Estimation of Hitting Time by Hitting Probability for Elitist Evolutionary Algorithms

Jun He *Senior Member, IEEE*, Siang Yew Chong and Xin Yao *Fellow, IEEE*

***Abstract*—Drift analysis is one of the main tools for analyzing the time complexity of evolutionary algorithms. However, it requires manual construction of drift functions to bound hitting time for each specific algorithm and problem. To address this limitation, linear drift functions were introduced for elitist evolutionary algorithms. But calculating good linear bound coefficients remains a problem. This paper proposes a new method called drift analysis of hitting probability to compute these coefficients. Each coefficient is interpreted as a bound on the hitting probability of a fitness level, transforming the task of estimating hitting time into estimating hitting probability. A new drift analysis method is then developed to estimate hitting probability, where paths are introduced to handle multimodal fitness landscapes. Explicit expressions are constructed to compute hitting probability, significantly simplifying the estimation process. An advantage of the proposed method is its ability to estimate both the lower and upper bounds of hitting time and to compare the performance of two algorithms in terms of hitting time. To demonstrate this application, two algorithms for the knapsack problem, each incorporating feasibility rules and greedy repair respectively, are compared. The analysis indicates that neither constraint handling technique consistently outperforms the other.**

***Index Terms*—evolutionary algorithms, hitting time, hitting probability, drift analysis, fitness levels**

I. Introduction

HITTING time is an important metric to evaluate the performance of evolutionary algorithms (EAs), referring to the minimum number of generations required for an EA to find the optimal solution. Drift analysis is one of the strongest tools used to analyze the hitting time of EAs [1], [2] and different drift analysis methods have been developed over the past two decades [3]–[9]. In drift analysis, a drift function is constructed to bound the hitting time, but it is manually tailored for each specific problem [5].

To overcome this limitation, the linear drift function for elitist EAs was proposed [10], which combines the strength of drift analysis with the convenience of fitness level partitioning. Given fitness levels $(S_0, \dots, S_K)$ from high to low, a lower bound on the hitting time from S_k to S_0 (where $1 \le k \le K$) is expressed as the following linear function.

$$\frac{1}{\max\limits_{X \in S_k} p(X, \cup_{j=0}^{k-1} S_j)} + \sum_{\ell=1}^{k-1} \frac{c_{k,\ell}}{\max\limits_{X \in S_\ell} p(X, \cup_{j=0}^{\ell-1} S_j)}, \tag{1}$$

Jun He is with the Department of Computer Science, Nottingham Trent University, Nottingham NG11 8NS, UK. Email: jun.he@ntu.ac.uk.
Siang Yew Chong is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and Centre of Artificial Intelligence and Optimization, School of Computer Science, University of Nottingham Ningbo China. Email: chongsy@sustech.edu.cn.
Xin Yao is with Lingnan University, Tuen Mun, Hong Kong. Email: xinyao@ln.edu.hk.

where the term $p(X, \cup_{j=0}^{\ell-1} S_j)$ represents the transition probability from $X \in S_\ell$ to levels $S_0 \cup \cdots \cup S_{\ell-1}$ and $c_{k,\ell} \in [0,1]$ is a linear coefficient. Similarly, an upper bound on the hitting time from S_k to S_0 is expressed as the following linear function.

$$\frac{1}{\min\limits_{X \in S_k} p(X, \cup_{j=0}^{k-1} S_j)} + \sum_{\ell=1}^{k-1} \frac{c_{k,\ell}}{\min\limits_{X \in S_\ell} p(X, \cup_{j=0}^{\ell-1} S_j)}. \tag{2}$$

The above drift functions are a family of linear functions used to bound the hitting time for elitist EAs. Although the calculation of transition probabilities is straightforward, determining the coefficients is more complex. The primary research question is how to effectively calculate the coefficients for tight linear bounds. Several methods for computing these coefficients have been proposed [10]–[13]. However, more efficient techniques are needed to determine coefficients for tight lower bounds [13], particularly on multimodal fitness landscapes with shortcuts [10]. While the linear bound coefficient has been interpreted in terms of visit probability [13], a general method for computing this visit probability is still lacking.

This paper aims to develop an efficient method for computing the coefficients in linear bounds (1) and (2). The method makes two significant contributions to the formal development and application of drift analysis. First, it reinterprets a coefficient as a bound on the hitting probability, which is the probability of reaching a fitness level for the first time. Consequently, the task of estimating the hitting time is transformed into estimating hitting probability.

Secondly, drift analysis of hitting probability is introduced to estimate the hitting probability or linear bound coefficients. Although the method is termed "drift analysis", it focuses on calculating hitting probabilities but not on hitting times. Hence, it is entirely different from the drift analysis of hitting time [5]. The method provides a new way to compute linear bound coefficients and introduces new explicit expressions for these coefficients. This greatly simplifies drift analysis because it allows direct estimation of hitting time using explicit formulas.

Comparing the performance of different EAs is crucial for empirical research. Since the proposed method can estimate both lower and upper bounds of hitting time, it provides a useful tool for theoretically comparing the performance of two EAs in terms of hitting time. The application is demonstrated through a case study that compares two EAs for the knapsack problem that incorporate feasibility rules [14], [15] and solution repair [16], [17], respectively.

This paper is structured as follows. Section II reviews related work. Section III provides preliminary definitions and results. Section IV interprets linear bound coefficients in terms of hitting probabilities. Section V develops a novel drift analysis method for computing hitting probabilities. Section VI describes the application of comparing two EAs. Finally, Section VII concludes the paper.

II. Related work

Since the introduction of drift analysis for bounding the hitting time of EAs [5], several variants have been developed, such as simplified drift analysis [6], multiplicative drift analysis [2], adaptive drift analysis [7], and variable drift analysis [8], [9]. A complete review of drift analysis can be found in [3], [4]. The main issue in drift analysis is the absence of some universal and explicit expression for the drift function that can be applied to various problems and EAs.

Recently, He and Zhou [10] proposed linear drift functions (1) and (2) designed for elitist EAs. Based on the coefficients in (1) and (2), they classified linear drift functions into three categories.

1) Type-c time bounds: for all $0 < \ell < k \leq K$, the coefficients $c_{k,\ell} = c$ are independent on k and ℓ.
2) Type-c_ℓ time bounds: for all $0 < \ell < k \leq K$, the coefficients $c_{k,\ell} = c_\ell$ depend on ℓ but not on k.
3) Type-$c_{k,\ell}$ time bounds: the coefficients $c_{k,\ell}$ depend on both k and ℓ.

