(n^6)$ for any $p_n \in [0, 1]$, and then the PNT is $[0, 1]$. □

Although we have shown that $(1) + (1 - \frac{1}{2en})$-smooth threshold selection can improve the PNT to be $[0, 1]$, the threshold value depends on the current search point. This implies that designing proper thresholds may require problem knowledge, which might be unrealistic. Thus, we are then to show that a smooth threshold without dependence on the current search point can also be effective. The proof of Theorem 13 is the same as that of Theorem 12, except that the acceptance probability for the fitness gap 1 is $\frac{1}{2en}$ instead of $\frac{1}{2en}$.

**Theorem 13:** For the $(1 + 1)$-EA with mutation probability $\frac{1}{n}$ on the OneMax problem under asymmetric one-bit noise, if using reevaluation with $(1) + (1 - \frac{1}{2en})$-smooth threshold selection, the PNT is $[0, 1]$.

We draw an intuitive understanding from the proof of Theorem 12 of why the smooth threshold selection can be better than the original ones. By changing the hard threshold to be smooth, it can make the probability of accepting a false better solution in one step small enough, i.e., $p_{i,i-1} \geq p_{i,i+1}$, and also make the probability of producing real progress in one step large enough, i.e., $p_{i,i-1}$ is not small.

## 5 Experiments

In this section we describe experiments done to complement the theoretical analyses. For any given configuration, we run the EA 1,000 times independently. In each run, we record the number of fitness evaluations until an optimal solution is found. Then the running time values of the 1,000 runs are averaged as the estimation of the expected running time, called the estimated ERT.

### 5.1 On Noise-Helpful Cases

We first present the experiment results on the Trap and Peak problems to verify Theorems 2 and 3.

For Theorem 2, we conduct experiments using the $(1 + n)$-EA, a specific case of the $(1 + \lambda)$-EA with $n$ being the dimensionality of the problem, on the Trap problem. We estimate the expected running time of the $(1 + n)$-EA starting from the solution $x$ with $|x|_0 = i$ for each $i$ ($0 \leq i \leq n$). Following Theorem 2, we compare the estimated ERT of the $(1 + n)$-EA without noise, with additive noise, and with multiplicative noise, respectively. For the mutation probability of the $(1 + n)$-EA, we use the common setting $p = \frac{1}{n}$. For additive noise, $\delta_1 = -n$ and $\delta_2 = n - 1$, and for multiplicative noise, $\delta_1 = 0.1$ and $\delta_2 = 10$. The results for $n = 5, 6, 7$ are plotted in Figure 1. We observe that the curves for these two kinds of noise are always below the curve without noise, which is consistent with our theoretical result. Note that the three curves meet at the first point, since the initial solution with $|x|_0 = 0$ is the optimal solution and then ERT = 1.
For Theorem 3, we conduct experiments using the $(1+1)$-EA* on the Peak problem. The one-bit noise is set with $p_n = 0.5$. The results for $n = 6, 7, 8$ are plotted in Figure 2. We observe that the curve with one-bit noise is always below the curve without noise when $|x_0|$ is large enough, which is consistent with the analysis result. We also run the $(1+1)$-EA on the Peak problem, the results of which are shown in Figure 3. The observation that the curve with noise is always below the one without noise agrees with Conjecture 1.

We have shown that noise can make deceptive and flat problems easier for EAs. For deceptive problems, the EA searches along the deceptive direction, so noise can
Analyzing Evolutionary Optimization in Noisy Environments

Figure 4: Estimated ERT comparison for the (1 + 1)-EA solving the MST problem with or without noise.

add some randomness to make the EA have some possibility of running along the right direction. For flat problems, the EA has no guided information for search, while in some situations noise can make the EA have a larger probability of running along the right direction than the wrong direction.

