Switch Analysis for Running Time Analysis of Evolutionary Algorithms

Yang Yu, *Member, IEEE,* Chao Qian, Zhi-Hua Zhou, *Fellow, IEEE*

***Abstract*—Evolutionary algorithms are a large family of heuristic optimization algorithms. They are problem independent, and have been applied in various optimization problems. Thus general analysis tools are especially appealing for guiding the analysis of evolutionary algorithms in various situations. This paper develops the *switch analysis* approach for running time analysis of evolutionary algorithms, revealing their average computational complexity. Unlike previous analysis approaches that analyze an algorithm from scratch, the switch analysis makes use of another well analyzed algorithm and, by contrasting them, can lead to better results. We investigate the power of switch analysis by comparing it with two commonly used analysis approaches, the *fitness level method* and the *drift analysis*. We define the *reducibility* between two analysis approaches for comparing their power. By the reducibility relationship, it is revealed that both the fitness level method and the drift analysis are *reducible* to the switch analysis, as they are equivalent to specific configurations of the switch analysis. We further show that the switch analysis is not reducible to the fitness level method, and compare it with the drift analysis on a concrete analysis case (the Discrete Linear Problem). The reducibility study might shed some light on the unified view of different running time analysis approaches.**

***Index Terms*—Evolutionary algorithms, running time complexity, analysis approaches, switch analysis**

I. Introduction

Evolutionary algorithms (EAs) [1] are a large family of general purpose randomized heuristic optimization algorithms, involving not only the algorithms originally inspired by the evolution process of natural species, i.e., genetic algorithms, evolutionary strategies and genetic programming, but also many other nature inspired heuristics such as simulated annealing and particle swarm optimization. In general, most EAs start with a random population of solutions, and then iteratively sample population of solutions, where the sampling depends only on the very previous population and thus satisfies the Markov property. In this paper, we study EAs with the Markov property.

As a general purpose technique, EAs are expected to be applied to solve various problems, even those that were never met before. This situation is different from the traditional mathematical programming and algorithm studies in which algorithms have bounded problem ranges, e.g. in convex optimization all problems are convex, and sorting algorithms apply on sorting problems. Therefore, to gain high confidence of applying EAs, we need evidence that EAs will work well in future problems. There have been many successful application cases, e.g., antenna design [17], circuit optimization [25], and scheduling [30]. These cases, however, serve more as intuitions from practice rather than rigorous evidences. Theoretical justifications to the effectiveness of EAs are, therefore, of great importance.

There has been a significant rise of theoretical studies on EAs in the recent decade. Increasing number of theoretical properties have being discovered, particularly on the running time, which is the average computation complexity of EAs and is thus a core theoretical issue. Probe problems (e.g. pseudo-Boolean linear problems [9]) are widely employed to facilitate the analysis on questions such as how efficient EAs can be and what parameters should be used. Interestingly, conflicting conclusions have been disclosed. For example, using crossover operators in EAs has been shown quite necessary in some problems (e.g. [8], [18], [28]), but is undesired in some other problems (e.g. [23]); using a large population can be helpful in some cases (e.g. [15]), and unhelpful in some other cases (e.g. [2]). These disclosures also imply the sophisticated situation we are facing with EAs. Because of the large variety of problems, general analysis tools are quite appealing, in order to guide the analysis of EAs on more problems rather than ad hoc analyses starting from scratch.

A few general analysis approaches have been developed, including the fitness level method [31] and the drift analysis [14]. Fitness level method divides the input space into levels, captures the transition probabilities between levels, and then bounds the expected running time from the transition probabilities. Drift analysis measures the progress of every step of an EA process[1], and then bounds its expected running time by dividing the total distance by the step size.

This work presents the *switch analysis* approach for running time analysis of EAs, extending largely our preliminary attempt [33]. Different from the existing

Manuscript received December 9, 2013.

All the authors are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mails: {yuy,qianc,zhouzh}@lamda.nju.edu.cn) (Zhi-Hua Zhou is the corresponding author)

[1]An *EA process* means the running of an EA on a problem instance.

approaches, switch analysis compares the expected running time of two EA processes. For analyzing a given EA on a given problem, switch analysis can be used to compare its expected running time with that of a reference EA process. The reference process can be particularly designed to be easily analyzable, so that the whole analysis can be simplified. An early form of switch analysis has been applied in the proof of the Theorem 4 of [11]. The switch analysis presented in this work is more general. We demonstrate the use of switch analysis by presenting a re-proof of the expected running time lower bound of any mutation-based EA on the Boolean function class with a unique global optimum, which extends our previous work [22] and has been partially proved in [27] by fitness level method and has been proved with stochastic dominance in [32] by drift analysis.

An interesting question is that how these general analysis approaches relate to each other. To investigate this question, we formally characterize an *analysis approach*, and define the *reducibility* between two approaches. Roughly speaking, an approach A is reducible to B if B can derive at least the same tight bound as A while requiring no more information, which implies that B is at least as powerful as A. We then prove that both the fitness level method and the drift analysis are reducible to the switch analysis. Meanwhile, we also find that switch analysis is not reducible to the fitness level method. We compare the switch analysis with the drift analysis on analyzing the (1+1)-EA solving the Discrete Linear Problem, where we also derived a new upper bound of its running time. These results not only disclose the power of the switch analysis, but also hint at a unified view of different running time analysis approaches.

The rest of this paper is organized into 7 sections. After the introduction of preliminaries in Section II, Section III presents the switch analysis. Section IV then demonstrates an application of switch analysis. Section V describes the formal characterization of analysis approaches and defines the reducibility relationship. The reducibility between switch analysis and fitness level method is studied in Section VI, and the reducibility between switch analysis and drift analysis is studied in Section VII. Finally, Section VIII concludes.

II. Preliminaries

A. Evolutionary Algorithms

Evolutionary algorithms (EAs) [1] simulate the natural evolution process by considering two key factors, variational reproduction and superior selection. They repeatedly reproduce solutions by varying currently maintained ones and eliminate inferior solutions, such that they improve the solutions iteratively. Although there exist many variants, the common procedure of EAs can be described as follows:

1. Generate an initial solution set (called population);
2. Reproduce new solutions from the current ones;
3. Evaluate the newly generated solutions;
4. Update the population by removing bad solutions;
5. Repeat steps 2-5 until some criterion is met.

In Algorithm 1, we describe the (1+1)-EA, which is a drastically simplified and deeply analyzed EA [9], [10]. It employs the population size 1, and uses mutation operator only.

Algorithm 1 ((1+1)-EA)
Given solution length n and pseudo-Boolean objective function f, (1+1)-EA maximizing f consists of the following steps:

1. $s :=$ *choose a solution from* $\mathcal{S} = \{0,1\}^n$ *uniformly at random.*
2. $s' :=$ *mutation*(s).
3. *If* $f(s') \geq f(s)$, $s := s'$.
4. *Terminate if* s *is optimal.*
5. *Goto step 2.*

where mutation$(\cdot) : \mathcal{S} \to \mathcal{S}$ *is a mutation operator.*

The mutation is commonly implemented by the one-bit mutation or the bit-wise mutation:
one-bit mutation for a solution, randomly choose one of the n bits, and flip (0 to 1 and vice versa) the chosen bit.
bit-wise mutation for a solution of length n, flip (0 to 1 and vice versa) each bit with probability $\frac{1}{n}$.
Note that the (1+1)-EA with one-bit mutation is usually called *randomized local search* (RLS). However, we still treat it as a specific EA in this paper for convenience.

B. Markov Chain Model

During the running of an EA, the offspring solutions are generated by varying the maintained solutions. Thus once the maintained solutions are given, the offspring solutions are drawn from a fixed distribution, regardless of how the maintained solutions are arrived at. This process is naturally modeled by a Markov chain, which has been widely used for the analysis of EAs [14], [34]. A Markov chain is a sequence of variables, $\{\xi_t\}_{t=0}^{+\infty}$, where the variable ξ_{t+1} depends only on the variable ξ_t, i.e., $P(\xi_{t+1} \mid \xi_t, \xi_{t-1}, \ldots, \xi_0) = P(\xi_{t+1} \mid \xi_t)$. Therefore, a Markov chain can be fully captured by the initial state ξ_0 and the transition probability $P(\xi_{t+1} \mid \xi_t)$.

Denote $\mathcal{S}$ as the solution space of a problem. An EA maintaining m solutions (i.e., the population size is m) has a search space $\mathcal{X} \subseteq \mathcal{S}^m$ (of which the exact size can be found in [29]). There are several possible ways of modeling the EAs as Markov chains. The most straightforward one might be taking $\mathcal{X}$ as the state space of the Markov chain, denoted as $\{\xi_t\}_{t=0}^{+\infty}$ where $\xi_t \in \mathcal{X}$. Let $\mathcal{X}^* \subset \mathcal{X}$ denote the optimal region, in which a population contains at least one optimal solution. It should be clear that a Markov chain models an EA process, i.e., the process of the running of an EA on a problem instance. In the rest of the paper, we will describe a Markov Chain $\{\xi_t\}_{t=0}^{+\infty}$ with state space $\mathcal{X}$ as "$\xi \in \mathcal{X}$" for simplicity.

The goal of the analysis is to disclose how soon the chain ξ (and thus the corresponding EA process) falls into $\mathcal{X}^*$ from some initial state. Particularly, we consider the performance measure *expected first hitting time* defined below:

Definition 1 (Conditional first hitting time, CFHT)
Given a Markov chain $\xi \in \mathcal{X}$ and a target subspace $\mathcal{X}^* \subset \mathcal{X}$, starting from time t_0 where $\xi_{t_0} = x$, let τ be a random variable that denotes the hitting events:

$$\tau = 0 : \xi_{t_0} \in \mathcal{X}^*,$$
$$\tau = i : \xi_{t_0+i} \in \mathcal{X}^* \wedge \xi_j \notin \mathcal{X}^* \ (j = t_0, \ldots, t_0 + i - 1).$$

The conditional expectation of τ,

$$\mathbb{E}[\![\tau \mid \xi_{t_0} = x]\!] = \sum\nolimits_{i=0}^{+\infty} i \cdot P(\tau = i),$$

is called the conditional first hitting time (CFHT) of the Markov chain from $t = t_0$ and $\xi_{t_0} = x$.

Definition 2 (Distribution-CFHT, DCFHT)
Given a Markov chain $\xi \in \mathcal{X}$ and a target subspace $\mathcal{X}^* \subset \mathcal{X}$, starting from time t_0 where ξ_{t_0} is drawn from a state distribution π, the expectation of the CFHT,

$$\mathbb{E}[\![\tau \mid \xi_{t_0} \sim \pi]\!] = \mathbb{E}_{x\sim\pi}[\![\tau \mid \xi_{t_0} = x]\!] = \sum\nolimits_{x\in\mathcal{X}} \pi(x)\mathbb{E}[\![\tau \mid \xi_{t_0} = x]\!],$$

is called the distribution-conditional first hitting time (DCFHT) of the Markov chain from $t = t_0$ and $\xi_{t_0} \sim \pi$.

Definition 3 (Expected first hitting time, EFHT)
Given a Markov chain $\xi \in \mathcal{X}$ and a target subspace $\mathcal{X}^* \subset \mathcal{X}$, the DCFHT of the chain from $t = 0$ and uniform distribution π_u,

$$\mathbb{E}[\![\tau]\!] = \mathbb{E}[\![\tau \mid \xi_0 \sim \pi_u]\!] = \mathbb{E}_{x\sim\pi_u}[\![\tau \mid \xi_0 = x]\!] = \sum\nolimits_{x\in\mathcal{X}} \mathbb{E}[\![\tau \mid \xi_0 = x]\!]/|\mathcal{X}|,$$

is called the expected first hitting time (EFHT) of the Markov chain.

The EFHT of an EA measures the average number of generations (iterations) that it takes to find an optimal solution. Within one generation, an EA takes time to manipulate and evaluate solutions that relate to the number of solutions it maintains. To reflect the computational time complexity of an EA, we count the number of evaluations to solutions, i.e., EFHT $\times$ the population size, which is called the *expected running time* of the EA.

We call a chain *absorbing* (with a slight abuse of the term) if all states in $\mathcal{X}^*$ are absorbing states.

Definition 4 (Absorbing Markov Chain)
Given a Markov chain $\xi \in \mathcal{X}$ and a target subspace $\mathcal{X}^* \subset \mathcal{X}$, ξ is said to be an absorbing chain, if

$$\forall x \in \mathcal{X}^*, \forall t \geq 0 : P(\xi_{t+1} \neq x \mid \xi_t = x) = 0.$$

Given a non-absorbing chain, we can construct a corresponding absorbing chain that simulates the non-absorbing chain but stays in the optimal state once it has been found. The EFHT of the constructed absorbing chain is the same as the EFHT of its corresponding non-absorbing chain. We then assume all chains considered in this paper are absorbing.

The following lemma on properties of Markov chains [20] (Theorem 1.3.5, page 17) will be used in this paper.

Lemma 1
Given an absorbing Markov chain $\xi \in \mathcal{X}$ and a target subspace $\mathcal{X}^* \subset \mathcal{X}$, we have about CFHT that $\mathbb{E}[\![\tau | \xi_t \in \mathcal{X}^*]\!] = 0$,

$$\forall x \notin \mathcal{X}^* : \mathbb{E}[\![\tau \mid \xi_t = x]\!] = 1 + \sum\nolimits_{y\in\mathcal{X}} P(\xi_{t+1} = y \mid \xi_t = x)\mathbb{E}[\![\tau \mid \xi_{t+1} = y]\!],$$

and about DCFHT that,

$$\begin{aligned}\mathbb{E}[\![\tau \mid \xi_t \sim \pi_t]\!] &= \mathbb{E}_{x\sim\pi_t}[\![\tau \mid \xi_t = x]\!] \\ &= 1 - \pi_t(\mathcal{X}^*) + \sum_{x\in\mathcal{X}-\mathcal{X}^*, y\in\mathcal{X}} \pi_t(x)P(\xi_{t+1}=y|\xi_t=x)\mathbb{E}[\![\tau|\xi_{t+1}=y]\!] \\ &= 1 - \pi_t(\mathcal{X}^*) + \mathbb{E}[\![\tau \mid \xi_{t+1} \sim \pi_{t+1}]\!],\end{aligned}$$

where $\pi_{t+1}(x) = \sum_{y\in\mathcal{X}} \pi_t(y)P(\xi_{t+1} = x \mid \xi_t = y)$.