Obviously, Type-c and Type-c_ℓ are special cases of the Type-$c_{k,\ell}$ bounds. Wegener [11] assigned the trivial constants $c_{k,\ell} = 0$ for the lower bound and $c_{k,\ell} = 1$ for the upper bound. Interestingly, assigning $c_{k,\ell} = 1$ provides a tight upper bound for many fitness functions, whereas assigning $c_{k,\ell} = 0$ usually leads to a loose lower bound. Several efforts have been made to improve the lower bound. Sudholt [12] investigated the non-trivial constant $c_{k,\ell} = c$ and used it to derive tight lower bounds for the (1+1) EA on various unimodal functions, including LeadingOnes, OneMax, and long k-paths. Sudholt referred to this constant c as viscosity. Doerr and Kötzing [13] significantly advanced this work by developing a Type-c_ℓ lower bound with $c_{k,\ell} = c_\ell$. They applied this to achieve tight lower bounds for the (1+1) EA on LeadingOnes, OneMax, and long k-paths jump functions, naming the coefficient c_ℓ a visit probability. However, Type-c and Type-c_ℓ lower bounds are loose on multimodal fitness landscapes with shortcuts [10].

To address the shortcut issue, He and Zhou [10] proposed drift analysis with fitness levels and developed the Type-$c_{k,\ell}$ linear bound, though this bound still necessitates recursive computation. Drift analysis with fitness levels has unified existing fitness level methods [11]–[13] within a single framework. The fitness level method can be viewed as a specific type of drift analysis that employs linear drift functions [10].

The study of hitting probability has received limited attention in the theory of EAs, with only a few studies available. The term hitting probability has been used in various contexts. He and Yao [18] and Chen et al. [19] used it to denote the probability of reaching an optimum among multiple optima. Jägersküpper [20] used it to describe the probability of a successful step. Yuen and Cheung [21] referred to it as "the first pass probability," indicating the probability that the hitting time does not exceed a threshold. Kötzing [22] also explored this type of "hitting probability" through negative drift analysis [6]. However, none of these studies relate to the explanation of the linear bound coefficients in (1) and (2).

III. Preliminaries

This section introduces several preliminary definitions and previous results.

A. The Markov Chain for the Elitist EA

Consider an EA designed to maximize a function $f(x)$, where $f(x)$ is defined over a finite set. The EA generates a sequence of solutions $(X^{[t]})_{t\geq 0}$, where $X^{[t]}$ represents the solution(s) at generation t. We model the sequence $(X^{[t]})_{t\geq 0}$ as a Markov chain, following the framework established in [23], [24] . This chain is hereafter referred to as a Markov chain related to the EA. Markov chain theory provides a solid foundation for analyzing the behavior and performance of EAs. Let $X \in S$ denote a state (a candidate solution), where S is the state space (all candidate solutions). Let $S_{\text{opt}} \subseteq S$ denote the subset of optimal solutions. We assume that the chain $(X^{[t]})_{t\geq 0}$ satisfies three key properties.

1) *Convergent (absorbing)*: Starting from any $X \in S$, the chain can reach (be absorbed into) the optimal set S_{opt} with probability 1.
2) *Homogeneous*: The transition probability $p(X, Y)$ from X to Y is independent on t.
3) *Elitist (increasing)*: Fitness values do not decrease. For any $t \geq 0$, $f(X^{[t+1]}) \geq f(X^{[t]})$, where $f(X)$ is the fitness of X.

B. Probability of Transition between Fitness Levels

The fitness level method utilizes the transition probabilities between fitness levels. A *fitness level partition* $(S_0, \dots, S_K)$ [11]–[13] is a partition of the state space S into fitness levels according to the fitness from high value to low such that:

1) The level S_0 is the optimal set S_{opt}.
2) For any pair of $X_k \in S_k$ and $X_{k+1} \in S_{k+1}$, the rank order holds: $f(X_k) > f(X_{k+1})$.

Thanks to the elitist property, the transition probability from $X_k \in S_k$ to S_ℓ (where $0 \leq \ell \leq K$) satisfies

$$p(X_k, S_\ell) = \begin{cases} \in [0,1] & \text{if } \ell \leq k, \\ 0 & \text{if } \ell > k. \end{cases} \tag{3}$$

Let $[i,j]$ denote the index set $\{i, i+1, \dots, j-1, j\}$ and $S_{[i,j]}$ denote the union of levels $S_i \cup \dots \cup S_j$. The transition probability from X_k to $S_{[i,j]}$ is denoted by $p(X_k, S_{[i,j]})$. The convergence property implies that the transition probability $p(X_k, S_{[0,k-1]}) > 0$ for any $k \geq 1$ and $X_k \in S_k$.

The transition probability from $X^{[t]} = X_k$ to $X^{[t+1]} \in S_\ell$ conditional on $X^{[t+1]} \notin S_k$ is denoted by

$$r(X_k, S_\ell) = \begin{cases} \frac{p(X_k, S_\ell)}{p(X_k, S_{[0,k-1]})} & \text{if } \ell < k, \\ 0 & \text{if } \ell \geq k. \end{cases} \tag{4}$$

C. Digraphs and Paths

Digraphs have been utilized to visualize the behavior of EAs [25]–[27]. In a digraph (V, A), the set V represents the vertices, where vertex k corresponds to level S_k. The set A represents the arcs, where arc (k, ℓ) indicates the transition from S_k to S_ℓ, provided that for some $X_k \in S_k$, $p(X_k, S_\ell) > 0$. Fig. 1 shows an example of a digraph.

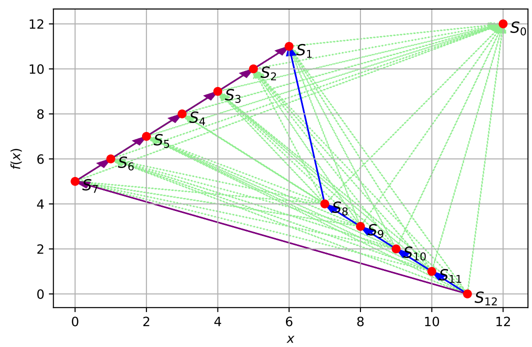


Fig. 1. The x-axis represents the state and the y-axis represents the fitness value. Each arc is a transition. Two paths from S_{12} to S_1 are highlighted.