Note that though the Trap and Peak problems are, respectively, extremely deceptive and flat, in real applications we often encounter optimization problems with some degree of deceptiveness and flatness. We then test whether the finding on the extreme cases also holds on other problems. We employ the (1 + 1)-EA with mutation probability \( \frac{1}{n} \) on the minimum spanning tree (MST) problem. Given an undirected connected graph \( G = (V, E) \) on \( n \) vertices and \( m \) edges, the MST problem is to find a connected graph \( G' = (V, E' \subseteq E) \) with the minimal sum of edge weights. As in Neumann and Wegener (2007), a solution \( x \) is represented by a Boolean string of length \( m \), i.e., \( x \in \{0, 1\}^m \), where \( x_i = 1 \) means that the edge \( i \) is selected by \( x \); and the following fitness function is used for minimization,

\[
    f(x) = (c(x) - 1)w_{ub}^2 + \left( \sum_{i=1}^{m} x_i - n + 1 \right) w_{ub} + \sum_{i=1}^{m} x_i w_i,
\]

where \( c(x) \) is the number of connected components of the subgraph represented by \( x \), and \( w_{ub} = n^2 \cdot \max\{w_i \mid 1 \leq i \leq m\} \). It is easy to see that each nonminimum spanning tree is local optimal in the Hamming space; thus the MST problem is a multimodal problem with local deceptiveness.

We conduct experiments to compare the (1 + 1)-EA without noise and with one-bit noise (\( p_n = 0.5 \)) on the graphs with the number of edges \( m \in \Theta(n), \Theta(n\sqrt{n}) \), and \( \Theta(n^2) \) respectively. Let \( v_1, v_2, \ldots, v_n \) denote the \( n \) nodes.

Sparse graph. We use a cyclic graph where \( v_1 \) is connected with \( v_n \) and \( v_2, v_1 (1 < i < n) \) is connected with \( v_{i-1} \) and \( v_{i+1} \), and \( v_n \) is connected with \( v_{n-1} \) and \( v_1 \). Thus, \( m = n \).

Moderate graph. We use the graph where \( v_i \) is connected with \( v_{i+1}, v_{i+2}, \ldots, v_{i+\lfloor \sqrt{n} \rfloor} \) for \( 1 \leq i \leq n - \lfloor \sqrt{n} \rfloor \). Thus, \( m = (n - \lfloor \sqrt{n} \rfloor)\lfloor \sqrt{n} \rfloor \).

Dense graph. We use complete graph where each node is connected with all the other nodes. Thus, \( m = n(n - 1)/2 \).

For each type of graph and each independent run, the graph is constructed by setting the weight of each edge to be an integer randomly selected from \([1, n]\). The experiment results are plotted in Figure 4. We observe that the curves with one-bit noise appear below the curves without noise, which supports our finding that noise can make problems easier for EAs.
We have derived that in the deceptive and the flat cases noise can make the problem easier for EAs. Noticing that deceptiveness and flatness are two factors that can block the search of EAs, we hypothesize that the negative effect by noise decreases as the problem hardness increases, and noise will bring a positive effect when the problem is quite hard. The effect of noise can be measured by the estimated ERT gap,

$$\text{gap} = \frac{E[\tau] - E[\tau']}{E[\tau']}$$,

where $E[\tau]$ and $E[\tau']$ denote the expected running time of the EA optimizing the problem with and without noise, respectively. Note that the noise is harmful if the gap is positive and helpful if the gap is negative.

**Definition 11 (Jump $m,n$ Problem):** Jump $m,n$ problem of size $n$ with $1 \leq m \leq n$ is to find an $n$ bits binary string $x^*$ such that

$$x^* = \arg \max_{x \in \{0, 1\}^n} \left\{ \text{Jump}_{m,n}(x) = \begin{cases} m + \sum_{i=1}^{n} x_i & \text{if } \sum_{i=1}^{n} x_i \leq n - m \text{ or } = n \\ n - \sum_{i=1}^{n} x_i & \text{otherwise} \end{cases} \right\}.$$

To verify our hypothesis, we test the $(1 + 1)$-EA with mutation probability $\frac{1}{n}$ on the Jump $m,n$ problem as in Definition 11, as well as the MST problem.

The Jump $m,n$ problem has an adjustable difficulty and can be configured as the OneMax problem when $m = 1$ and the Trap problem when $m = n$. It is known that the expected running time of the $(1 + 1)$-EA on the Jump $m,n$ problem is $\Theta(n^m + n \log n)$ (Droste et al., 2002), which implies that the Jump $m,n$ problem with larger value of $m$ is harder. In the experiment, we set $n = 10$, and for noise, we use additive noise with $\delta_1 = -0.5n \land \delta_2 = 0.5n$, multiplicative noise with $\delta_1 = 1 \land \delta_2 = 2$, and one-bit noise with $p_n = 0.5$, respectively. The experiment results on gap values are plotted in Figure 5. We see that the gap values for larger $m$ are lower (i.e., the negative effect of noise decreases as the problem hardness increases).