Note that the first two "$y \in \mathcal{X}$" in Lemma 1 can be replaced by "$y \in \mathcal{X} - \mathcal{X}^*$" as in the book [20], since $\mathbb{E}[\![\tau|\xi_t \in \mathcal{X}^*]\!] = 0$; and "$x \in \mathcal{X} - \mathcal{X}^*$" can be replaced by $x \in \mathcal{X}$, since $P(\xi_{t+1} \in \mathcal{X} - \mathcal{X}^*|\xi_t \in \mathcal{X}^*) = 0$.

III. Switch Analysis

Given two Markov chains ξ and ξ', let τ and τ' denote the hitting events of the two chains, respectively. We present the switch analysis in Theorem 1 that compares the DCFHT of the two chains, i.e., $\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!]$ and $\mathbb{E}[\![\tau' \mid \xi'_0 \sim \pi_0^\phi]\!]$, where π_0 and π_0^ϕ are their initial state distribution. Since we are dealing with two chains, which may have different state spaces, we utilize aligned mappings as in Definition 5.

Definition 5 (Aligned Mapping)
Given two spaces $\mathcal{X}$ and $\mathcal{Y}$ with target subspaces $\mathcal{X}^*$ and $\mathcal{Y}^*$, respectively, a function $\phi : \mathcal{X} \to \mathcal{Y}$ is called
(a) a left-aligned mapping *if* $\forall x \in \mathcal{X}^* : \phi(x) \in \mathcal{Y}^*$;
(b) a right-aligned mapping *if* $\forall x \in \mathcal{X} - \mathcal{X}^* : \phi(x) \notin \mathcal{Y}^*$;
(c) an optimal-aligned mapping *if it is both left-aligned and right-aligned.*

Note that the function $\phi : \mathcal{X} \to \mathcal{Y}$ implies that for all $x \in \mathcal{X}$ there exists one and only one $y \in \mathcal{Y}$ such that $\phi(x) = y$, but it may not be an injective or surjective mapping. To simplify the notation, we denote the mapping $\phi^{-1}(y) = \{x \in \mathcal{X} \mid \phi(x) = y\}$ as the *inverse solution set* of the function. Note that $\phi^{-1}(y)$ can be the empty set for some $y \in \mathcal{Y}$. We also extend the notation of ϕ to have set input, i.e., $\phi(X) = \cup_{x\in X}\{\phi(x)\}$ for any set $X \subseteq \mathcal{X}$ and $\phi^{-1}(Y) = \cup_{y\in Y}\phi^{-1}(y)$ for any set $Y \subseteq \mathcal{Y}$. By the set extension, we have that, if ϕ is a left-aligned mapping, $\mathcal{X}^* \subseteq \phi^{-1}(\mathcal{Y}^*)$; if ϕ is a right-aligned mapping, $\phi^{-1}(\mathcal{Y}^*) \subseteq \mathcal{X}^*$; and if ϕ is an optimal-aligned mapping, $\mathcal{X}^* = \phi^{-1}(\mathcal{Y}^*)$.

The main theoretical result is presented in Theorem 1. The idea is that, if we can bound the difference of the two chains on the one-step change of the DCFHT, we

can obtain the difference of their DCFHT by summing up all the one-step differences. Following the idea, we find that the calculation of the one-step difference can be drastically simplified: the one-step transitions of the two chains under the same distribution of one chain (i.e., π_t in Eq.(1)) and on the same ground of CFHT of the other chain (i.e., $\mathbb{E}[\![\tau']\!]$ in Eq.(1)). The one-step differences, ρ_t, are then summed up to bound the difference of their DCFHT. Note that the right (or left)-aligned mapping is used to allow the two chains to have different state spaces.

Theorem 1 (Switch Analysis)
Given two absorbing Markov chains $\xi \in \mathcal{X}$ and $\xi' \in \mathcal{Y}$, let τ and τ' denote the hitting events of ξ and ξ', respectively, and let π_t denote the distribution of ξ_t. Given a series of values $\{\rho_t \in \mathbb{R}\}_{t=0}^{+\infty}$ with $\rho = \sum_{t=0}^{+\infty} \rho_t$ and a right (or left)-aligned mapping $\phi : \mathcal{X} \to \mathcal{Y}$, if $\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!]$ is finite and

$$\begin{aligned}\forall t : &\sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_t(x)P(\xi_{t+1} \in \phi^{-1}(y) \mid \xi_t = x)\mathbb{E}[\![\tau' \mid \xi'_0 = y]\!] \\ &\leq (\textit{or } \geq) \sum_{u,y\in\mathcal{Y}} \pi_t^{\phi}(u)P(\xi'_1 = y \mid \xi'_0 = u)\mathbb{E}[\![\tau' \mid \xi'_1 = y]\!] \\ &\quad + \rho_t, \end{aligned} \tag{1}$$

where $\pi_t^{\phi}(y) = \pi_t(\phi^{-1}(y)) = \sum_{x\in\phi^{-1}(y)} \pi_t(x)$, we have

$$\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!] \leq (\textit{or } \geq)\mathbb{E}[\![\tau' \mid \xi'_0 \sim \pi_0^{\phi}]\!] + \rho.$$

For proving the theorem, we define the intermediate Markov chain ξ^k for $k \in \{0, 1, \cdots\}$. Denote the one-step transition of ξ as tr, and the one-step transition of ξ' as tr', ξ^k is a Markov chain that

1) is initially in the state space $\mathcal{X}$ and has the same initial state distribution as ξ, i.e., $\pi_0^k = \pi_0$;
2) uses transition tr at time $\{0, 1, \ldots, k-1\}$ if $k > 0$, i.e., it is identical to the chain ξ at the first k steps;
3) switches to the state space $\mathcal{Y}$ at time k, which is by mapping the distribution π_k of states over $\mathcal{X}$ to the distribution π_k^{ϕ} of states over $\mathcal{Y}$ via ϕ;
4) uses transition tr' from time k, i.e., it then acts like the chain ξ' from time 0.

For the intermediate Markov chain ξ^k, its first hitting event τ^k is counted as $\xi_t^k \in \mathcal{X}^*$ for $t = 0, 1, \ldots, k-1$ and as $\xi_t^k \in \mathcal{Y}^*$ for $t \geq k$. Therefore the first hitting event of ξ^0 is the same as ξ' and ξ^{∞} is the same as ξ.

Lemma 2
Given two absorbing Markov chains $\xi \in \mathcal{X}$ and $\xi' \in \mathcal{Y}$, and a right-aligned mapping $\phi : \mathcal{X} \to \mathcal{Y}$, let τ and τ' denote the hitting events of ξ and ξ', respectively, and let π_t denote the distribution of ξ_t, we have for the hitting events τ^k of the intermediate chain ξ^k with any $k \in \{0, 1, \cdots\}$ that

$$\begin{aligned}\mathbb{E}[\![\tau^k \mid \xi_0^k \sim \pi_0]\!] &= k - \sum\nolimits_{t=0}^{k-1} \pi_t(\mathcal{X}^*) \\ &\quad + \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_{k-1}(x)P(\xi_k \in \phi^{-1}(y) | \xi_{k-1} = x)\mathbb{E}[\![\tau' | \xi'_0 = y]\!] \\ &\quad - \sum_{x\in\mathcal{X}^*,y\in\mathcal{Y}} \pi_{k-1}(x)P(\xi_k \in \phi^{-1}(y) | \xi_{k-1} = x)\mathbb{E}[\![\tau' | \xi'_0 = y]\!].\end{aligned}$$

Proof. Let π^k denote the distribution of ξ^k. For the chain ξ^k at time $k-1$, since it will be mapped into the space $\mathcal{Y}$ from time k via ϕ, by Lemma 1 we have

$$\begin{aligned}\mathbb{E}[\![\tau^k \mid \xi_{k-1}^k \sim \pi_{k-1}^k]\!] &= 1 - \pi_{k-1}^k(\mathcal{X}^*) \\ &\quad + \sum_{x\in\mathcal{X}-\mathcal{X}^*,y\in\mathcal{Y}} \pi_{k-1}^k(x)P(\xi_k^k = y | \xi_{k-1}^k = x)\mathbb{E}[\![\tau^k | \xi_k^k = y]\!].\end{aligned}$$

The chain ξ^k at time $k-1$ acts like the chain ξ, thus $P(\xi_k^k = y | \xi_{k-1}^k = x) = P(\xi_k \in \phi^{-1}(y) | \xi_{k-1} = x)$. It acts like the chain ξ' from time k, thus $\mathbb{E}[\![\tau^k | \xi_k^k = y]\!] = \mathbb{E}[\![\tau' | \xi'_0 = y]\!]$. We then have

$$\begin{aligned}\mathbb{E}[\![\tau^k \mid \xi_{k-1}^k \sim \pi_{k-1}^k]\!] &= 1 - \pi_{k-1}^k(\mathcal{X}^*) \\ &\quad + \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_{k-1}^k(x)P(\xi_k \in \phi^{-1}(y) | \xi_{k-1} = x)\mathbb{E}[\![\tau' | \xi'_0 = y]\!] \\ &\quad - \sum_{x\in\mathcal{X}^*,y\in\mathcal{Y}} \pi_{k-1}^k(x)P(\xi_k \in \phi^{-1}(y) | \xi_{k-1} = x)\mathbb{E}[\![\tau' | \xi'_0 = y]\!].\end{aligned} \tag{2}$$

Note that the last minus term of Eq.(2) is necessary, because that if $\xi_{k-1}^k \in \mathcal{X}^*$ the chain should stop running, but the right-aligned mapping may map states in $\mathcal{X}^*$ to $\mathcal{Y} - \mathcal{Y}^*$ and continue running the chain ξ', which is excluded by the last minus term.

By Lemma 1 we have that

$$\begin{aligned}&\mathbb{E}[\![\tau^k \mid \xi_0^k \sim \pi_0^k]\!] = 1 - \pi_0^k(\mathcal{X}^*) + \mathbb{E}[\![\tau^k \mid \xi_1^k \sim \pi_1^k]\!] \\ &= \cdots \\ &= (k-1) - \sum\nolimits_{t=0}^{k-2} \pi_t^k(\mathcal{X}^*) + \mathbb{E}[\![\tau^k \mid \xi_{k-1}^k \sim \pi_{k-1}^k]\!].\end{aligned}$$

Applying Eq.(2) to the last term, results in that

$$\begin{aligned}\mathbb{E}[\![\tau^k \mid \xi_0^k \sim \pi_0^k]\!] &= k - \sum\nolimits_{t=0}^{k-1} \pi_t^k(\mathcal{X}^*) \\ &\quad + \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_{k-1}^k(x)P(\xi_k \in \phi^{-1}(y) | \xi_{k-1} = x)\mathbb{E}[\![\tau' | \xi'_0 = y]\!] \\ &\quad - \sum_{x\in\mathcal{X}^*,y\in\mathcal{Y}} \pi_{k-1}^k(x)P(\xi_k \in \phi^{-1}(y) | \xi_{k-1} = x)\mathbb{E}[\![\tau' | \xi'_0 = y]\!].\end{aligned}$$

For any $t < k$, since the chain ξ^k and ξ are identical before the time k, we have $\pi_t^k = \pi_t$, applying which obtains the lemma. □

Proof of Theorem 1 ("$\leq$" case).

Firstly we prove the "$\leq$" case which requires a right-aligned mapping.

For any $t < k$, since the chain ξ^k and ξ are identical before the time k, we have $\pi_t = \pi_t^k$, and thus

$$\forall t < k : \pi_t^k(\mathcal{X}^*) = \pi_t(\mathcal{X}^*) \geq \pi_t(\phi^{-1}(\mathcal{Y}^*)) = \pi_t^{\phi}(\mathcal{Y}^*), \tag{3}$$

since ϕ is right-aligned and thus $\phi^{-1}(\mathcal{Y}^*) \subseteq \mathcal{X}^*$.

We prove the theorem by induction on the k of the intermediate Markov chain ξ^k.

(a) Initialization is to prove the case $k = 0$, which is trivial since $\xi^0 = \xi'$ and thus $\mathbb{E}[\![\tau^0 \mid \xi_0^0 \sim \pi_0]\!] = \mathbb{E}[\![\tau' \mid \xi_0' \sim \pi_0^\phi]\!]$.