A *path* from S_k to S_ℓ is a sequence of distinct vertices $k \to v_{m-1} \to \cdots \to v_1 \to \ell$ and each pair $v_{j+1} \to v_j$ is an arc. This path is denoted by $P[\ell, k]$. The special path $k \to k-1 \to \cdots \to \ell+1 \to \ell$ is abbreviated as $[\ell, k]$, which is the same as the index set. Fig. 1 shows two paths from S_{12} to S_1. Several sub-paths of the path $P[\ell, k]$ are represented as follows.

- $P[\ell, k) = P[\ell, k] \setminus \{k\}$ without vertex k.
- $P(\ell, k] = P[\ell, k] \setminus \{\ell\}$ without vertex ℓ.
- $P(\ell, k) = P[\ell, k] \setminus \{k, \ell\}$ without vertices k and ℓ.

D. Hitting Time and Hitting Probability

Given the Markov chain $(X^{[t]})_{t \geq 0}$ associated with an EA and a fitness level partition $(S_0, \cdots, S_K)$, we introduce the concepts of hitting time and hitting probability based on the textbook [28].

Definition 1: Assume that the Markov chain $(X^{[t]})_{t \geq 0}$ starts from $X_k \in S_k$, the first *hitting time* from X_k to S_ℓ (where $0 \leq \ell, k \leq K$) is

$$\tau(X_k, S_\ell) = \inf\{t : X^{[t]} \in S_\ell\}.$$

The *mean hitting time* $m(X_k, S_\ell)$ denotes the expected value of $\tau(X_k, S_\ell)$. The *hitting probability* to hit the set S_ℓ is

$$h(X_k, S_\ell) = \Pr(\tau(X_k, S_\ell) < \infty).$$

The *hitting probability* to hit a specific state Y_ℓ in S_ℓ is

$$h(X_k, Y_\ell) = \Pr(\tau(X_k, S_\ell) < +\infty \text{ and } X^{[\tau(X_k, S_\ell)]} = Y_\ell).$$

The following proposition adapts [28, Theorem 1.3.2] for the hitting probability $h(X_k, S_\ell)$. It states that the hitting probability from X_k to S_ℓ is composed of two parts: (i) transitioning from X_k to an intermediate state Y_i and (ii) transitioning from the intermediate state Y_i to S_ℓ.

Proposition 1: Given a fitness level partition $(S_0, \dots, S_K)$ and two levels $k, \ell : 0 \leq \ell < k \leq K$, the hitting probability $h(X_k, S_\ell)$ satisfies

$$\begin{cases} h(X_k, S_k) = 1. \\ h(X_k, S_\ell) = \sum_{i=\ell}^{k} \sum_{Y_i \in S_i} p(X_k, Y_i) h(Y_i, S_\ell). \end{cases}$$

The following proposition is a modification of [28, Theorem 1.3.5], adapted to the mean hitting time $m(X_k, \overline{S}_k)$, where $\overline{S}_k = S \setminus S_k$ is the complement of the set S_k. It states that the the mean hitting time $m(X_k, \overline{S}_k)$ consists of two components: (i) the transition from X_k to an intermediate state Y_k within S_k, and (ii) the subsequent transition from Y_k to a state outside S_k. The value of 1 is included to account for the initial step.

Proposition 2: Given a fitness level partition $(S_0, \dots, S_K)$ and a level $k : 0 < k \leq K$, the mean hitting time $m(X_k, \overline{S}_k)$ satisfies

$$m(X_k, \overline{S_k}) = 1 + \sum_{Y_k \in S_k} p(X_k, Y_k) m(Y_k, \overline{S_k}).$$

For clarity of notation, Table I presents abbreviations for the minimum and maximum values of transition probabilities, hitting probabilities, and mean hitting times.

TABLE I
ABBREVIATIONS FOR MINIMUM AND MAXIMUM VALUES

$p^{\min}_{S_k,S_\ell} := \min_{X_k \in S_k} p(X_k, S_\ell)$	$p^{\max}_{S_k,S_\ell} := \max_{X_k \in S_k} p(X_k, S_\ell)$
$r^{\min}_{S_k,S_\ell} := \min_{X_k \in S_k} r(X_k, S_\ell)$	$r^{\max}_{S_k,S_\ell} := \max_{X_k \in S_k} r(X_k, S_\ell)$
$h^{\min}_{S_k,S_\ell} := \min_{X_k \in S_k} h(X_k, S_\ell)$	$h^{\max}_{S_k,S_\ell} := \max_{X_k \in S_k} h(X_k, S_\ell)$
$m^{\min}_{S_k,S_\ell} := \min_{X_k \in S_k} m(X_k, S_\ell)$	$m^{\max}_{S_k,S_\ell} := \max_{X_k \in S_k} m(X_k, S_\ell)$

E. Lower and Upper Bounds on Hitting Time

Assuming that the chain $(X^{[t]})_{t \geq 0}$ starts from X, $m(X, S_0)$ is the mean hitting time from X to the optimal set S_0. If $d(X) \leq m(X, S_0)$, then $d(X)$ is called a lower bound of $m(X, S_0)$. Conversely, if $d(X) \geq m(X, S_0)$, then $d(X)$ is called an upper bound of $m(X, S_0)$.

Asymptotic notations such as O, Ω, and o are used to differentiate between tight and loose bounds, as described in the textbook [29]. Let n represent the dimension of the search space. The mean hitting time $m(X, S_0)$ is a function of n. A tight bound $d(X)$ differs from the mean hitting time $m(X, S_0)$ by only a constant factor, meaning $d(X) = O(m(X, S_0))$ for an upper bound and $d(X) = \Omega(m(X, S_0))$ for a lower bound.

F. Previous Results of Drift Analysis Using Linear Drift Functions

The main results of drift analysis using linear drift functions [10] are summarized in Proposition 3. A drift function $d(X)$ is used to approximate the mean hitting time $m(X, S_0)$ to the optimal set S_0.