The MST problem with sparse, moderate, and dense graphs is tested. The expected running time of the $(1 + 1)$-EA on the MST problem has been proved to be $\Theta(m^2 (\log n + \log w_{\max}))$ (Neumann and Wegener, 2007; Doerr et al., 2012b). With the assumption that this theoretical upper bound is tight, the hardness order of the three types of graphs is sparse $<$ moderate $<$ dense. The results are plotted in Figure 6. We observe that the height order of the curves is sparse $>$ moderate $>$ dense, which is consistent with our hypothesis.
Analyzing Evolutionary Optimization in Noisy Environments

Figure 6: Estimated ERT gap for the (1 + 1)-EA solving the MST problem with or without one-bit noise.

Figure 7: Estimated ERT comparison for the (1 + n)-EA on the OneMax problem with or without noise.

Experiments on both the artificial Jump\(m,n\) problem and the combinatorial MST problem reveal the same trend: the effect of the noise is related to the hardness of the problem to the EA. When the problem is hard, the noise is helpful to the EA, and thus noise handling is not necessary.

5.2 On Noise-Harmful Cases

We first verify the theoretical result that any noise will do harm to the OneMax problem. The experiment setting is the same as that for the (1 + \(\lambda\))-EA on the Trap problem (see Section 5.1). The results for \(n = 10, 20, 30\) are plotted in Figure 7. We observe that the curve for any noise is always above the curve without noise, which is consistent with our theoretical result.

Smooth threshold selection allows us to choose a fractional threshold, and through the running time analysis on the OneMax problem, we have shown that the fractional threshold is essential for the EA to keep efficient with noise. We then run the (1 + 1)-EA with mutation probability \(\frac{1}{n}\) on the OneMax problem under asymmetric one-bit noise. For the noise strength, we set \(p_n\) to be the maximum value 1. We test the smooth threshold values \((A) + (B)\) with \(A = 0, 1\) and \(B = 0, 0.1, \ldots, 0.9, 1\), which correspond to the threshold value set \(\{0, 0.1, 0.2, \ldots, 2\}\) on the \(x\)-axis. The results are plotted in Figure 8. Note that \(x = 0\) corresponds to the reevaluation strategy, and \(x = 1, 2\) corresponds to original threshold selection with \(\tau = x\). We observe that the curves reach the lowest point when \(x \approx 1.9\), which corresponds to a smooth threshold. We also tested the
C. Qian, Y. Yu, and Z.-H. Zhou

Figure 8: Estimated ERT for the (1 + 1)-EA with different threshold values solving the OneMax problem under asymmetric one-bit noise $p_n = 1$.

Figure 9: Estimated ERT for the (1 + 1)-EA with different threshold values solving the maximum matching problem under one-bit noise $p_n = 1$.

single-evaluation strategy, and the running time is empirically shown to be infinite. Thus, these empirical observations agree with our theoretical analyses.

To verify whether the fractional threshold is also useful in practice, we then carry out experiments to test the (1 + 1)-EA with mutation probability $\frac{1}{n}$ solving the maximum matching problem under one-bit noise. For the noise strength, we set $p_n = 1$. Given an undirected graph $G = (V, E)$ on $n$ vertices and $m$ edges, a matching is a subset $E'$ of the edge set $E$ such that no two edges in $E'$ share a common vertex. The maximum matching problem is to find a matching with the largest number of edges. As in Giel and Wegener (2003; 2006), a solution is represented as a Boolean string $x \in \{0, 1\}^m$, where $x_i = 1$ means that the edge $i$ is selected by $x$, and the following fitness function is used for maximization,

$$f(x) = \sum_{i=1}^{m} x_i - c \cdot \sum_{v \in V} p(v, x),$$

where $p(v, x) = \max[0, d(v, x) - 1]$, $d(v, x)$ is the degree of the vertex $v$ on the subgraph represented by $x$, and $c \geq m + 1$ is a penalty coefficient that makes any matching have a larger fitness than any nonmatching. Note that Qian et al. (2015b) found that a variable solution representation can be better than a Boolean string.