(b) Inductive Hypothesis assumes that for all $k \leq K-1$ ($K \geq 1$),

$$\mathbb{E}[\![\tau^k \mid \xi_0^k \sim \pi_0]\!] \leq \mathbb{E}[\![\tau' \mid \xi_0' \sim \pi_0^\phi]\!] + \sum\nolimits_{t=0}^{k-1} \rho_t,$$

we are going to prove

$$\mathbb{E}[\![\tau^K \mid \xi_0^K \sim \pi_0]\!] \leq \mathbb{E}[\![\tau' \mid \xi_0' \sim \pi_0^\phi]\!] + \sum\nolimits_{t=0}^{K-1} \rho_t. \quad (4)$$

Applying Lemma 2,

$$\begin{aligned}
&\mathbb{E}[\![\tau^K \mid \xi_0^K \sim \pi_0]\!] = K - \sum\nolimits_{t=0}^{K-1} \pi_t(\mathcal{X}^*) \\
&+ \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_{K-1}(x) P(\xi_K \in \phi^{-1}(y) | \xi_{K-1} = x) \mathbb{E}[\![\tau' | \xi_0' = y]\!] \\
&- \sum_{x\in\mathcal{X}^*,y\in\mathcal{Y}} \pi_{K-1}(x) P(\xi_K \in \phi^{-1}(y) | \xi_{K-1} = x) \mathbb{E}[\![\tau' | \xi_0' = y]\!]
\end{aligned}$$

we denote $\Delta(K) = \sum_{x\in\mathcal{X}^*,y\in\mathcal{Y}} \pi_{K-1}(x) P(\xi_K \in \phi^{-1}(y) | \xi_{K-1} = x) \mathbb{E}[\![\tau' | \xi_0' = y]\!]$, the derivation continues as

$$\begin{aligned}
&\leq K - \sum\nolimits_{t=0}^{K-1} \pi_t(\mathcal{X}^*) + \rho_{K-1} - \Delta(K) \\
&+ \sum_{u,y\in\mathcal{Y}} \pi_{K-1}^\phi(u) P(\xi_1' = y \mid \xi_0' = u) \mathbb{E}[\![\tau' \mid \xi_1' = y]\!] \\
&\leq K - \sum\nolimits_{t=0}^{K-2} \pi_t(\mathcal{X}^*) - \pi_{K-1}^\phi(\mathcal{Y}^*) + \rho_{K-1} - \Delta(K) \\
&+ \sum_{u,y\in\mathcal{Y}} \pi_{K-1}^\phi(u) P(\xi_1' = y \mid \xi_0' = u) \mathbb{E}[\![\tau' \mid \xi_1' = y]\!],
\end{aligned}$$

where the 1st inequality is by Eq.(1) and the 2nd inequality is by Eq.(3). Meanwhile, by Lemma 2 we have

$$\begin{aligned}
&\mathbb{E}[\![\tau^{K-1} \mid \xi_0^{K-1} \sim \pi_0]\!] \\
&= (K-1) - \sum\nolimits_{t=0}^{K-2} \pi_t(\mathcal{X}^*) - \Delta(K-1) \\
&+ \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_{K-2}(x) P(\xi_{K-1} \in \phi^{-1}(y) | \xi_{K-2} = x) \mathbb{E}[\![\tau' | \xi_0' = y]\!] \\
&= (K-1) - \sum\nolimits_{t=0}^{K-2} \pi_t(\mathcal{X}^*) - \Delta(K-1) \\
&+ \sum_{y\in\mathcal{Y}} \pi_{K-1}^\phi(y) \mathbb{E}[\![\tau' | \xi_0' = y]\!] \\
&= K - \sum\nolimits_{t=0}^{K-2} \pi_t(\mathcal{X}^*) - \pi_{K-1}^\phi(\mathcal{Y}^*) - \Delta(K-1) \\
&+ \sum_{u,y\in\mathcal{Y}} \pi_{K-1}^\phi(u) P(\xi_1' = y \mid \xi_0' = u) \mathbb{E}[\![\tau' | \xi_1' = y]\!],
\end{aligned}$$

where the last two equalities are by Lemma 1. Substituting this equation into the above inequality obtains

$$\begin{aligned}
&\mathbb{E}[\![\tau^K \mid \xi_0^K \sim \pi_0]\!] \\
&\leq \mathbb{E}[\![\tau^{K-1} \mid \xi_0^{K-1} \sim \pi_0]\!] + \rho_{K-1} + \Delta(K-1) - \Delta(K) \\
&\leq \mathbb{E}[\![\tau' \mid \xi_0' \sim \pi_0^\phi]\!] + \sum\nolimits_{t=0}^{K-1} \rho_t + \Delta(K-1) - \Delta(K), \quad (5)
\end{aligned}$$

where the last inequality is by the inductive hypothesis. On $\Delta(K-1) - \Delta(K)$, note our definition of absorption that $\forall x \in \mathcal{X}^* : P(\xi_{t+1} \neq x | \xi_t = x) = 0$, we have

$$\forall x \in \mathcal{X}^* : \pi_{K-1}(x) \geq \pi_{K-2}(x),$$

and

$$\Delta(K) = \sum\nolimits_{x\in\mathcal{X}^*} \pi_{K-1}(x) \mathbb{E}[\![\tau' | \xi_0' = \phi(x)]\!].$$

So we have

$$\begin{aligned}
\Delta(K-1) - \Delta(K) &= \sum_{x\in\mathcal{X}^*} \pi_{K-2}(x) \mathbb{E}[\![\tau' | \xi_0' = \phi(x)]\!] \\
&\quad - \sum_{x\in\mathcal{X}^*} \pi_{K-1}(x) \mathbb{E}[\![\tau' | \xi_0' = \phi(x)]\!] \\
&\leq 0
\end{aligned}$$

So that Eq.(5) results in Eq.(4).

(c) Conclusion from (a) and (b), it holds

$$\mathbb{E}[\![\tau^\infty \mid \xi_0^\infty \sim \pi_0]\!] \leq \mathbb{E}[\![\tau' \mid \xi_0' \sim \pi_0^\phi]\!] + \sum\nolimits_{t=0}^{+\infty} \rho_t.$$

Since $\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!]$ is finite, $\mathbb{E}[\![\tau^\infty \mid \xi_0^\infty \sim \pi_0]\!] = \mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!]$. Finally, by $\rho = \sum_{t=0}^{+\infty} \rho_t$, we get

$$\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!] \leq \mathbb{E}[\![\tau' \mid \xi_0' \sim \pi_0^\phi]\!] + \rho.$$

□

Proof of Theorem 1 ("≥" case).

The "≥" case requires a left-aligned mapping. Its proof is similar to that of the "≤" case, and is easier since the last minus term of Eq.(2) is zero.

Since ϕ is left-aligned and thus $\mathcal{X}^* \subseteq \phi^{-1}(\mathcal{Y}^*)$, we have that $\pi_t(\mathcal{X}^*) \leq \pi_t(\phi^{-1}(\mathcal{Y}^*)) = \pi_t^\phi(\mathcal{Y}^*)$.

The theorem is again proved by induction. The initialization is the same as for the "≤" case. The inductive hypothesis assumes that for all $k \leq K-1$ ($K \geq 1$),

$$\mathbb{E}[\![\tau^k \mid \xi_0^k \sim \pi_0]\!] \geq \mathbb{E}[\![\tau' \mid \xi_0' \sim \pi_0^\phi]\!] + \sum\nolimits_{t=0}^{k-1} \rho_t.$$

Applying Lemma 2,

$$\begin{aligned}
&\mathbb{E}[\![\tau^K \mid \xi_0^K \sim \pi_0]\!] = K - \sum\nolimits_{t=0}^{K-1} \pi_t(\mathcal{X}^*) \\
&+ \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_{K-1}(x) P(\xi_K \in \phi^{-1}(y) | \xi_{K-1} = x) \mathbb{E}[\![\tau' | \xi_0' = y]\!]
\end{aligned}$$

by Eq.(1) with "≥" and $\pi_{K-1}(\mathcal{X}^*) \leq \pi_{K-1}^\phi(\mathcal{Y}^*)$,

$$\begin{aligned}
&\geq K - \sum\nolimits_{t=0}^{K-2} \pi_t(\mathcal{X}^*) - \pi_{K-1}^\phi(\mathcal{Y}^*) + \rho_{K-1} \\
&+ \sum_{u,y\in\mathcal{Y}} \pi_{K-1}^\phi(u) P(\xi_1' = y \mid \xi_0' = u) \mathbb{E}[\![\tau' \mid \xi_1' = y]\!]
\end{aligned}$$

Meanwhile, by Lemma 1 and Lemma 2, we also have

$$\begin{aligned}
&\mathbb{E}[\![\tau^{K-1} \mid \xi_0^{K-1} \sim \pi_0]\!] = K - \sum_{t=0}^{K-2} \pi_t(\mathcal{X}^*) - \pi_{K-1}^\phi(\mathcal{Y}^*) \\
&+ \sum_{u,y\in\mathcal{Y}} \pi_{K-1}^\phi(u) P(\xi_1' = y \mid \xi_0' = u) \mathbb{E}[\![\tau' | \xi_1' = y]\!].
\end{aligned}$$

Therefore,

$$\mathbb{E}[\![\tau^K \mid \xi_0^K \sim \pi_0]\!] \geq \mathbb{E}[\![\tau^{K-1} \mid \xi_0^{K-1} \sim \pi_0]\!] + \rho_{K-1}$$

$$\geq \mathbb{E}[\![\tau' \mid \xi'_0 \sim \pi_0^{\phi}]\!] + \sum_{t=0}^{K-1} \rho_t,$$

which proves the induction. Therefore, we come to the conclusion that the theorem holds, using the argument of the "$\leq$" case. □

Though the theorem is proved by treating the state spaces $\mathcal{X}$ and $\mathcal{Y}$ as discrete spaces, we can show that the theorem still holds if $\mathcal{X}$ or (and) $\mathcal{Y}$ is continuous, by replacing the sum over the state space with the integral, which does not affect the inductive proof. We only present the discrete space version since this paper studies discrete optimizations.

Using the theorem to compare two chains, we can waive the long-term behavior of one chain, since Eq.(1) does not involve the term $\mathbb{E}[\![\tau \mid \xi_0 = y]\!]$. Therefore, the theorem can simplify the analysis of an EA by comparing it with an easy-to-analyze one.

When the Markov chain $\xi' \in \mathcal{Y}$ is homogeneous, i.e., the transition is static regardless of the time, we can have a compact expression of the switch analysis theorem, rewriting Eq.(1) as

$$\forall t : \rho_t \geq (\text{or } \leq) \sum_{y \in \mathcal{Y}} \mathbb{E}[\![\tau' \mid \xi'_0 = y]\!] \cdot \big(P(\phi(\xi_{t+1}) = y \mid \xi_t \sim \pi_t) - P(\xi'_{t+1} = y \mid \xi'_t \sim \pi_t^{\phi})\big).$$

By this expression, we can interpret that ρ_t bounds the sum of the weighted distribution difference of the two intermediate chains ξ^{t+1} and ξ^t at time $t+1$, where the weight is given by the CFHT of the chain ξ'. Because ξ^{t+1} and ξ^t are different only at time t, the ρ_t actually bounds the difference of using transition tr and tr' at time t. Thus, the difference of the DCFHT of the original two chains ξ and ξ' (i.e., ρ) is the sum of the difference of using tr and tr' at each time (i.e., $\sum_{t=0}^{+\infty} \rho_t$).

IV. Switch Analysis for Class-wise Analysis

This section gives an example of applying switch analysis. The *mutation-based EA* in Algorithm 2 is a general scheme of mutation-based EAs. It abstracts a general population-based EA which only employs mutation operator, including many variants of EAs with parent and offspring populations as well as parallel EAs as introduced in [27], [32]. The UBoolean in Definition 6 is a wide class of nontrivial pseudo-Boolean functions.

In the following, we give a re-proof using the switch analysis that the expected running time of any mutation-based EA with mutation probability $p \in (0, 0.5)$ on UBoolean function class (Definition 6) is at least as large as that of (1+1)-EA$_u$ (Algorithm 3) on the OneMax problem (Definition 7). Doerr et al. [5] first proved that the expected running time of (1+1)-EA with mutation probability $\frac{1}{n}$ on UBoolean is at least as large as that on OneMax. Later, this result was extended to arbitrary mutation-based EA with mutation probability $\frac{1}{n}$ in [27] by using fitness level method and (1+1)-EA with arbitrary mutation probability $p \in (0, 0.5)$ in [22] by using our early version of switch analysis. Our reproved result here combines these two generalizations. Recently, Witt [32] proved the same result with stochastic dominance by using drift analysis.

A. Definitions

Algorithm 2 (Scheme of a mutation-based EA)
Given solution length n and objective function f, a mutation-based EA consists of the following steps:

1. *choose μ solutions $s_1, \ldots, s_\mu \in \{0,1\}^n$ uniformly at random.*
   let $t := \mu$, and select a parent s from $\{s_1, \ldots, s_t\}$ according to t and $f(s_1), \ldots, f(s_t)$.
2. *$s_{t+1} :=$ Mutation(s).*
3. *select a parent s from $\{s_1, \ldots, s_{t+1}\}$ according to $t+1$ and $f(s_1), \ldots, f(s_{t+1})$.*
4. *terminates until some criterion is met.*
5. *let $t \leftarrow t+1$, Goto step 2.*

Algorithm 3 ((1+1)-EA$_\mu$)
Given solution length n and objective function f, the (1+1)-EA$_\mu$ consists of the following steps:

1. *choose μ solutions $s_1, \ldots, s_\mu \in \{0,1\}^n$ uniformly at random.*
   $s :=$ the best one among $s_1, \ldots, s_\mu$.
2. *$s' :=$ Mutation(s).*
3. *if $f(s') \geq f(s)$ $\quad s := s'$.*
4. *terminates until some criterion is met.*
5. *goto step 2.*

Definition 6 (UBoolean Function Class)
A function $f : \{0,1\}^n \to \mathbb{R}$ in UBoolean satisfies that

$$\exists s \in \{0,1\}^n, \forall s' \in \{0,1\}^n - \{s\}, f(s') < f(s).$$

For any function in UBoolean, we assume without loss of generality that the optimal solution is $11\ldots1$ (briefly denoted as 1^n). This is because EAs treat the bits 0 and 1 symmetrically, and thus the 0 bits in an optimal solution can be interpreted as 1 bits without affecting the behavior of EAs.

The OneMax problem in Definition 7 is a particular instance of UBoolean, which requires to maximize the number of 1 bits of a solution. It has been proved [10] that the expected running time of (1+1)-EA on OneMax is $\Theta(n \ln n)$.

Definition 7 (OneMax Problem)
OneMax Problem of size n is to find an n bits binary string s^* such that

$$s^* = \arg\max_{s \in \{0,1\}^n} \sum_{i=1}^{n} s_i,$$

where s_i is the i-th bit of solution $s \in \{0,1\}^n$.