Proposition 3: Given a fitness level partition $(S_0, \dots, S_K)$,

(1) Let a drift function $d(X)$ satisfy that for any $X_0 \in S_0$, $d(X_0) = 0$ and for $1 \le k \le K$ and any $X_k \in S_k$,

$$d(X_k) = \frac{1}{p^{\max}_{S_k, S_{[0,k-1]}}} + \sum_{\ell=1}^{k-1} \frac{c_{k,\ell}}{p^{\max}_{S_\ell, S_{[0,\ell-1]}}},$$

where coefficients $c_{k,\ell}$ satisfy $c_{\ell,\ell} = 1$ and for $k > \ell$,

$$c_{k,\ell} \le \min_{X_k \in S_k} \sum_{j=\ell}^{k-1} r(X_k, S_j) c_{j,\ell}. \tag{5}$$

Then for $k \ge 1$ and any $X_k \in S_k$, the drift

$$\Delta d(X_k) = d(X_k) - \sum_{i=0}^{K} \sum_{Y_i \in S_i} p(X_k, Y_i) d(Y_i) \le 1,$$

and the mean hitting time $m(X_k, S_0) \ge d(X_k)$.

(2) Let a drift function $d(X)$ satisfy that for any $X_0 \in S_0$, $d(X_0) = 0$ and for $1 \le k \le K$ and any $X_k \in S_k$,

$$d(X_k) = \frac{1}{p^{\min}_{S_k, S_{[0,k-1]}}} + \sum_{\ell=1}^{k-1} \frac{c_{k,\ell}}{p^{\min}_{S_\ell, S_{[0,\ell-1]}}},$$

where coefficients $c_{k,\ell}$ satisfy where coefficients $c_{k,\ell} \in [0, 1]$ satisfy $c_{\ell,\ell} = 1$ and for $k > \ell$,

$$c_{k,\ell} \ge \max_{X_k \in S_k} \sum_{j=\ell}^{k-1} r(X_k, S_j) c_{j,\ell}. \tag{6}$$

Then for $k \ge 1$ and any $X_k \in S_k$, the drift

$$\Delta d(X_k) = d(X_k) - \sum_{i=0}^{K} \sum_{Y_i \in S_i} p(X_k, Y_i) d(Y_i) \ge 1,$$

and the mean hitting time $m(X_k, S_0) \ge d(X_k)$.

In this paper, we interpret $c_{k,\ell}$ as the hitting probability from X_k to S_ℓ and introduce another drift analysis to estimate this hitting probability.

G. Previous Results in the Fitness Level Method

The Type-c linear bound is a special cases of the Type-$c_{k,\ell}$ bound by setting $c_{k,\ell} = c$ [10]. Sudholt [12] investigated Type-c time bounds. Given a random initial state $X^{[0]}$, he gave the lower time bound as follows:

$$\sum_{k=1}^{K} \Pr(X^{[0]} \in S_k) \left[\frac{1}{p^{\max}_{S_k, S_{[0,k-1]}}} + \sum_{\ell=1}^{k-1} \frac{c}{p^{\max}_{S_\ell, S_{[0,\ell-1]}}} \right].$$

where the coefficient c is calculated as follows:

$$c \le \min_{k:1<k\le K} \min_{\ell:1\le \ell<k} \min_{X_k: p(X_k, S_{[0,\ell]})>0} \frac{p(X_k, S_\ell)}{p(X_k, S_{[0,\ell]})}. \tag{7}$$

Sudholt [12] used a different expression, but it is equivalent to (7). The constant c is called viscosity, which is a lower bound on the probability of visiting S_ℓ conditional on visiting $S_{[0,\ell]}$.

The Type-c_ℓ linear bound is a special cases of the Type-$c_{k,\ell}$ bound by setting $c_{k,\ell} = c_\ell$ [10]. Doerr and Kötzing [13] investigated Type-c_ℓ time bounds. They gave the lower time bound as follows:

$$\sum_{\ell=1}^{K} \frac{c_\ell}{p^{\max}_{S_\ell, S_{[0,\ell-1]}}}.$$

where the coefficient c_ℓ is a lower bound on the probability of visiting S_ℓ at least once. It can be calculated as follows,

$$\begin{cases} c_\ell \le \min_{k:\ell<k\le K} \ \min_{X_k: p(X_k, S_{[0,\ell]})>0} \frac{p(X_k, S_\ell)}{p(X_k, S_{[0,\ell]})}. \\ c_\ell \le \frac{\Pr(X^{[0]} \in S_\ell)}{\Pr(X^{[0]} \in S_{[0,\ell]})}. \end{cases}$$

IV. Estimate Hitting time by Hitting Probability

This section explains the linear bound coefficient as the hitting probability between two fitness levels.

A. Exact Hitting Time

Given a fitness-level partition $(S_0, \ldots, S_K)$, the hitting time from a state $X_k \in S_k$ to the optimal set S_0 corresponds to the cumulative time the chain spends in the non-optimal set $S_1 \cup \cdots \cup S_k$. This intuition is rigorously captured in the following theorem.

Theorem 1: Given a fitness level partition $(S_0, \ldots, S_K)$ and a fitness level $k : 0 < k \le K$, the mean hitting time from $X_k \in S_k$ to the optimal set S_0 is equal to

$$m(X_k, S_0) = \sum_{\ell=1}^{k} \sum_{Y_\ell \in S_\ell} h(X_k, Y_\ell)\, m(Y_\ell, \overline{S}_\ell).$$

Proof: When the chain $(X^{[t]})_{t\ge 0}$ starts from $X_k \in S_k$, the probability that it first hits a state $Y_\ell \in S_\ell$ (for $\ell > 0$) is given by the hitting probability $h(X_k, Y_\ell)$. Upon reaching Y_ℓ, the expected time the chain stays in S_ℓ before transitioning to the set $\overline{S}_\ell$ is $m(Y_\ell, \overline{S}_\ell)$ where $\overline{S}_\ell = S_0 \cup \cdots \cup S_{\ell-1}$.

Due to the elitist property, once the chain exits S_ℓ, it cannot return. The total expected time that is spent in all non-optimal levels $S_1 \cup \cdots \cup S_k$ before reaching the absorbing set S_0 is

$$\sum_{\ell=1}^{k} \sum_{Y_\ell \in S_\ell} h(X_k, Y_\ell)\, m(Y_\ell, \overline{S}_\ell).$$

This cumulative time corresponds exactly to the mean hitting time $m(X_k, S_0)$, since the event of reaching S_0 for the first time is equivalent to the event of leaving the non-optimal set $S_1 \cup \cdots \cup S_k$ for the first time. ■

B. Lower and Upper Bounds on Hitting Time

Theorem 1 gives the exact hit time. However, in most cases, the exact hitting time cannot be calculated. Therefore, it is necessary to estimate upper and lower bounds. The following theorem gives linear upper and lower bounds on the hitting time based on the hitting probability.