We test on the dense (complete) graph with the number of nodes $n = 7, 8, 9$. We test the smooth threshold values $(A) + (B)$ with $A = 0, 1, 2, 3$ and $B = 0, 0.1, \ldots, 0.9, 1$, which correspond to the threshold value set $\{0, 0.1, 0.2, \ldots, 4\}$ on the x-axis. The results are plotted in Figure 9. We observe that the curves reach the lowest point when $x$ is fractional between 1 and 2, which corresponds to a smooth threshold. The running time using the single-evaluation strategy is empirically shown to be infinite. These empirical
observations suggest that smooth threshold selection can lead to better performance in noisy environments.

6 Discussion and Conclusions

This work studies some theoretical issues of noisy optimization using EAs.

First, we proved that on deceptive and flat problems, noise can make the optimization easier for EAs. Experiments on the minimum spanning tree problem (a multimodal problem with local deceptiveness) support our theoretical findings. As deceptive and flat problems are EA-hard, while the noise can also be shown to be harmful on the EA-easy problem OneMax, we hypothesize that the negative effect of noise decreases as the problem hardness increases, and noise can even bring a positive effect when the problem is quite hard. This hypothesis is supported by experiments on the Jump\(_{m,n}\) problem and the minimum spanning tree problem, both of which have an adjustable difficulty parameter.

In problems where the noise has a negative effect, we studied the usefulness of two commonly employed noise-handling strategies: reevaluation and threshold selection. We took the OneMax problem as the representative problem, where the noise significantly harms the expected running time of the (1 + 1)-EA. We used the PNT as the performance measure, and analyzed the PNT of each EA under one-bit noise (see Table 1).

The reevaluation strategy seems to be a reasonable method for reducing random noise. However, we derived that the (1 + 1)-EA with single-evaluation (i.e., the (1 + 1)-EA without any noise-handling method) has the PNT \([0, 1 - \frac{1}{\Theta_1(\text{poly}(n))}]\) from Theorem 5, while the (1 + 1)-EA with reevaluation has the PNT \([0, \Theta(\log \frac{n}{n})]\). It is surprising to see that the reevaluation strategy leads to a much worse noise tolerance than that without any noise-handling method.

Reevaluation with threshold selection has a better PNT compared to reevaluation alone. When the threshold is 1, we derived the PNT \([0, 1]\) from Theorem 8, and when the threshold is 2, we obtained \([\frac{1}{\Theta_1(\text{poly}(n))}, 1 - \frac{1}{\Theta_1(\text{poly}(n))}]\) from Theorem 9. The improvement from reevaluation alone could be explained by the fact that threshold selection filters out false progress that is caused by noise. Furthermore, it shows an improvement from the (1 + 1)-EA without any noise-handling method when selecting the proper threshold \(\tau = 1\).

We also studied single-evaluation with threshold selection. The PNT is \([0, 0]\), which implies that threshold selection alone cannot help single-evaluation.

Finally, we analyzed the usefulness of these noise-handling strategies under a variant of one-bit noise. All of them are shown to be ineffective when the noise probability reaches the maximum value 1. We then proposed smooth threshold selection, which allows a fractional threshold to be effective. We proved that the (1 + 1)-EA with \((1 - \frac{1}{2n})\)-smooth threshold selection has the PNT \([0, 1]\) from Theorems 12 and 13, and found that the fractional threshold is essential to the proof. Our explanation is that like the original threshold selection, the proposed one filters out false progress while also retaining some chance of accepting true progress. We further carried out experiments to verify whether the smooth threshold could be helpful in practical problems. The experiments on the maximum matching problem show that the best performance can be achieved at fractional thresholds.

For analyzing the usefulness of noise-handling strategies, we studied a simplified noise model called one-bit noise. A direct generalization that will be studied in the future.
is to analyze bitwise noise, which flips each bit independently with some probability. Bitwise noise can change the solution greatly in evaluation and thus may make the analysis much more difficult. We shall also improve some currently derived running time bounds, for example, the current running time upper and lower bounds for single-evaluation still have a gap of $n$. To theoretically analyze the relation between the effect of noise and the hardness of optimization problems is also interesting future work.

**Acknowledgments**

The authors thank the associate editor and anonymous reviewers for helpful comments and suggestions. This research was supported by the National Science Foundation of China (61375061, 61333014), Jiangsu Science Foundation (BK2012303), Foundation for the Author of National Excellent Doctoral Dissertation of China (201451), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

**References**


C. Qian, Y. Yu, and Z.-H. Zhou