B. Analysis

Before the proof, we give some lemmas which will be used in the following analysis. Since the bits of OneMax problem are independent and their weights are the same, it is not hard to see that the CFHT $\mathbb{E}[\![\tau' \mid \xi'_t = x]\!]$ of (1+1)-EA on OneMax only depends on the number

of 0 bits of the solution x, i.e., $|x|_0$. Thus, we denote $\mathbb{E}(j)$ as the CFHT $\mathbb{E}[\![\tau' \mid \xi'_t = x]\!]$ with $|x|_0 = j$. Then, it is obvious that $\mathbb{E}(0) = 0$, which implies the optimal solution. Lemma 3 (from [22]) gives the order on $\mathbb{E}(j)$, which discloses that $\mathbb{E}(j)$ increases with j. Lemma 4 (from [32]) says that it is more likely that the offspring generated by mutating a parent solution with less 0 bits has smaller number of 0 bits. Note that we consider $|\cdot|_0$ instead of $|\cdot|_1$ in their original lemma. It obviously still holds due to the symmetry.

Lemma 3 ([22])
For any mutation probability $0 < p < 0.5$, it holds that

$$\mathbb{E}(0) < \mathbb{E}(1) < \mathbb{E}(2) < \ldots < \mathbb{E}(n).$$

Lemma 4 ([32])
Let $a, b \in \{0,1\}^n$ be two search points satisfying $|a|_0 < |b|_0$. Denote by $mut(x)$ the random string obtained by mutating each bit of x independently with probability p. Let j be an arbitrary integer in $[0, n]$. If $p \leq 0.5$ then

$$P(|mut(a)|_0 \leq j) \geq P(|mut(b)|_0 \leq j).$$

We will also need an inequality in Lemma 5 that, given two random variables, when the cumulative distribution of one is always smaller than the other's, the expectation with ordered events of the former is larger.

Lemma 5
Let m $(m \geq 1)$ be an integer. If two distributions P and Q over $\{0, 1, \ldots, m\}$ (i.e., for any $i = 0, \ldots, m$, P_i and Q_i are non-negative, and the sum of each is 1) satisfy that

$$\forall 0 \leq k \leq m-1, \sum_{i=0}^{k} P_i \leq \sum_{i=0}^{k} Q_i,$$

then for any $0 \leq E_0 < E_1 < \ldots < E_m$ 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 E_i is increasing, f is Schur-concave by Theorem A.3 in Chapter 3 of [19]. The condition implies that the distribution $(Q_0, \ldots, Q_m)$ 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. □

Theorem 2
The expected running time of any mutation-based EA with μ initial solutions and any mutation probability $p \in (0, 0.5)$ on UBoolean is at least as large as that of (1+1)-EA$_\mu$ with the same p on the OneMax problem.

Proof. We construct a history-encoded Markov chain to model the mutation-based EAs as in Algorithm 2. Let $\mathcal{X} = \{(s_1, \ldots, s_t) \mid s_j \in \{0,1\}^n, t \geq \mu\}$, where $(s_1, \ldots, s_t)$ is a sequence of solutions that are the search history of the EA until time t and μ is the number of initial solutions, and $\mathcal{X}^* = \{x \in \mathcal{X} \mid 1^n \in x\}$, where $s \in x$ means that s appears in the sequence. Therefore, the chain $\xi \in \mathcal{X}$ models an arbitrary mutation-based EA on any function in UBoolean. Obviously, $\forall i \geq 0$, $\xi_i \in \{(s_1, \ldots, s_t) \mid s_j \in \{0,1\}^n, t = \mu + i\}$.

Let $\xi' \in \mathcal{Y}$ model the reference process that is the (1+1)-EA running on the OneMax problem. Then $\mathcal{Y} = \{0,1\}^n$ and $\mathcal{Y}^* = \{1^n\}$. We construct the function $\phi : \mathcal{X} \to \mathcal{Y}$ that $\phi(x) = 1^{n-i}0^i$ with $i = \min\{|s|_0 \mid s \in x\}$. It is easy to see that such a ϕ is an optimal-aligned mapping because $\phi(x) = 1^n$ iff $1^n \in x$ iff $x \in \mathcal{X}^*$.

Then, we investigate the condition Eq.(1) of switch analysis. For any $x \notin \mathcal{X}^*$, assume $|\phi(x)|_0 = i > 0$. Let P_j be the probability that the offspring solution generated on $\phi(x)$ by bit-wise mutation has j number of 0 bits. For ξ', it accepts only the offspring solution with no more 0 bits than the parent, thus, we have

$$\begin{aligned}&\sum\nolimits_{y \in \mathcal{Y}} P(\xi'_1 = y \mid \xi'_0 = \phi(x))\mathbb{E}[\![\tau' \mid \xi'_1 = y]\!] \\ &= \sum\nolimits_{j=0}^{i-1} P_j \mathbb{E}(j) + (1 - \sum\nolimits_{j=0}^{i-1} P_j)\mathbb{E}(i).\end{aligned}$$

For ξ, it selects a solution s from x for reproduction. Let P'_j be the probability that the offspring solution s' generated on s by bit-wise mutation has j number of 0 bits. If $|s'|_0 < i$, $|\phi((x, s'))|_0 = |s'|_0$; otherwise, $|\phi((x, s'))| = i$, where (x, s') is the solution sequence until time $t+1$. Thus, we have

$$\begin{aligned}&\sum\nolimits_{y \in \mathcal{Y}} P(\xi_{t+1} \in \phi^{-1}(y) \mid \xi_t = x)\mathbb{E}[\![\tau' \mid \xi'_0 = y]\!] \\ &= \sum\nolimits_{j=0}^{i-1} P'_j \mathbb{E}(j) + (1 - \sum\nolimits_{j=0}^{i-1} P'_j)\mathbb{E}(i).\end{aligned}$$

By the definition of ϕ, we have $|s|_0 \geq |\phi(x)|_0 = i$. Then, by Lemma 4, $\sum_{j=0}^{k} P_j \geq \sum_{j=0}^{k} P'_j$ for any $k \in [0, n]$. Meanwhile, $\mathbb{E}(i)$ increases with i as in Lemma 3. Thus, by Lemma 5, we have

$$\begin{aligned}&\sum\nolimits_{j=0}^{i-1} P'_j \mathbb{E}(j) + (1 - \sum\nolimits_{j=0}^{i-1} P'_j)\mathbb{E}(i) \\ &\geq \sum\nolimits_{j=0}^{i-1} P_j \mathbb{E}(j) + (1 - \sum\nolimits_{j=0}^{i-1} P_j)\mathbb{E}(i),\end{aligned}$$

which is equivalent to

$$\begin{aligned}&\sum\nolimits_{y \in \mathcal{Y}} P(\xi_{t+1} \in \phi^{-1}(y) \mid \xi_t = x)\mathbb{E}[\![\tau' \mid \xi'_0 = y]\!] \\ &\geq \sum\nolimits_{y \in \mathcal{Y}} P(\xi'_1 = y \mid \xi'_0 = \phi(x))\mathbb{E}[\![\tau' \mid \xi'_1 = y]\!].\end{aligned}$$

Thus, the condition Eq.(1) of switch analysis holds with $\rho_t = 0$. We have $\mathbb{E}[\![\tau | \xi_0 \sim \pi_0]\!] \geq \mathbb{E}[\![\tau' | \xi'_0 \sim \pi_0^\phi]\!]$.

Then, we investigate $\mathbb{E}[\![\tau' | \xi'_0 \sim \pi_0^\phi]\!]$. For mutation-based EAs (i.e., Algorithm 2), the initial population consists of μ solutions $s_1, \ldots, s_\mu$ randomly selected from $\{0,1\}^n$. By the definition of ϕ, we know that $\forall 0 \leq j \leq n : \pi_0^\phi(\{y \in \mathcal{Y} \mid |y|_0 = j\})$ is the probability that $\min\{|s_1|_0, \ldots, |s_\mu|_0\} = j$. Thus,

$$\begin{aligned}\mathbb{E}[\![\tau' | \xi'_0 \sim \pi_0^\phi]\!] &= \sum\nolimits_{j=0}^{n} \pi_0^\phi(\{y \in \mathcal{Y} \mid |y|_0 = j\})\mathbb{E}(j) \\ &= \sum\nolimits_{j=0}^{n} P(\min\{|s_1|_0, \ldots, |s_u|_0\} = j)\mathbb{E}(j),\end{aligned}$$

which is actually the EFHT of the Markov chain modeling (1+1)-EA$_\mu$ on OneMax.

Since both mutation-based EAs and (1+1)-EA$_\mu$ evaluate μ solutions in the initial process and evaluate one solution in each iteration, $\mathbb{E}[\![\tau | \xi_0 \sim \pi_0]\!] \geq$

$\mathbb{E}[\![\tau'|\xi'_0 \sim \pi_0^\phi]\!]$ implies that the expected running time of any mutation-based EA on UBoolean is at least as large as that of (1+1)-EA$_\mu$ on OneMax. □

V. Analysis Approach Characterization

As we have shown that the switch analysis can help analyze the running time of EAs, a natural question is how powerful the switch analysis is, particularly comparing with existing approaches. There have developed several analysis approaches for the running time of EAs, including the fitness level method [31] and the drift analysis [14]. We will compare the switch analysis with the two approaches.

To support the comparative study, we notice that there are rarely rigorous definitions of an "analysis approach", which is a necessary basis for formal discussions. Analysis approaches, in general, are usually conceptually described rather than rigorously defined, and are applied on problems case-by-case. However, a general analysis approach for EA processes commonly specifies a set of variables to look at and a procedure to follow with. Therefore, in this context, it is possible to treat an analysis approach like an algorithm with input, parameters and output. The input is a variable assignment derived from the concerning EA process; the parameters are variable assignments which should rely on no more information of the EA process than the input; and the output is a lower and upper bound of the running time, as described in Definition 8. We distinguish the input with parameters in order to clarify the amount of information that an approach requires from the concerning EA process. Note that the state space $\mathcal{X}$ itself of the EA process is not regarded as part of the input, since it can be known ahead of specifying the optimization problem. For an analysis approach $\mathfrak{A}$, we denote $\mathfrak{A}^u(\Theta;\Omega)$ and $\mathfrak{A}^l(\Theta;\Omega)$ as the upper and lower bounds respectively, given the input Θ and parameters Ω. When the context is clear, we will omit Θ and parameters Ω.

Definition 8 (EA Analysis Approach)
A procedure $\mathfrak{A}$ is called an EA analysis approach if for any EA process $\xi \in \mathcal{X}$ with initial state ξ_0 and transition probability P, $\mathfrak{A}$ provided with $\Theta = g(\xi_0, P)$ for some function g and a set of parameters $\Omega(\Theta)$ outputs a lower running time bound of ξ notated as $\mathfrak{A}^l(\Theta;\Omega)$ and/or an upper bound $\mathfrak{A}^u(\Theta;\Omega)$.

We are interested in the tightness of the output bounds of an analysis approach with limited information from the concerning EA process, rather than its "computational cost". Note that some mathematical operators, such as the inverse of an irregular matrix, may not be practical. We assume that all calculations discussed in this paper are efficient for simplicity.

As for the formal characterization of switch analysis, we need to specify the input, the parameters and the output. Since we are considering analyzing the running time of an EA process, all the variables derived from the reference process used in the switch analysis are regarded as parameters, which include bounds of the one-step transition probabilities and the CFHT of the reference process. The input of the switch analysis includes bounds of one-step transition probabilities of the concerning EA process. It should be noted that the tightness of the input bounds determines the optimal tightness of the output bounds we can have, and then the goodness of the selected parameter values determines how close the actually derived bounds are to the optimal bounds; thus we do not specify how tight the input and how good the parameters should be when characterizing approaches. The switch analysis is formally characterized in Characterization 1.

Characterization 1 (Switch Analysis)
For an EA process $\xi \in \mathcal{X}$, the switch analysis approach $\mathfrak{A}_{SA}$ is defined by its parameters, input and output:
Parameters: a reference process $\xi' \in \mathcal{Y}$ with bounds of its transition probabilities $P(\xi'_1|\xi'_0)$ and CFHT $\mathbb{E}[\![\tau' \mid \xi'_t = y]\!]$ for all $y \in \mathcal{Y}$ and $t \in \{0,1\}$, and a right-aligned mapping $\phi^u : \mathcal{X} \to \mathcal{Y}$ or a left-aligned mapping $\phi^l : \mathcal{X} \to \mathcal{Y}$.
Input: bounds of one-step transition probabilities $P(\xi_{t+1} \mid \xi_t)$.
Output: denoting $\pi_t^\phi(y) = \pi_t(\phi^{-1}(y))$ for all $y \in \mathcal{Y}$,
$\mathfrak{A}^u_{SA} = \mathbb{E}[\![\tau' \mid \xi'_0 \sim \pi_0^\phi]\!] + \rho^u$ *where* $\rho^u = \sum_{t=0}^{+\infty} \rho_t^u$ *and*
$\rho_t^u \geq \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_t(x)P(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!] - \sum_{u,y\in\mathcal{Y}} \pi_t^\phi(u)P(\xi'_1 = y \mid \xi'_0 = u)\mathbb{E}[\![\tau' \mid \xi'_1 = y]\!]$ *for all t;*
$\mathfrak{A}^l_{SA} = \mathbb{E}[\![\tau' \mid \xi'_0 \sim \pi_0^\phi]\!] + \rho^l$ *where* $\rho^l = \sum_{t=0}^{+\infty} \rho_t^l$ *and*
$\rho_t^l \leq \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_t(x)P(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!] - \sum_{u,y\in\mathcal{Y}} \pi_t^\phi(u)P(\xi'_1 = y \mid \xi'_0 = u)\mathbb{E}[\![\tau' \mid \xi'_1 = y]\!]$ *for all t.*

As analysis approaches are characterized by their input, parameters and output, we can then study their relative power. In the first thought, if one approach derives tighter running time bounds than another, the former is more powerful. However, the tightness is effected by many aspects. Different usages of a method can result in different bounds. We shall not compare the results of particular uses of two approaches. Therefore, we define the reducibility between analysis approaches in Definition 9.