Theorem 2: Given a fitness level partition $(S_0, \ldots, S_K)$ and a fitness level $k : 0 < k \le K$,

(1) The mean hitting time from $X_k \in S_k$ to the optimal set S_0 is lower-bounded by

$$m(X_k, S_0) \ge \sum_{\ell=1}^{k} \frac{h^{\min}_{S_k, S_\ell}}{p^{\max}_{S_\ell, S_{[0,\ell-1]}}}. \tag{8}$$

(2) The mean hitting time from $X_k \in S_k$ to the optimal set S_0 is upper-bounded by

$$m(X_k, S_0) \le \sum_{\ell=1}^{k} \frac{h^{\max}_{S_k, S_\ell}}{p^{\min}_{S_\ell, S_{[0,\ell-1]}}}. \tag{9}$$

Proof: 1) In the proof, the notation $X^\sharp_\ell \in S_\ell$ denotes a state such that $m(X^\sharp_\ell, \overline{S_\ell}) = m^{\min}_{S_\ell,\overline{S_\ell}}$. First, we estimate a lower bound on $m(X_\ell, \overline{S_\ell})$. According to Proposition 2, we have

$$\begin{aligned} m^{\min}_{S_\ell,\overline{S_\ell}} &= m(X^\sharp_\ell, \overline{S_\ell}) = 1 + \sum_{Y_\ell \in S_\ell} p(X^\sharp_\ell, Y_\ell) m(Y_\ell, \overline{S_\ell}) \\ &\geq 1 + p(X^\sharp_\ell, S_\ell) m^{\min}_{S_\ell,\overline{S_\ell}}. \end{aligned}$$

Then we get

$$m^{\min}_{S_\ell,\overline{S_\ell}} \geq \frac{1}{1 - p(X^\sharp_\ell, S_\ell)} = \frac{1}{p(X^\sharp_\ell, S_{[0,\ell-1]})} \geq \frac{1}{p^{\max}_{S_\ell, S_{[0,\ell-1]}}}. \tag{10}$$

Second, we estimate a lower bound on the mean hitting time $m(X_k, S_0)$ to the optimal set. According to Theorem 1,

$$\begin{aligned} m(X_k, S_0) &= \sum_{\ell=1}^{k} \sum_{Y_\ell \in S_\ell} h(X_k, Y_\ell) m(Y_\ell, \overline{S}_\ell) \\ &\geq \sum_{\ell=1}^{k} h^{\min}_{S_k,S_\ell} m^{\min}_{S_\ell,\overline{S_\ell}} \\ &\geq \sum_{\ell=1}^{k} \frac{h^{\min}_{S_k,S_\ell}}{p^{\max}_{S_\ell,S_{[0,\ell-1]}}} \quad \text{(by (10))}. \end{aligned}$$

Then we get Inequality (8).

2) The proof is similar to the first part. ■

The above theorems provide an explanation of the linear bound coefficients. The hitting probability $h^{\min}_{S_k,S_\ell}$ is a lower bound coefficient, and $h^{\max}_{S_k,S_\ell}$ is an upper bound coefficient. Any $c_{k,\ell} \leq h^{\min}_{S_k,S_\ell}$ is a lower bound coefficient, while any $c_{k,\ell} \geq h^{\max}_{S_k,S_\ell}$ an upper bound coefficient. Since $h(X_k, S_k) = 1$, the coefficient $c_{k,k}$ is always assigned to 1.

The terms viscosity [12], visit probability [13], and coefficient [10] fundamentally study the same subject: the probability of visiting a fitness level. Their distinctions can be characterized as viscosity for Type-c time bounds, visit probability for Type-c_ℓ bounds, and coefficient for Type-$c_{k,\ell}$ bounds. We use the term hitting probability as it aligns with the standard terminology found in the textbook [28].

V. Drift Analysis of Hitting Probability

This section outlines a drift analysis method for estimating hitting probabilities.

A. Drift Function

Computing exact values of hitting probabilities is challenging. In this paper, a new drift analysis method is proposed to estimate their bounds. A drift function $c(X_k, S_\ell)$ is used to approximate the hitting probability $h(X_k, S_\ell)$ from $X_k \in S_k$ to S_ℓ for any k, ℓ. It is designed specifically for elitist EAs based on fitness level partitioning.

Definition 2: Given a fitness level partition $(S_0, \dots, S_K)$, a *drift function* $c(X_k, S_\ell)$ (where $X_k \in S_k$ and $0 \leq k, \ell \leq K$) is a function such that

$$c(X_k, S_\ell) := c_{k,\ell} = \begin{cases} 0 & \text{if } \ell > k, \\ 1 & \text{if } \ell = k, \\ \in [0,1] & \text{if } \ell < k. \end{cases} \tag{11}$$

The above drift function makes use of two observations that (i) the hitting probability from S_k to S_ℓ (where $\ell > k$) is 0, and (ii) the hitting probability to the same level is 1.

Definition 3: Based on the conditional transition probability, the *conditional drift* from X_k to S_ℓ is

$$\begin{aligned} \tilde{\Delta} c(X_k, S_\ell) &:= c_{k,\ell} - \sum_{i=0}^{K} \sum_{Y_i \in S_i} r(X_k, Y_i) c_{i,\ell} \\ &= c_{k,\ell} - \sum_{i=\ell}^{k-1} r(X_k, S_i)\, c_{i,\ell}, \end{aligned} \tag{12}$$

The drift (12) does not contain the terms $i \leq \ell$ and $i \geq k$, since for $i \geq k$, $r(X_k, S_i) = 0$ and for $i < \ell$, $c_{i,\ell} = 0$.

B. Drift Conditions

The following theorem provides drift conditions to determine that a drift function is a lower or upper bound on the hitting probability.

Theorem 3: Given a fitness level partition $(S_0, \dots, S_K)$, a drift function (11) and two levels $\ell, k : 1 \leq \ell < k \leq K$,

(1) If for $\ell < j \leq k$ and any $X_j \in S_j$, the conditional drift $\tilde{\Delta} c(X_j, S_\ell) \leq 0$, equivalently, the coefficient

$$c_{j,\ell} \leq \min_{X_j \in S_j} \sum_{i=\ell}^{j-1} r(X_j, S_i) c_{i,\ell}, \tag{13}$$

then for any $\ell < j \leq k$, the coefficient $c_{j,\ell} \leq h^{\min}_{S_j,S_\ell}$.

(2) If for $\ell < j \leq k$ and any $X_j \in S_j$, the conditional drift $\tilde{\Delta} c(X_j, S_\ell) \geq 0$, equivalently, the coefficient

$$c_{j,\ell} \geq \max_{X_j \in S_j} \sum_{i=\ell}^{j-1} r(X_j, S_i) c_{i,\ell}, \tag{14}$$

then for any $\ell < j \leq k$, the coefficient $c_{j,\ell} \geq h^{\max}_{S_j,S_\ell}$.

Proof: 1) In the proof, the notation $X^\sharp_j \in S_j$ (where $\ell < j \leq k$) denotes the state such that $h(X^\sharp_j, S_\ell) = h^{\min}_{S_j,S_\ell}$.

First, we prove that for $\ell < j \leq k$, the state $X^\sharp_j$ satisfies the following inequality (15).

$$h^{\min}_{S_j,S_\ell} \geq \sum_{i=\ell}^{j-1} r(X^\sharp_j, S_i) h^{\min}_{S_i,S_\ell}. \tag{15}$$

By Proposition 1, the hitting probability from $X^\sharp_j$ to S_ℓ (where $\ell < j \leq k$) satisfies

$$\begin{aligned} h^{\min}_{S_j,S_\ell} &= h(X^\sharp_j, S_\ell) = \sum_{i=\ell}^{j} \sum_{Y_i \in S_i} p(X^\sharp_j, Y_i) h(Y_i, S_\ell) \\ &\geq \sum_{i=\ell}^{j} p(X^\sharp_j, S_i) h^{\min}_{S_i,S_\ell}. \end{aligned}$$

Moving the term $i = j$ from the right-hand side sum to the left, we obtain inequality (15)

$$h^{\min}_{S_j,S_\ell} \geq \sum_{i=\ell}^{j-1} \frac{p(X^\sharp_j, S_i)}{1 - p(X^\sharp_j, S_j)} h^{\min}_{S_i,S_\ell} = \sum_{i=\ell}^{j-1} r(X^\sharp_j, S_i) h^{\min}_{S_i,S_\ell}.$$

Next, using the inequality (15), we prove the first conclusion of the theorem, that is, for $j = \ell+1, \cdots, k$, $c_{j,\ell} \le h^{\min}_{S_j,S_\ell}$ by induction. Since $h^{\min}_{S_\ell,S_\ell} = 1$, we get

$$\begin{aligned} c_{\ell+1,\ell} &\le r(X^\sharp_{\ell+1}, S_\ell) \quad \text{(by (13))} \\ &= r(X^\sharp_{\ell+1}, S_\ell) h^{\min}_{S_\ell,S_\ell} \le h^{\min}_{S_{\ell+1},S_\ell}. \quad \text{(by (15))} \end{aligned}$$

We make an inductive assumption that for $i = \ell+1, \ldots, j$,

$$c_{i,\ell} \le h^{\min}_{S_i,S_\ell}. \tag{16}$$

Recall that $X^\sharp_{j+1}$ satisfies $h(X^\sharp_{j+1}, S_\ell) = h^{\min}_{S_{j+1},S_\ell}$. We get

$$\begin{aligned} c_{j+1,\ell} &\le \sum_{i=\ell}^{j} r(X^\sharp_{j+1}, S_i) c_{i,\ell} \quad \text{(by (13))} \\ &\le \sum_{i=\ell}^{j} r(X^\sharp_{j+1}, S_i) h^{\min}_{S_i,S_\ell} \quad \text{(by (16))} \\ &\le h^{\min}_{S_{j+1},S_\ell}. \quad \text{(by (15), replace } j \text{ by } j+1) \end{aligned}$$

Therefore, the inequality $c_{j+1,\ell} \le h^{\min}_{S_{\ell+1},S_\ell}$ holds. This completes the inductive step, and hence, by induction, the first conclusion is proven.

2) The proof is similar to the first part. ■

Interestingly, the recursive expressions (13) and (14) in the above theorem are identical to that in Proposition 3, despite being derived through entirely different proofs. But Theorem 3 makes a new contribution: drift analysis of hitting probability. A lower bound coefficient is a lower bound on the hitting probability, while an upper bound coefficient is an upper bound on the hitting probability. A drift function is used to bound the hitting probability, and the drift condition determines whether the drift function serves as a lower or upper bound.

Type-c and Type-c_ℓ bounds [10] can be rewritten in terms of necessary and sufficient drift conditions. A detailed analysis is given in the supplementary material. The drift conditions in Theorem 3 are based on pointwise drift. Average drift [30] can be used to handle random initialization.

C. Direct Calculation and Alternative Calculation

The coefficients in Theorem 3 are determined recursively. He and Zhou [10] gave explicit expressions for calculating the coefficients as follows.

Corollary 1: Given a fitness level partition $(S_0, \ldots, S_K)$ and two levels $\ell, k : 1 \le \ell < k \le K$,

(1) Let a drift function (11) (where $c(X_k, S_\ell) = c^{\min}_{k,\ell}$) satisfy that $c^{\min}_{\ell,\ell} = 1$ and for $1 \le \ell < j \le k$,

$$\begin{aligned} c^{\min}_{j,\ell} &= r^{\min}_{S_j,S_\ell} + \sum_{\ell<j_1<j} r^{\min}_{S_j,S_{j_1}} r^{\min}_{S_{j_1},S_\ell} \\ &\quad + \sum_{\ell<j_1<j_2<j} r^{\min}_{S_j,S_{j_2}} r^{\min}_{S_{j_2},S_{j_1}} r^{\min}_{S_{j_1},S_\ell} + \cdots \end{aligned}$$

Then $c^{\min}_{j,\ell} \le h^{\min}_{S_j,S_\ell}$.

(2) Let a drift function (11) (where $c(X_k, S_\ell) = c^{\max}_{k,\ell}$) satisfy that $c^{\max}_{\ell,\ell} = 1$ and for $1 \le \ell < j \le k$,

$$\begin{aligned} c^{\max}_{j,\ell} &= r^{\max}_{S_j,S_\ell} + \sum_{\ell<j_1<j} r^{\max}_{S_j,S_{j_1}} r^{\max}_{S_{j_1},S_\ell} \\ &\quad + \sum_{\ell<j_1<j_2<j} r^{\max}_{S_j,S_{j_2}} r^{\max}_{S_{j_2},S_{j_1}} r^{\max}_{S_{j_1},S_\ell} + \cdots \end{aligned}$$

Then $c^{\max}_{j,\ell} \ge h^{\max}_{S_j,S_\ell}$.