Definition 9 (Reducible)
For two EA analysis approaches $\mathfrak{A}_1$ and $\mathfrak{A}_2$, if for any input Θ and any parameter Ω_A, there exist a transformation T and parameter Ω_B (which possibly depends on Ω_A) such that
(a) $\mathfrak{A}_1^u(\Theta;\Omega_A) \geq \mathfrak{A}_2^u(T(\Theta);\Omega_B)$, then $\mathfrak{A}_1$ is upper-bound reducible *to $\mathfrak{A}_2$;*
(b) $\mathfrak{A}_1^l(\Theta;\Omega_A) \leq \mathfrak{A}_2^l(T(\Theta);\Omega_B)$, then $\mathfrak{A}_1$ is lower-bound reducible *to $\mathfrak{A}_2$.*
Moreover, $\mathfrak{A}_1$ is reducible *to $\mathfrak{A}_2$ if it is both upper-bound reducible and lower-bound reducible.*

By the definition, for analysis approaches $\mathfrak{A}_1$ and $\mathfrak{A}_2$, we say that "$\mathfrak{A}_1$ is reducible to $\mathfrak{A}_2$", if it is possible to construct an input of $\mathfrak{A}_2$ by the transformation T solely

from the input of $\mathfrak{A}_1$, while $\mathfrak{A}_2$ using some parameters outputs a bound that is at least as good as that of $\mathfrak{A}_1$. If no such transformation or parameters exists, $\mathfrak{A}_1$ is not reducible to $\mathfrak{A}_2$. Intuitively there are two possible reasons that one approach is not reducible to another: one is that the latter cannot take all the input of the former, i.e., T has to lose important information in the input; and the other is that, though T does not lose information, the latter cannot make full use of it. When $\mathfrak{A}_1$ is proved to be reducible to $\mathfrak{A}_2$, we can say that $\mathfrak{A}_2$ is at least as powerful as $\mathfrak{A}_1$ since it takes no more input information but derives no loose bounds. However, this does not imply that $\mathfrak{A}_2$ is *easier* to use. The usability of an analysis approach can also depend on its intuitiveness and the background of the analyst, which is out of the consideration of this work.

VI. Switch Analysis v.s. Fitness Level Method

A. Fitness Level Method

Fitness level method [31] is an intuitive method for analyzing expected running time of EAs. Given an EA process, we partition the solution space into level sets according to their fitness values, and order the level sets according to the fitness of the solutions in the sets. This partition is formally described in Definition 10 for maximizing problems.

Definition 10 ($<_f$-Partition [31])
Given a problem $f : \mathcal{S} \to \mathbb{R}$ and the solution space $\mathcal{S}$ with target subspace $\mathcal{S}^*$, for all $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{S}$, the relation $\mathcal{S}_1 <_f \mathcal{S}_2$ holds if $f(a) < f(b)$ for all $a \in \mathcal{S}_1$ and $b \in \mathcal{S}_2$. A $<_f$-partition of $\mathcal{S}$ is a partition of $\mathcal{S}$ into non-empty sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ such that $\mathcal{S}_1 <_f \mathcal{S}_2 <_f \ldots <_f \mathcal{S}_m$ and $\mathcal{S}_m = \mathcal{S}^*$.

Note that elitist (i.e., never lose the best solution) EAs select solutions with better fitness. The level sets, intuitively, form stairs, based on which an upper bound can be derived by summing up the maximum time taken for leaving every stair, and a lower bound is the minimum time of leaving a stair (i.e., we optimistically assume that the optimum is reached by a single jump). This is formally described in Lemma 6, with a slight but equivalent modification from the original definition to unify the lower and upper bounds. In the lemma, when the EA uses a population of solutions, the notation $\xi_t \in \mathcal{S}_i$ will denote that the best solution of the population ξ_t is in the solution space $\mathcal{S}_i$.

Lemma 6 (Fitness Level Method [31])
For an elitist EA process ξ on a problem f, let $\mathcal{S}_1, ..., \mathcal{S}_m$ be a $<_f$-partition, and let $v_i \leq P(\xi_{t+1} \in \cup_{j=i+1}^{m} \mathcal{S}_j \mid \xi_t = x)$ for all $x \in \mathcal{S}_i$, and $u_i \geq P(\xi_{t+1} \in \cup_{j=i+1}^{m} \mathcal{S}_j \mid \xi_t = x)$ for all $x \in \mathcal{S}_i$. Then, the DCFHT of the EA process is at most

$$\sum\nolimits_{1 \leq i \leq m-1} \pi_0(\mathcal{S}_i) \cdot \sum\nolimits_{j=i}^{m-1} \frac{1}{v_j},$$

and is at least

$$\sum\nolimits_{1 \leq i \leq m-1} \pi_0(\mathcal{S}_i) \cdot \frac{1}{u_i}.$$

Later on, more elaborated fitness level method was discovered by Sudholt [27], which we call as the refined fitness level method in this paper as in Lemma 7.

Lemma 7 (Refined Fitness Level Method [26], [27])
For an elitist EA process ξ on a problem f, let $\mathcal{S}_1, ..., \mathcal{S}_m$ be a $<_f$-partition, let $v_i \leq \min_j \frac{1}{\gamma_{i,j}} P(\xi_{t+1} \in \mathcal{S}_j \mid \xi_t = x)$ and $u_i \geq \max_j \frac{1}{\gamma_{i,j}} P(\xi_{t+1} \in \mathcal{S}_j \mid \xi_t = x)$ for all $x \in \mathcal{S}_i$ where $\sum_{j=i+1}^{m} \gamma_{i,j} = 1$, and let $\chi_u, \chi_l \in [0,1]$ be constants such that $\chi_u \geq \gamma_{i,j} / \sum_{k=j}^{m} \gamma_{i,k} \geq \chi_l$ for all $i < j < m$, $\chi_u \geq 1 - v_{j+1}/v_j$ and $\chi_l \geq 1 - u_{j+1}/u_j$ for all $1 \leq j \leq m-2$. Then the DCFHT of the EA process is at most

$$\sum\nolimits_{i=1}^{m-1} \pi_0(\mathcal{S}_i) \cdot \big(\frac{1}{v_i} + \chi_u \sum\nolimits_{j=i+1}^{m-1} \frac{1}{v_j}\big),$$

and is at least

$$\sum\nolimits_{i=1}^{m-1} \pi_0(\mathcal{S}_i) \cdot \big(\frac{1}{u_i} + \chi_l \sum\nolimits_{j=i+1}^{m-1} \frac{1}{u_j}\big).$$

The refined fitness level method follows the general idea of the fitness level method, while introduces a variable χ that reflects the distribution of the probability that the EA jumps to better levels. When χ is small, the EA has a high probability to jump across many levels and thus make a large progress; when χ is large, the EA can only take a small progress in every step. Obviously, χ can take 1 for upper bounds and 0 for lower bounds, which degrades the refined method to be the original fitness level method. Therefore, the original fitness level method is a special case of the refined one.

Since Lemma 6 is a special case of Lemma 7 in respect of upper and lower bounds, we characterize the fitness level method using the latter lemma in Characterization 2.

Characterization 2 (Fitness Level Method)
For an EA process $\xi \in \mathcal{X}$, the fitness level method $\mathfrak{A}_{FL}$ is defined by its parameters, input and output:
Parameters: a $<_f$-partition $\{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$, where $\mathcal{S}_1 <_f \mathcal{S}_2 <_f \ldots <_f \mathcal{S}_m = \mathcal{S}^*$.
Input: for some non-negative variables $\sum_{j=i+1}^{m} \gamma_{i,j} = 1$, transition probability bounds
$v_i \leq \min_{x \in \mathcal{S}_i} \min_j \frac{1}{\gamma_{i,j}} P(\xi_{t+1} \in \mathcal{S}_j \mid \xi_t = x)$,
$u_i \geq \max_{x \in \mathcal{S}_i} \max_j \frac{1}{\gamma_{i,j}} P(\xi_{t+1} \in \mathcal{S}_j \mid \xi_t = x)$,
$\chi_u \geq \gamma_{i,j} / \sum_{k=j}^{m} \gamma_{i,k} \geq \chi_l$ for all $i < j < m$,
$\chi_u \geq 1 - v_{j+1}/v_j$ and $\chi_l \geq 1 - u_{j+1}/u_j$ for all $1 \leq j \leq m-2$.
Output:
$\overline{\mathfrak{A}^u_{FL}} = \sum_{i=1}^{m-1} \pi_0(\mathcal{S}^i) \cdot \big(\frac{1}{v_i} + \chi_u \sum_{j=i+1}^{m-1} \frac{1}{v_j}\big)$,
$\mathfrak{A}^l_{FL} = \sum_{i=1}^{m-1} \pi_0(\mathcal{S}^i) \cdot \big(\frac{1}{u_i} + \chi_l \sum_{j=i+1}^{m-1} \frac{1}{u_j}\big)$.

B. The Power of Switch Analysis from Fitness Level Method

Theorem 3
$\mathfrak{A}_{FL}$ is reducible to $\mathfrak{A}_{SA}$.

Before proving the theorem, we introduce a simple Markov chain, OneJump, that will be used as the reference chain for switch analysis.

Definition 11 (OneJump)
OneJump chain with state space dimension n is a homogeneous Markov chain $\xi \in \{0,1\}^n$ with $n+1$ parameters $\{p_0,\ldots,p_n\}$ each is in $[0,1]$ and target state 1^n. Its initial state is selected from $\{0,1\}^n$ uniformly at random, and its transition probability is defined as, for any $x \in \{0,1\}^n$ and any t,

$$P(\xi_{t+1}=y \mid \xi_t=x)=\begin{cases} p_{|x|_1}, & y=1^n \\ 1-p_{|x|_1}, & y=x \\ 0, & \textit{otherwise} \end{cases},$$

where $|x|_1$ is the number of 1 bits in x.

It is straightforward to calculate the CFHT of OneJump, which is $\frac{1}{p_{n-j}}$ when starting from a solution s with $|s|_1 = n-j$. As the CFHT depends only on the number of 0 bits, we denote $\mathbb{E}_{oj}(j)=\frac{1}{p_{n-j}}$ as the CFHT of OneJump starting from solutions with j 0 bits, for simplicity.

Theorem 3 is proved by combining Lemma 8 and Lemma 9, which respectively prove the upper bound and lower bound reducibility.

Lemma 8
$\mathfrak{A}_{FL}$ is upper-bound reducible to $\mathfrak{A}_{SA}$.
Proof. The proof is to find the parameters of $\mathfrak{A}_{SA}$ and the input of $\mathfrak{A}_{SA}$ from that of $\mathfrak{A}_{FL}$, and show that $\mathfrak{A}^u_{SA} \le \mathfrak{A}^u_{FL}$.

Denote $\xi \in \mathcal{X}$ as the EA process we are going to analyze. By the parameter of $\mathfrak{A}_{FL}$, we have a $<_f$-partition $\{\mathcal{S}_1,\mathcal{S}_2,\ldots,\mathcal{S}_m\}$, which divides the solution space into m subspaces. We moreover know some variables v_i, $\gamma_{i,j}$ and χ_u as in Characterization 2, which are the input of $\mathfrak{A}_{FL}$.

We choose the reference process $\xi' \in \mathcal{Y}=\{0,1\}^{m-1}$ to be the OneJump with dimension $m-1$ and parameters $p_i = 1/(\frac{1}{v_{i+1}}+\chi_u\sum_{j=i+2}^{m-1}\frac{1}{v_j})$ $(0 \le i < m-1)$. We construct the function $\phi : \mathcal{X} \to \mathcal{Y}$ as that $\phi(x) = 1^{i-1}0^{m-i}$ for all $x \in \mathcal{S}_i$. It is easy to verify that ϕ is an optimal-aligned mapping since $\phi(x) \in \mathcal{Y}^* = \{1^{m-1}\}$ if and only if $x \in \mathcal{S}_m$.