Proof: (1) By applying induction, it is straightforward to confirm that for $\ell < j \le k$, the coefficient $c^{\min}_{j,\ell}$ satisfies

$$c^{\min}_{j,\ell} = \sum_{i=\ell}^{j-1} r^{\min}_{S_j,S_i} c^{\min}_{i,\ell} \le \sum_{i=\ell}^{j-1} r(X_j, S_i) c^{\min}_{i,\ell}.$$

Consequently, according to Theorem 3.(1), for $\ell < j \le k$, we have $c^{\min}_{j,\ell} \le h^{\min}_{S_j,S_\ell}$.

(2) The proof is similar to the first part. ■

Each term in $c^{\min}_{j,\ell}$ or $c^{\max}_{j,\ell}$ represents the product of conditional probabilities of reaching S_ℓ along a path originating from S_j. The hitting probability is obtained by summing the conditional probabilities over all paths connecting S_j to S_ℓ.

In Theorem 3, coefficients are computed recursively in the direction from $\ell+1$ to k: $c_{\ell+1,\ell}, c_{\ell+2,\ell}, \ldots, c_{k,\ell}$. They can also be computed recursively in the opposite direction from $k-1$ to ℓ: $c_{k,k-1}, c_{k,k-2}, \ldots, c_{k,\ell}$, as shown below.

Theorem 4: Given a fitness level partition $(S_0, \ldots, S_K)$ and two levels $\ell, k : 1 \le \ell < k \le K$,

(1) Let a drift function (11) satisfy that for $1 \le \ell < j \le k$,

$$c_{j,\ell} \le \sum_{i=\ell+1}^{j} c_{j,i}\, r^{\min}_{S_i,S_\ell}, \tag{17}$$

then for any $\ell < j \le k$, the coefficient $c_{j,\ell} \le h^{\min}_{S_j,S_\ell}$.

(2) Let a drift function (11) satisfy that for $1 \le \ell < j \le k$,

$$c_{j,\ell} \ge \sum_{i=\ell+1}^{j} c_{j,i}\, r^{\max}_{S_i,S_\ell}, \tag{18}$$

then for any $\ell < j \le k$, the coefficient $c_{j,\ell} \ge h^{\max}_{S_j,S_\ell}$.

Proof: (1) By applying induction to (17), we can establish that $c_{j,\ell} \le c^{\min}_{j,\ell}$. Since $c^{\min}_{j,\ell} \le h^{\min}_{S_j,S_\ell}$, we arrive at the desired conclusion.

(2) The proof is similar to the first part. ■

D. Lower Bound Coefficients Using Paths

For multimodal fitness landscapes, there are multiple paths from one fitness level to another. For example, in Fig. 1, there are 11! paths from S_{12} to S_1. To calculate a lower bound coefficient $c_{k,\ell}$, it is sufficient to use one path $P[\ell, k]$ from S_k to S_ℓ, rather than using all paths from S_k to S_ℓ. For example, in Fig. 1, coefficient $c_{12,1}$ can be estimated using a longer path $S_{12} \to \cdots \to S_2 \to S_1$, or a shorter path $S_{12} \to \cdots \to S_8 \to S_1$. In this case, two values of $c_{12,1}$ can be generated, however, it suffices to utilize any one of them.

The following theorem uses a path to obtain the lower bound coefficient $c_{k,\ell}$. If vertex $j \in (\ell, k]$ is not on the path $P(\ell, k]$, then we directly assign the coefficient $c_{j,\ell} = 0$.

Theorem 5: Given a fitness level partition $(S_0,\dots,S_K)$ and two levels $\ell,k: 1\le \ell<k\le K$, let $P[\ell,k]$ be a path from k to ℓ. Let a drift function (11) satisfy that for $j\in(\ell,k]\setminus P(\ell,k]$, $c_{j,\ell}=0$, and for $j\in P(\ell,k]$,

$$c_{j,\ell}\le \min_{X_j\in S_j}\sum_{i\in P[\ell,j)} r(X_j,S_i)\,c_{i,\ell}. \tag{19}$$

Then for $\ell<j\le k$, coefficient $c_{j,\ell}\le h^{\min}_{S_j,S_\ell}$.

Proof: For $j\in(\ell,k]\setminus P(\ell,k]$, since $c_{j,\ell}=0$, the conditional drift $\tilde{\Delta}c(X_j,S_\ell)\le 0$. For $j\in P(\ell,k]$, the conditional drift $\tilde{\Delta}c(X_j,S_\ell)\le 0$ by (19). According to Theorem 3.(1), we get the conclusion. ■

To avoid recursive computation in (19), the following corollary provides a non-recursive formula to compute the coefficients. It is a path-based version of [27, Theorem 4].

Corollary 2: Given a fitness level partition $(S_0,\dots,S_K)$ and two levels $\ell,k: 1\le \ell<k\le K$, let $P[\ell,k]$ be a path from k to ℓ. Let a drift function (11) satisfy that for $j\in(\ell,k]\setminus P(\ell,k]$, $c_{j,\ell}=0$, and for $j\in P(\ell,k]$,

$$c_{j,\ell}=\prod_{i\in P(\ell,j]} r^{\min}_{S_i,S_{P[\ell,i)}}, \tag{20}$$

then for $\ell<j\le k$, coefficient $c_{j,\ell}\le h^{\min}_{S_j,S_\ell}$.

Proof: From the product (20), we get

$$\begin{aligned}\min_{i\in P(\ell,j)} c_{i,\ell} &= \prod_{i\in P(\ell,j)} r^{\min}_{S_i,S_{P[\ell,i)}},\\ c_{j,\ell} &= r^{\min}_{S_i,S_{P[\ell,i)}}\min_{i\in P(\ell,j)} c_{i,\ell}.\end{aligned} \tag{21}$$

For $j\in P(\ell,k]$,

$$\begin{aligned}&\min_{X_j\in S_j}\sum_{i\in P[\ell,j)} r(X_j,S_i)\,c_{i,\ell}\\ &\ge \sum_{i\in P[\ell,j)} r^{\min}_{S_j,S_i}\min_{i\in P(\ell,j)} c_{i,\ell}\\ &= r^{\min}_{S_j,S_{P[\ell,j)}}\min_{i\in P(\ell,j)} c_{i,\ell}=c_{j,\ell}\quad \text{(by (21))}.\end{aligned}$$

According to Theorem 5, we get the conclusion. ■

In (20), the transition probability $r^{\min}_{S_i,S_{P[\ell,i)}}$ corresponds to the transition from S_i to the path $P[\ell,i)$. An intuitive interpretation of this corollary is that the hitting probability $h^{\min}_{S_k,S_\ell}$ is lower-bounded by the product of the conditional probabilities of staying on the path $P(\ell,k)$.