We then calculate the upper bound output of $\mathfrak{A}_{SA}$ using the input of $\mathfrak{A}_{FL}$ and the reference process. By the function ϕ, populations in the partition $\mathcal{S}_i$ are mapped to the solution $1^{i-1}0^{m-i}$ for the reference process, starting from which OneJump takes $\mathbb{E}_{oj}(m-i)$ steps to reach its optimum. We then consider one-step transition of ξ. By the input of $\mathfrak{A}_{FL}$, we know that variables v_i (together with $\gamma_{i,j}$ and χ_u) bound from below the probability of generating better solutions. Thus we can have an upper bound on the left part of Eq.(1): for any non-optimal state $x \in \mathcal{S}_i$ $(i<m)$,

$$\begin{aligned}&\sum\nolimits_{y\in\mathcal{Y}} P(\xi_{t+1}\in\phi^{-1}(y)\mid\xi_t=x)\mathbb{E}[\![\tau'\mid\xi'_0=y]\!] \qquad (6)\\ &\le v_i\sum\nolimits_{j=i+1}^{m}\gamma_{i,j}\mathbb{E}_{oj}(m-j)+(1-v_i)\mathbb{E}_{oj}(m-i),\end{aligned}$$

where the first part is the sum of the events that better solutions are found and the second part is the remaining event. The inequality holds because $\chi_u \ge 1-v_{j+1}/v_j$ leading to $\mathbb{E}_{oj}(m-i) \ge \mathbb{E}_{oj}(m-j)$ for all $j=i+1,\ldots,m$. Meanwhile considering the reference process, for any $x \in \mathcal{S}_i$, since $\phi(x)=1^{i-1}0^{m-i}$, we have

$$\begin{aligned}&\sum\nolimits_{y\in\mathcal{Y}} P(\xi'_1=y\mid\xi'_0=\phi(x))\mathbb{E}[\![\tau'\mid\xi'_1=y]\!] \qquad (7)\\ &=(1-p_{i-1})\mathbb{E}_{oj}(m-i)=\mathbb{E}_{oj}(m-i)-p_{i-1}\cdot\frac{1}{p_{i-1}}\\ &=\mathbb{E}_{oj}(m-i)-1.\end{aligned}$$

By comparing Eq.(6) with Eq.(7), we get, for any $x \in S_i$ $(i<m)$, we have

$$\begin{aligned}&\sum_{y\in\mathcal{Y}} P(\xi_{t+1}\in\phi^{-1}(y)\mid\xi_t=x)\mathbb{E}[\![\tau'\mid\xi'_0=y]\!]\\ &-\sum_{y\in\mathcal{Y}} P(\xi'_1=y\mid\xi'_0=\phi(x))\mathbb{E}[\![\tau'\mid\xi'_1=y]\!]\\ &\le 1+\sum_{j=i+1}^{m-1} v_i\gamma_{i,j}\mathbb{E}_{oj}(m-j)-v_i\mathbb{E}_{oj}(m-i)\\ &=1+\sum_{j=i+1}^{m-1} v_i\gamma_{i,j}(\frac{1}{v_j}+\chi_u\sum_{k=j+1}^{m-1}\frac{1}{v_k})-v_i(\frac{1}{v_i}+\chi_u\sum_{j=i+1}^{m-1}\frac{1}{v_j})\\ &\quad(\text{by } \mathbb{E}_{oj}(m-j)=\frac{1}{p_{j-1}} \text{ and } \mathbb{E}_{oj}(m-i)=\frac{1}{p_{i-1}})\\ &=1+v_i\sum_{j=i+1}^{m-1}\frac{1}{v_j}(\gamma_{i,j}+\chi_u\sum_{k=i+1}^{j-1}\gamma_{i,k})-v_i(\frac{1}{v_i}+\chi_u\sum_{j=i+1}^{m-1}\frac{1}{v_j})\\ &\le 1+v_i\sum_{j=i+1}^{m-1}\frac{1}{v_j}(\chi_u\sum_{k=j}^{m}\gamma_{i,k}+\chi_u\sum_{k=i+1}^{j-1}\gamma_{i,k})\\ &\quad-v_i(\frac{1}{v_i}+\chi_u\sum_{j=i+1}^{m-1}\frac{1}{v_j}) \quad(\text{by } \chi_u \ge \gamma_{i,j}/\sum_{k=j}^{m}\gamma_{i,k})\\ &=0. \quad(\text{by } \sum_{k=i+1}^{m}\gamma_{i,k}=1)\end{aligned}$$

Thus, for all $t \ge 0$,

$$\begin{aligned}&\sum_{x\in\mathcal{X},y\in\mathcal{Y}}\pi_t(x)P(\xi_{t+1}\in\phi^{-1}(y)|\xi_t=x)\mathbb{E}[\![\tau'|\xi'_0=y]\!]\\ &\quad-\sum_{u,y\in\mathcal{Y}}\pi^{\phi}_t(u)P(\xi'_1=y|\xi'_0=u)\mathbb{E}[\![\tau'|\xi'_1=y]\!]\\ &\le 0.\end{aligned}$$

Therefore, we find $\rho^u_t=0$ for all t as a proper assignment of ρ^u_t in Characterization 1, and thus ρ^u=0. We then can calculate the upper bound output, noticing $\pi^{\phi}_0(y)=\pi_0(\phi^{-1}(y))$,

$$\begin{aligned}\mathfrak{A}^u_{SA}&=\mathbb{E}[\![\tau'|\xi'_0\sim\pi^{\phi}_0]\!]+0\\ &=\sum_{i=1}^{m}\pi_0(\mathcal{S}_i)\mathbb{E}_{oj}(m-i)\\ &=\sum_{i=1}^{m-1}\pi_0(\mathcal{S}_i)(\frac{1}{v_i}+\chi_u\sum_{j=i+1}^{m-1}\frac{1}{v_j})=\mathfrak{A}^u_{FL},\end{aligned}$$

which proves the lemma. □

Lemma 9
$\mathfrak{A}_{FL}$ is lower-bound reducible to $\mathfrak{A}_{SA}$.

The proof for Lemma 9 is similar to that for Lemma 8 except the change of variables and the corresponding inequality directions.

The proof of the reducibility is constructive, which provides a way that the switch analysis can simulate the fitness level method. Through the way, any proofs accomplished using the fitness level method can be also accomplished using the switch analysis.

C. Case Study on Peak Problem

As we have proven that $\mathfrak{A}_{FL}$ is reducible to $\mathfrak{A}_{SA}$, a following natural question is whether the inverse is also true, i.e., "is $\mathfrak{A}_{SA}$ reducible to $\mathfrak{A}_{FL}$?". In this section, through investigation on the Peak problem, we find the answer to the question is negative, which implies that the switch analysis is strictly more powerful than the fitness level method.

Definition 12 (Peak Problem)
Peak Problem of size n is to find an n bits binary string s^* such that

$$s^* = \arg\max_{s\in\{0,1\}^n} \prod_{i=1}^{n} s_i,$$

where s_i is the i-th bit of a solution $s \in \{0,1\}^n$.

The Peak problem is to seek a needle in a haystack. The optimal solution is 1^n with fitness value 1, while all the other solutions are with fitness value 0. The only possible $<_f$-partition for the peak problem contains two sets: $\mathcal{S}_1$ containing all non-optimal solutions and $\mathcal{S}_2$ containing the optimal solution.

We study the running time of (1+1)-EA$^{\neq}$ on the Peak problem. (1+1)-EA$^{\neq}$ is the same as (1+1)-EA but does not accept solutions with equal fitness value in the selection. Consequently, on the Peak problem, (1+1)-EA will perform a random walk in non-optimal solutions, since they are with the same fitness value, while (1+1)-EA$^{\neq}$ stays where it is until the optimal solution is found.

Theorem 4
$\mathfrak{A}_{SA}$ is not reducible to $\mathfrak{A}_{FL}$.

The theorem is proved by contrasting Lemma 10 and Lemma 11.

Lemma 10
For the process that (1+1)-EA$^{\neq}$ with bit-wise mutation on the Peak problem with size n under uniform initial distribution, for all possible parameters, $\mathfrak{A}^u_{FL} \geq (1-\frac{1}{2^n})n^n$ and $\mathfrak{A}^l_{FL} \leq (1-\frac{1}{2^n})n(\frac{n}{n-1})^{n-1}$.

Proof. For the Peak problem, the $<_f$-partition can only contain two sets $\{\mathcal{S}_1, \mathcal{S}_2\}$ where $\mathcal{S}_1 = \{0,1\}^n - \{1^n\}$ and $\mathcal{S}_2 = \{1^n\}$. Therefore, we can only have $m = 2$ and $\gamma_{1,2} = 1$, which lead to $\mathfrak{A}^u_{FL} = \pi_0(\mathcal{S}_1)\cdot\frac{1}{v_1} = (1-\frac{1}{2^n})\cdot\frac{1}{v_1}$ and $\mathfrak{A}^l_{FL} = (1-\frac{1}{2^n})\cdot\frac{1}{u_1}$ where v_1 and u_1 are respectively lower and upper bounds of transition probability from $\mathcal{S}_1$ to $\mathcal{S}_2$.

Given a solution s with $|s|_1 = n-j$ $(j>0)$, the probability that it is mutated to 1^n in one step is $\frac{1}{n^j}(1-\frac{1}{n})^{n-j}$, which decreases with j. By definition, $v_1 \leq \min_{s\in\mathcal{S}_1} P(\xi_{t+1}\in\mathcal{S}_2 \mid \xi_t = s) = \frac{1}{n^n}$, and $u_1 \geq \max_{s\in\mathcal{S}_1} P(\xi_{t+1}\in\mathcal{S}_2 \mid \xi_t = s) = \frac{1}{n}(1-\frac{1}{n})^{n-1}$. Therefore, $\mathfrak{A}^u_{FL} \geq (1-\frac{1}{2^n})n^n$ and $\mathfrak{A}^l_{FL} \leq (1-\frac{1}{2^n})n(\frac{n}{n-1})^{n-1}$. □

Lemma 11
For the process that (1+1)-EA$^{\neq}$ with bit-wise mutation on the Peak problem with size n under uniform initial distribution, there exists an assignment of parameters such that $\mathfrak{A}^u_{SA} \leq \left(\frac{n}{2}+\frac{n}{2(n-1)}\right)^n$ and $\mathfrak{A}^l_{SA} \geq \left(\frac{n}{2}\right)^n$.

Proof. We choose the reference process to be the One-Jump with dimension n and parameters $p_i = (\frac{1}{n})^{n-i}(1-\frac{1}{n})^i$ $(i<n)$. We simply use the function $\phi(x) = x$, which is obviously an optimal-aligned mapping.

We investigate Eq.(1). For any $x \notin \mathcal{X}^*$ with $|x|_1 = n-j$ $(j>0)$, we have

$$\begin{aligned}
&\sum_{y\in\mathcal{Y}} P(\xi'_1 = y|\xi'_0 = \phi(x))\mathbb{E}[\![\tau'|\xi'_1 = y]\!] \\
&= (1-p_{n-j})\mathbb{E}_{oj}(j) \\
&= (1-\frac{1}{n^j}(1-\frac{1}{n})^{n-j})\mathbb{E}_{oj}(j) \\
&= \sum_{y\in\mathcal{Y}} P(\xi_{t+1}\in\phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!].
\end{aligned}$$

Thus, we have found proper $\rho^u_t = \rho^l_t = 0$, then $\rho^u = \rho^l = 0$. By the switch analysis theorem, we have $\mathbb{E}[\![\tau|\xi_0\sim\pi_0]\!] \leq$ (and $\geq$)$\mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!]$. Moreover,

$$\begin{aligned}
\mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!] &= \sum_{j=1}^{n}\frac{\binom{n}{j}}{2^n}\cdot\mathbb{E}_{oj}(j) = \sum_{j=1}^{n}\frac{\binom{n}{j}}{2^n}\cdot n^j(\frac{n}{n-1})^{n-j} \\
&= \frac{1}{2^n}((n+\frac{n}{n-1})^n - (\frac{n}{n-1})^n).
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\mathfrak{A}^u_{SA} &= \mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!] \\
&= \frac{1}{2^n}((n+\frac{n}{n-1})^n - (\frac{n}{n-1})^n) \leq \left(\frac{n}{2}+\frac{n}{2(n-1)}\right)^n,
\end{aligned}$$

and

$$\begin{aligned}
\mathfrak{A}^l_{SA} &= \mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!] \\
&= \frac{1}{2^n}((n+\frac{n}{n-1})^n - (\frac{n}{n-1})^n) \geq \left(\frac{n}{2}\right)^n,
\end{aligned}$$

which proves the lemma. □

It is straightforward to verify by comparing Lemma 10 with Lemma 11 that $\mathfrak{A}^u_{SA} < \mathfrak{A}^u_{FL}$ and $\mathfrak{A}^l_{SA} > \mathfrak{A}^l_{FL}$. Actually, the fitness level method can only derive a polynomial lower bound for this process, which is quite loose; while switch analysis can lead to tight bounds since it allows a fine investigation between fitness levels. In other words, the fitness level method cannot take all the input of switch analysis. Therefore, the case study proves that $\mathfrak{A}_{SA}$ is not reducible to $\mathfrak{A}_{FL}$.

VII. Switch Analysis v.s. Drift Analysis

A. Drift Analysis

Drift analysis [12], [14], [16], [24] might be nowadays the most widely used approach for analyzing running time of EAs. There have emerged several variants of

drift analysis [3], [5], [13], [21], which simplify and combine other techniques with drift analysis. In this work, we focus on the classical drift analysis, which is also called additive drift analysis as in Lemma 12.

The drift analysis embodied by Lemma 12, can be intuitively understood. Drift analysis introduces a distance function (not necessarily a metric) to measure the distance from any population to the optimal population space. Relying on the distance function, drift analysis estimates the average progress toward the optimum of every step of an EA, and then the number of steps the EA takes to arrive at the optimum can be derived through dividing the initial distance by one-step progress.

Note that the recently proposed two variants of classical drift analysis, the multiplicative drift analysis [5] and the adaptive drift analysis [3], are not stronger than the classical one, since the multiplicative drift analysis is proved by the classical one and the adaptive drift analysis is actually the classical (multiplicative) one with carefully-designed distance functions which may depend on the objective functions and some parameter values of the analyzed EAs.

Definition 13 (Distance Function)
For a state space $\mathcal{X}$ with the optimal subspace $\mathcal{X}^*$, a function V satisfying $V(x) = 0$ for all $x \in \mathcal{X}^*$ and $V(x) > 0$ for all $x \in \mathcal{X} - \mathcal{X}^*$ is called a distance function.

Lemma 12 (Drift Analysis [14], [16])
For an EA process $\xi \in \mathcal{X}$, let V be a distance function, if there exists a positive value c_l such that

$$\forall t \geq 0, \xi_t \notin \mathcal{X}^* : c_l \leq \mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]\!],$$

we have $\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!] \leq \sum_{x\in\mathcal{X}} \pi_0(x)V(x)/c_l$;
and if there exists a positive value c_u such that

$$\forall t \geq 0, \xi_t \notin \mathcal{X}^* : c_u \geq \mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]\!],$$

we have $\mathbb{E}[\![\tau \mid \xi_0 \sim \pi_0]\!] \geq \sum_{x\in\mathcal{X}} \pi_0(x)V(x)/c_u$.

It should be noted that, when c_l is negative, we will say that the drift analysis fails with such a distance function.

Characterization 3 (Drift Analysis)
For an EA process $\xi \in \mathcal{X}$, the drift analysis approach $\mathfrak{A}_{DA}$ is defined by its parameters, input and output:
Paramters: a distance function V.
Input:
$c_l > 0$ for upper bound analysis such that $c_l \leq \mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]\!]$ for all $t \geq 0, \xi_t \notin \mathcal{X}^*$;
$c_u > 0$ for lower bound analysis such that $c_u \geq \mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]\!]$ for all $t \geq 0, \xi_t \notin \mathcal{X}^*$.
Output:
$\mathfrak{A}^u_{DA} = \sum_{x\in\mathcal{X}} \pi_0(x)V(x)/c_l$;
$\mathfrak{A}^l_{DA} = \sum_{x\in\mathcal{X}} \pi_0(x)V(x)/c_u$.

B. The Power of Switch Analysis from Drift Analysis

Theorem 5
$\mathfrak{A}_{DA}$ is reducible to $\mathfrak{A}_{SA}$.

Lemma 13
$\mathfrak{A}_{DA}$ is upper-bound reducible to $\mathfrak{A}_{SA}$.

Proof. The proof is to construct parameters and input of $\mathfrak{A}_{SA}$ from that of $\mathfrak{A}_{DA}$, such that $\mathfrak{A}^u_{SA} \leq \mathfrak{A}^u_{DA}$.

Denote ξ as the EA process we are going to analyze. By $\mathfrak{A}_{DA}$, we have a distance function V as the parameter, and c_l as the input, as in Characterization 3.

Let $\mathcal{V} = \{V(x) \mid x \in \mathcal{X}\} = \{V_0, V_1, \ldots, V_m\}$ be the set of distinct values of the distance function, the size of which is $m+1$. We order the elements of $\mathcal{V}$ as $V_0 = 0 < V_1 < \ldots < V_m$, and denote the value index of x as $\mathcal{V}_x = i$ where $V(x) = V_i$. Without loss of generality, we assume $V_1 \geq 1$, since if not, we multiply every element in $\mathcal{V}$ as well as c_l with $\frac{1}{V_1}$, which does not affect the drift condition and result.

We choose the reference process $\xi' \in \mathcal{Y} = \{0,1\}^m$ to be the OneJump with dimension m and parameters $p_i = \frac{1}{V_{m-i}}$ $(i = 0, \ldots, m-1)$. We then construct the function $\phi(x) = 1^{m-\mathcal{V}_x}0^{\mathcal{V}_x}$, which uses only the distance function value index. It is easy to verify that the function is an optimal-aligned mapping, since all populations with distance function value 0 are mapped to 1^m.

We investigate Eq.(1). For any $x \notin \mathcal{X}^*$, we have

$$\begin{aligned}
&\sum_{y\in\mathcal{Y}} P(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!] \\
&= \sum_{i=0,\ldots,m} P(V(\xi_{t+1}) = V_i \mid \xi_t = x) \cdot \mathbb{E}_{oj}(i) \\
&= \sum_{i=0,\ldots,m} P(V(\xi_{t+1}) = V_i \mid \xi_t = x) \cdot V_i \text{ (by } \mathbb{E}_{oj}(i) = \frac{1}{p_{m-i}}) \\
&= \mathbb{E}[\![V(\xi_{t+1}) \mid \xi_t = x]\!].
\end{aligned}$$

Since

$$\begin{aligned}
&\mathbb{E}[\![V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]\!] \\
&= V(x) - \mathbb{E}[\![V(\xi_{t+1}) \mid \xi_t = x]\!] \geq c_l,
\end{aligned}$$

we have

$$\sum_{y\in\mathcal{Y}} P(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!] \leq V(x) - c_l.$$

Meanwhile, since $\phi(x) = 1^{m-\mathcal{V}_x}0^{\mathcal{V}_x}$, for any $x \notin \mathcal{X}^*$ we have

$$\begin{aligned}
&\sum_{y\in\mathcal{Y}} P(\xi'_1 = y|\xi'_0 = \phi(x))\mathbb{E}[\![\tau'|\xi'_1 = y]\!] \\
&= \mathbb{E}_{oj}(\mathcal{V}_x) - 1 = V(x) - 1.
\end{aligned}$$

We then have, for all $x \notin \mathcal{X}^*$,

$$\begin{aligned}
&\sum_{y\in\mathcal{Y}} P(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!] \\
&- \sum_{y\in\mathcal{Y}} P(\xi'_1 = y|\xi'_0 = \phi(x))\mathbb{E}[\![\tau'|\xi'_1 = y]\!] \leq 1 - c_l,
\end{aligned}$$

and thus, by summing up all x, we get

$$\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \pi_t(x)P(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!]$$

$$\leq \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_t(x)P(\xi'_1=y|\xi'_0=\phi(x))\mathbb{E}[\![\tau'|\xi'_1=y]\!] + (1-c_l)\cdot(1-\pi_t(\mathcal{X}^*)).$$

Therefore, we have found a proper $\rho^u_t = (1-c_l)\cdot(1-\pi_t(\mathcal{X}^*))$. We then calculate

$$\rho^u = \sum_{t=0}^{+\infty}\rho^u_t = (1-c_l)\cdot\sum_{t=0}^{+\infty}(1-\pi_t(\mathcal{X}^*)) = (1-c_l)\cdot\mathbb{E}[\![\tau \mid \xi_0\sim\pi_0]\!].$$

Substituting ρ^u, we get

$$\mathbb{E}[\![\tau|\xi_0\sim\pi_0]\!]\leq\mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!] + (1-c_l)\cdot\mathbb{E}[\![\tau \mid \xi_0\sim\pi_0]\!],$$

which derives $\mathbb{E}[\![\tau|\xi_0\sim\pi_0]\!] \leq \mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!]/c_l$. Moreover, by the function ϕ,

$$\mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!] = \sum_{i=0,\ldots,m}\pi_0(\{x \mid \mathcal{V}_x=i\})\cdot\mathbb{E}_{oj}(i) = \sum_{i=0,\ldots,m}\pi_0(\{x \mid V(x)=V_i\})\cdot V_i = \sum_{x\in\mathcal{X}}\pi_0(x)\cdot V(x).$$

Overall, we have,

$$\mathfrak{A}^u_{SA} = \mathbb{E}[\![\tau'|\xi'_0\sim\pi^\phi_0]\!]/c_l = \sum_{x\in\mathcal{X}}\pi_0(x)\cdot V(x)/c_l = \mathfrak{A}^u_{DA},$$

which proves the lemma. □

Lemma 14
$\mathfrak{A}_{DA}$ is lower-bound reducible to $\mathfrak{A}_{SA}$.

The proof of Lemma 14 is similar to that of Lemma 13 except replacing c_l with c_u and changing the direction of the corresponding inequalities. The two lemmas together prove Theorem 5. Moreover, the proofs of the lemmas are constructive, which provide a way of using switch analysis based on the drift analysis.

C. Case Study on Discrete Linear Problem

We are then interested in whether switch analysis is also reducible to drift analysis. It is however hard to obtain a full answer, since that will need to investigate all possible distance functions in an unconstrained function space. By investigating the Discrete Linear Problem, we show in this section that switch analysis is not reducible to a restricted version of drift analysis that uses any fixed linear distance function.

Definition 14 (Discrete Linear Problem)
A Discrete Linear Problem with size n and vocabulary set $\{0,1,\ldots,r\}$ is to find a string s^* from $\{0,1,\ldots,r\}^n$ such that, for given positive weights $w_1,\ldots,w_n$,

$$s^* = \arg\max_{s\in\{0,1,\ldots,r\}^n}\sum_{i=1}^n w_i s_i,$$

where s_i is the value on the i-th position of a solution $s\in\{0,\ldots,r\}^n$. Without loss of generality, we assume that $w_1\leq w_2\leq\ldots\leq w_n$.

To run (1+1)-EA with bit-wise mutation on the Discrete Linear Problem, we need a modification of the mutation. When the input space is $\{0,1\}^n$, the mutation flips a bit of the solution from 0 to 1 and vice versa. While the input space is $\{0,\ldots,r\}^n$, we define that the mutation flips a bit from its own value k to be a random value in $\{0,\ldots,r\}-\{k\}$. Note that the bit-wise mutation defined in the previous sections uses the mutation probability $\frac{1}{n}$ (i.e., each bit is flipped with probability $\frac{1}{n}$); here we generalize it to $p\in(0,1)$.

It has been proved that, for Discrete Linear Problem as in Definition 14, there exists no universal linear distance function such that (1+1)-EA has positive drift.

Definition 15 (Linear Distance Function Space)
For solutions with length n, the linear distance function space $\mathcal{L}$ consists of all distance functions that are linear combination of solution bits, i.e.,

$$\mathcal{L} = \{V \mid \forall \boldsymbol{w}\in\mathbb{R}^n : V(s) = \sum_{i=1}^n w_i s_i\}.$$

Lemma 15 ([4], [6])
For the process of
(a) (1+1)-EA with $p\geq 7/n$ on Discrete Linear Problem with vocabulary $\{0,1\}$,
(b) (1+1)-EA with $p=1/n$ on Discrete Linear Problem with vocabulary $\{0,\ldots,r\}(r\geq 43)$,
$\mathfrak{A}^u_{DA}$ with any fixed parameter $V\in\mathcal{L}$ fails.
Proof. In [6] and [4], it has been proved that under the condition of this lemma, the minimum drift c_l is negative for any distance function in $\mathcal{L}$. Therefore $\mathfrak{A}^u_{DA}$ fails. □

Lemma 16
For the process of (1+1)-EA with $p=c/n$ for some constant $c>0$ on Discrete Linear Problem,
(a) with vocabulary $\{0,1\}$, there exists an assignment of parameters such that $\mathfrak{A}^u_{SA}=O(n\log n)$,
(b) with vocabulary $\{0,\ldots,r\}$, there exists an assignment of parameters such that $\mathfrak{A}^u_{SA} = (1+o(1))\frac{e^c}{c}rn\log n + O(r^3 n\log\log n)$.

Lemma 16 can be straightforwardly proved by applying the Theorem 5 that drift analysis is reducible to switch analysis, and that the bounds have been proved in [3] and [7] using the multiplicative adaptive drift analysis. Contrasting Lemma 16 with Lemma 15, it is obvious that switch analysis is not reducible to the restricted version of drift analysis.

Since the bound in (b) of the Lemma involves r^3, in Lemma 17 we give another upper bound using switch analysis, which is tighter in term of r although looser in term of n.

We define RLS$^{\neq}$ as a modification of RLS, where the only difference is that RLS$^{\neq}$ uses the selection as $f(s')>f(s)$ in step 3 of Algorithm 1. In other words, RLS accepts the offspring solution with equal fitness, while RLS$^{\neq}$ only accepts a better offspring solution. The reference process used in the proof of Lemma 17 is the RLS$^{\neq}$ running on the LeadingOnes problem in Definition 16 with size n. We denote the reference process as ξ' (and thus the first hitting event as τ'), which has a property in Proposition 1.

Definition 16 (LeadingOnes Problem)
LeadingOnes Problem of size n is to find an n bits binary

string s^* such that, defining $LO(s) = \sum_{i=1}^{n} \prod_{j=1}^{i} s_j$,

$$s^* = \arg\max_{s\in\{0,1\}^n} f(s) = \arg\max_{s\in\{0,1\}^n} LO(s),$$

where s_j is the j-th bit of a solution $s \in \{0,1\}^n$.

Proposition 1
$\forall t \geq 0\, \forall s \in \{0,1\}^n : \mathbb{E}[\![\tau' | \xi'_t = s]\!] = n(n - |s|_1)$, *where $|s|_1$ denotes the number of 1 bits of s.*

Proof. Let the initial solution have j $(1 \leq j \leq n)$ 0 bits, which are placed randomly. In the run of the RLS$^{\neq}$, the number of 0 bits of the maintained solution will never increase and the 0 bits will not change places, which is because the RLS$^{\neq}$ flips one bit at a time and that turning a 1 bit to be 0 will never increase the fitness thus will be rejected by the strict selection strategy. Therefore, for one step, with probability $\frac{1}{n}$ a solution with $j-1$ 0 bits will be generated and replaces the maintained solution (i.e., the first 0 bit is flipped); with the remaining probability, the maintained solution keeps unchanged. Thus, the expected steps to decrease the number of 0 bits by 1 is n. Through stepwise improvement, the expected running time to get to the optimal solution from a solution with j 0 bits is $n \cdot j$. Note that "$\forall t \geq 0$" holds since the process is homogeneous, i.e., the process would be the same if starting from another time point. □

Due to the proposition, we simplify our notation by denoting $\mathbb{E}_{rls}(j)$ as the CFHT $\mathbb{E}[\![\tau' | \xi'_t = s]\!]$ with $|s|_1 = n - j$, i.e., $\mathbb{E}_{rls}(j) = n \cdot j$.

Lemma 17
For the process of (1+1)-EA with $p < 0.5$ on Discrete Linear Problem with vocabulary $\{0,\ldots,r\}$, there exists an assignment of parameters such that $\mathfrak{A}^u_{SA} \leq r^2 n/(p(1-p)^{n-1})$.

Proof. We construct the reference process ξ' by running RLS$^{\neq}$ on the LeadingOnes problem of size $r \cdot n$, where the solution space is therefore $\mathcal{Y} = \{0,1\}^{rn}$.

We then need to construct a function from $\mathcal{X} = \{0,\ldots,r\}^n$ to $\mathcal{Y} = \{0,1\}^{rn}$. Given a Discrete Linear Problem with weights $w_1 \leq \ldots \leq w_n$, for any solution x, let $\delta(x,j) = \sum_{i=1}^{n} w_i x_i - r\sum_{i=j}^{n} w_i$, and $\theta(x) = \min\{j \in \{1,\ldots,n+1\} \mid \delta(x,j) \geq 0\}$ (note the sum from $j = n+1$ to n is zero), which is the threshold index that the sum of the last weights is not larger than the fitness value. So for two solutions x and x' with $f(x) \leq f(x')$, we have $\theta(x) \geq \theta(x')$. Then let

$$m(x) = r(n - \theta(x) + 1) + \lfloor \delta(x,\theta(x))/w_{\theta(x)-1} \rfloor.$$

So for two solutions x and x' with $f(x) \leq f(x')$, we have $m(x) \leq m(x')$. This is because, when $\theta(x) = \theta(x')$, denoting $a = w_{\theta(x)-1}$ and $b = r\sum_{i=\theta(x)}^{n} w_i$,

$$\begin{aligned} m(x) - m(x') &= \lfloor \delta(x,\theta(x))/a \rfloor - \lfloor \delta(x',\theta(x))/a \rfloor \\ &= \lfloor (f(x) - b)/a \rfloor - \lfloor (f(x') - b)/a \rfloor \leq 0, \end{aligned}$$

since $f(x) \leq f(x')$; when $\theta(x) \geq \theta(x') + 1$, note that $\delta(x,\theta(x)) < r w_{\theta(x)-1}$ since otherwise $\theta(x)$ is not the minimum index,

$$\begin{aligned} m(x) - m(x') &\leq -r + \lfloor \frac{\delta(x,\theta(x))}{w_{\theta(x)-1}} \rfloor - \lfloor \frac{\delta(x',\theta(x'))}{w_{\theta(x')-1}} \rfloor \\ &\leq -r + r - 0 = 0. \end{aligned}$$

The function is $\phi(x) = 1^{m(x)}0^{rn-m(x)}$. It is also easy to verify that ϕ is an optimal-aligned mapping, since for $x^* = r^n$ we have $\theta(x^*) = 1$, $m(x^*) = rn$ and thus $\phi(x^*) = 1^{rn}$, while vice versa.

We investigate Eq.(1). For any $x \notin \mathcal{X}^*$, we have

$$\begin{aligned} &\sum\nolimits_{y\in\mathcal{Y}} P(\xi'_1 = y \mid \xi'_0 = \phi(x))\mathbb{E}[\![\tau' \mid \xi'_1 = y]\!] \\ &= \mathbb{E}_{rls}(rn - m(x)) - 1 = rn(rn - m(x)) - 1. \end{aligned}$$

For the process ξ, let x' be the next solution after mutating x and passing the selection. By the behavior of the selection, we have $f(x') \geq f(x)$, then $m(x') \geq m(x)$. Moreover, since x is non-optimal, there is at least one position $j \in [\theta(x) - 1, n]$ such that $x_j < r$. Thus, the probability of $m(x') \geq m(x) + 1$ is at least $\frac{1}{r}p(1-p)^{n-1}$ since it is sufficient to flip the value on this position to r and keep other bits unchanged. As we know that $E_{rls}(i)$ increases with i, we then have

$$\begin{aligned} &\sum\nolimits_{y\in\mathcal{Y}} P(\xi_{t+1} \in \phi^{-1}(y) \mid \xi_t = x)\mathbb{E}[\![\tau' \mid \xi'_0 = y]\!] \\ &\leq \frac{1}{r}p(1-p)^{n-1} \cdot \mathbb{E}_{rls}(rn - m(x) - 1) \\ &\quad + (1 - \frac{1}{r}p(1-p)^{n-1}) \cdot \mathbb{E}_{rls}(rn - m(x)) \\ &= rn(rn - m(x)) - np(1-p)^{n-1}. \end{aligned}$$

We then have

$$\begin{aligned} &\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \pi_t(x) P(\xi_{t+1} \in \phi^{-1}(y) | \xi_t = x)\mathbb{E}[\![\tau' | \xi'_0 = y]\!] \\ &\leq \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \pi_t(x) P(\xi'_1 = y | \xi'_0 = \phi(x))\mathbb{E}[\![\tau' | \xi'_1 = y]\!] \\ &\quad + (1 - np(1-p)^{n-1}) \cdot (1 - \pi_t(\mathcal{X}^*)), \end{aligned}$$

thus we have found a proper $\rho^u_t = (1 - np(1-p)^{n-1}) \cdot (1 - \pi_t(\mathcal{X}^*))$, and therefore,

$$\mathbb{E}[\![\tau | \xi_0 \sim \pi_0]\!] \leq \mathbb{E}[\![\tau' | \xi'_0 \sim \pi^{\phi}_0]\!]/(np(1-p)^{n-1}),$$

since $\sum_{t=0}^{+\infty}(1 - \pi_t(\mathcal{X}^*)) = \mathbb{E}[\![\tau | \xi_0 \sim \pi_0]\!]$. Moreover, for the reference process, we have $\mathbb{E}[\![\tau' | \xi'_0 \sim \pi^{\phi}_0]\!] \leq \mathbb{E}_{rls}(rn) = r^2n^2$. Finally, we get

$$\mathfrak{A}^u_{SA} = \mathbb{E}[\![\tau' | \xi'_0 \sim \pi^{\phi}_0]\!]/(np(1-p)^{n-1}) \leq r^2 n/(p(1-p)^{n-1}),$$

which proves the lemma. □

The upper bound $r^2n/(p(1-p)^{n-1})$ arrives at its minimum of $O(r^2n^2)$ at $p = 1/n$. We know now that the expected running time of the (1+1)-EA with mutation rate $1/n$ on the Discrete Linear problem with vocabulary size $r+1$ is $O(\min\{r^2n^2, rn\log n + r^3 n \log\log n\})$.

VIII. Discussion and Conclusion

This paper extends our preliminary attempt [33] and develops the switch analysis approach for running time analysis of evolutionary algorithms (EAs). Switch analysis compares two EA processes. In the comparison, we are able to eliminate the long-term behavior of one process, and need to compare only the one-step transition probabilities of the two processes. This allows us to derive the running time bounds of the target process by comparing with a simple reference process. As an example of using switch analysis, we give a reproof of the expected running time lower bound of mutation-based EAs on pseudo-Boolean functions with a unique global optimum, which extends our previous work [22] and has been partially proved in [27] and more generally in [32] using different techniques.

In order to investigate the relationship between general analysis approaches for EAs, we formally characterize these approaches and define the reducibility relationship. The reducibility is defined following the intuition that an approach is at least as powerful as another if it can derive no worse result using no more information. We have shown that the fitness level method and the drift analysis, the two major analysis approaches, are both reducible to switch analysis. On the opposite direction, by studying on the Peak problem, we have shown that switch analysis is not reducible to the fitness level method. By studying on the Discrete Linear Problem, we have shown that switch analysis is not reducible to a restricted version of drift analysis; and moreover, comparing with a recent running time upper bound $(1 + o(1))(e^c/c)rn \log n + O(r^3 n \log \log n)$ for mutation probability $p = c/n$ using multiplicative adaptive drift analysis [7], we derive another upper bound $r^2 n/(p(1-p)^{n-1})$ that is $(1 + o(1))(e^c/c)r^2 n^2$ for $p = c/n$, which is, although larger in term of n by a factor $n/\log n$, tighter in term of r. These results disclose the power of switch analysis for running time analysis of EAs.

If one has already obtained an analysis result using the fitness level method or the drift analysis, there could be a simple way to use switch analysis to further improve the result. Noticed that the proofs of the reducibility are constructive (Theorems 3 and 5), thus we can first transform the analysis process to be using switch analysis, and then try to replace some components of the switch analysis (such as the reference process or the mapping function) to improve the analysis result.

An important future work is to study the relationship between the switch analysis and the drift analysis in the continuous solution space situation. In the current paper, the solution space is discrete thus there are limited distinct distance function values. However, when the solution space is continuous, it is unknown if the reducibility from drift analysis to switch analysis is still invalid.

Acknowledgments

The authors want to 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) and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

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

[2] T. Chen, K. Tang, G. Chen, and X. Yao, "A large population size can be unhelpful in evolutionary algorithms," *Theoretical Computer Science*, vol. 436, no. 8, pp. 54–70, 2012.

[3] B. Doerr and L. A. Goldberg, "Adaptive drift analysis," *Algorithmica*, vol. 65, pp. 224–250, 2013.

[4] B. Doerr, D. Johannsen, and M. Schmidt, "Runtime analysis of the (1+1) evolutionary algorithm on strings over finite alphabets," in *Proceedings of the 11th International Workshop on Foundations of Genetic Algorithms (FOGA'11)*, Schwarzenberg, Austria, 2011, pp. 119–126.

[5] B. Doerr, D. Johannsen, and C. Winzen, "Multiplicative drift analysis," *Algorithmica*, vol. 64, pp. 673–697, 2012.

[6] ——, "Non-existence of linear universal drift functions," *Theoretical Computer Science*, vol. 436, pp. 71–86, 2012.

[7] B. Doerr and S. Pohl, "Run-time analysis of the (1+1) evolutionary algorithm optimizing linear functions over a finite alphabet," in *Proceedings of the 14th ACM Annual Conference on Genetic and Evolutionary Computation (GECCO'12)*, New York, NY, 2012, pp. 1317–1324.

[8] B. Doerr, C. Doerr, and F. Ebel, "Lessons from the black-box: fast crossover-based genetic algorithms," in *Proceedings of the 15th ACM Annual Conference on Genetic and Evolutionary Computation (GECCO'13)*, Amsterdam, The Netherlands, 2013, pp. 781–788.

[9] S. Droste, T. Jansen, and I. Wegener, "A rigorous complexity analysis of the (1+1) evolutionary algorithm for linear functions with Boolean inputs," *Evolutionary Computation*, vol. 6, no. 2, pp. 185–196, 1998.

[10] ——, "On the analysis of the (1+1) evolutionary algorithm," *Theoretical Computer Science*, vol. 276, no. 1-2, pp. 51–81, 2002.

[11] S. Fischera and I. Wegener, "The one-dimensional Ising model: Mutation versus recombination," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 208–225, 2005.

[12] B. Hajek, "Hitting-time and occupation-time bounds implied by drift analysis with applications," *Advances in Applied Probability*, vol. 14, no. 3, pp. 502–525, 1982.

[13] E. Happ, D. Johannsen, C. Klein, and F. Neumann, "Rigorous analyses of fitness-proportional selection for optimizing linear functions," in *Proceedings of the 10th ACM Annual Conference on Genetic and Evolutionary Computation (GECCO'08)*, Atlanta, GA, 2008, pp. 953–960.

[14] J. He and X. Yao, "Drift analysis and average time complexity of evolutionary algorithms," *Artifical Intelligence*, vol. 127, no. 1, pp. 57–85, 2001.

[15] ——, "From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 495–511, 2002.

[16] ——, "A study of drift analysis for estimating computation time of evolutionary algorithms," *Natural Computing*, vol. 3, no. 1, pp. 21–35, 2004.

[17] G. S. Hornby, A. Globus, D. S. Linden, and J. D. Lohn, "Automated antenna design with evolutionary algorithms," in *Proceedings of 2006 American Institute of Aeronautics and Astronautics Conference on Space*, San Jose, CA, 2006, pp. 19–21.

[18] T. Jansen and I. Wegener, "The analysis of evolutionary algorithms-a proof that crossover really can help," *Algorithmica*, vol. 34, no. 1, pp. 47–66, 2002.

[19] A. W. Marshall, I. Olkin, and B. Arnold, *Inequalities: Theory of Majorization and Its Applications*. second edition, Springer, 2011.
[20] J. R. Norris, *Markov Chains*. Cambridge, UK: Cambridge University Press, 1997.
[21] P. S. Oliveto and C. Witt, "Simplified drift analysis for proving lower bounds in evolutionary computation," *Algorithmica*, vol. 59, no. 3, pp. 369–386, 2011.
[22] C. Qian, Y. Yu, and Z.-H. Zhou, "On algorithm-dependent boundary case identification for problem classes," in *Proceedings of the 12th International Conference on Parallel Problem Solving from Nature (PPSN'12)*, Taormina, Italy, 2012, pp. 62–71.
[23] J. Richter, A. Wright, and J. Paxton, "Ignoble trails-where crossover is provably harmful," in *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature (PPSN'08)*, Dortmund, Germany, 2008, pp. 92–101.
[24] G. Sasaki and B. Hajek, "The time complexity of maximum matching by simulated annealing," *Journal of the ACM*, vol. 35, no. 2, pp. 387–403, 1988.
[25] A. Somani, P. P. Chakrabarti, and A. Patra, "An evolutionary algorithm-based approach to automated design of analog and RF circuits using adaptive normalized cost functions," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 336–353, 2007.
[26] D. Sudholt, "General lower bounds for the running time of evolutionary algorithms," in *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN'10)*, Krakow, Poland, 2010, pp. 124–133.
[27] ——, "A new method for lower bounds on the running time of evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 418–435, 2013.
[28] ——, "Crossover speeds up building-block assembly," in *Proceedings of the 14th ACM Annual Conference on Genetic and Evolutionary Computation (GECCO'12)*, Philadelphia, PA, 2012, pp. 689–702.
[29] J. Suzuki, "A markov chain analysis on simple genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 4, pp. 655–659, 1995.
[30] L.-Y. Tseng and S.-C. Chen, "Two-phase genetic local search algorithm for the multimode resource-constrained project scheduling problem," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 848–857, 2009.
[31] I. Wegener, "Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions," in *Evolutionary Optimization*, M. M. Ruhul A. Sarker and X. Yao, Eds. Kluwer, 2002.
[32] C. Witt, "Tight bounds on the optimization time of a randomized search heuristic on linear functions," *Combinatorics, Probability and Computing*, vol. 22, no. 2, pp. 294–318, 2013.
[33] Y. Yu, C. Qian, and Z.-H. Zhou, "Towards analyzing recombination operators in evolutionary search," in *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN'10)*, Krakow, Poland, 2010, pp. 144–153.
[34] Y. Yu and Z.-H. Zhou, "A new approach to estimating the expected first hitting time of evolutionary algorithms," *Artificial Intelligence*, vol. 172, no. 15, pp. 1809–1832, 2008.