E. Upper Bound Coefficients Using Paths

Intuitively, it seems impossible to obtain an upper bound coefficient $c_{k,\ell}$ using a path, since the hitting probability of going from S_k to S_ℓ is not less than that of going from S_k to S_ℓ via a path $P[\ell,k]$. Counterintuitively, the following theorem establishes an upper bound on the hitting probability using one path. If vertex $i\in[\ell,k]$ is not on the path $P(\ell,k]$, we simply assign the coefficient $c_{i,\ell}=1$.

Theorem 6: Given a fitness level partition $(S_0,\dots,S_K)$ and two levels $\ell,k: 1\le \ell<k\le K$, let $P[\ell,k]$ be a path from k to ℓ. Let a drift function (11) satisfy that for $j\in[\ell,k]\setminus P(\ell,k]$, $c_{j,\ell}=1$, and for $j\in P(\ell,k]$,

$$c_{j,\ell}\ge \max_{X_j\in S_j}\left\{ r(X_j,S_{[\ell,j)\setminus P(\ell,j)})+\sum_{i\in P(\ell,j)} r(X_j,S_i)c_{i,\ell}\right\}, \tag{22}$$

then for $\ell<j\le k$, the coefficient $c_{j,\ell}\ge h^{\max}_{S_j,S_\ell}$.

Proof: For $j\in(\ell,k]\setminus P(\ell,k]$, since $c_{j,\ell}=1$, the conditional drift $\tilde{\Delta}c(X_j,S_\ell)\ge 0$. For $j\in P(\ell,k]$, the conditional drift $\tilde{\Delta}c(X_j,S_\ell)\ge 0$ by (22). According to Theorem 3.(2), we get the conclusion. ■

The coefficient computation in Theorem 6 is recursive. The following corollary provides a non-recursive formula to compute the coefficients.

Corollary 3: Given a fitness level partition $(S_0,\dots,S_K)$ and two levels $\ell,k: 1\le \ell<k\le K$, let $P[\ell,k]$ be a path from k to ℓ. Let a drift function (11) satisfy that for $j\in(\ell,k]\setminus P(\ell,k]$, $c_{j,\ell}=1$, and for $j\in P(\ell,k]$,

$$c_{j,\ell}=\sum_{i\in P(\ell,j]} r^{\max}_{S_i,S_{[\ell,i)\setminus P(\ell,i)}}, \tag{23}$$

then for $\ell<j\le k$, the coefficient $c_{j,\ell}\ge h^{\max}_{S_j,S_\ell}$.

Proof: From the sum (23), we get

$$\begin{aligned}\max_{i\in P(\ell,j)} c_{i,\ell} &= \sum_{i\in P(\ell,j)} r^{\max}_{S_i,S_{[\ell,i)\setminus P(\ell,i)}},\\ c_{j,\ell} &= r^{\max}_{S_j,S_{[\ell,j)\setminus P(\ell,j)}}+\max_{i\in P(\ell,j)} c_{i,\ell}.\end{aligned} \tag{24}$$

For $j\in P(\ell,k]$,

$$\begin{aligned}&\max_{X_j\in S_j}\left\{ r(X_j,S_{[\ell,j)\setminus P(\ell,j)})+\sum_{i\in P(\ell,j)} r(X_j,S_i)c_{i,\ell}\right\}\\ &\le r^{\max}_{S_j,S_{[\ell,j)\setminus P(\ell,j)}}+r^{\max}_{S_j,S_{P(\ell,j)}}\max_{i\in P(\ell,j)} c_{i,\ell}\\ &\le r^{\max}_{S_j,S_{[\ell,j)\setminus P(\ell,j)}}+\max_{i\in P(\ell,j)} c_{i,\ell}=c_{j,\ell}\quad \text{(by (24))}.\end{aligned}$$

According to Theorem 6, we get the conclusion. ■

In (23), the conditional probability $r^{\max}_{S_i,S_{[\ell,i)\setminus P(\ell,i)}}$ corresponds to the transitions from S_i to $[\ell,i)\setminus P(\ell,i)$, the vertices not on the path $P(\ell,j)$. An intuitive interpretation of this corollary is that the hitting probability $h^{\max}_{S_k,S_\ell}$ is upper-bounded by the sum of the conditional probabilities of leaving the path $P(\ell,k)$ to $[\ell,k)\setminus P(\ell,k)$.

VI. Comparison of Two Algorithms

This section applies the proposed method to a comparative analysis of two EAs for the knapsack problem.

A. Comparison of Two EAs

In computer experiments, the performance of two EAs is assessed using a benchmark suite that includes both easy and hard problems [31]. Similarly, theoretical studies should compare EAs using a benchmark suite.

In this paper, three instances of the knapsack problem are designed to serve as a benchmark suite for comparison. They represent both easy and hard scenarios. The knapsack problem is chosen because of its NP-complete complexity and its well-established role as a classic problem for explaining EAs [32]. Unlike common benchmarks such as OneMax and LeadingOnes [12], [13], the knapsack problem is subject to a constraint. Various constraint-handling techniques have been employed in EAs, such as feasibility rules [14], [15], the penalty method [33], solution repair [16], [17].

In this section, we compare an EA employing feasibility rules (algorithm 1) with another EA employing solution repair (algorithm 2). To evaluate their performance, we examine the ratio of their mean hitting times (speedup), defined as

$$\frac{\text{mean hitting time of algorithm 1}}{\text{mean hitting time of algorithm 2}}.$$

B. The Knapsack Problem

The knapsack problem is described as follows. There are n items, each with a specific weight w_i and value v_i. The goal is to select a subset of these items to include in the knapsack, ensuring that the total weight does not exceed the capacity of the knapsack C while maximizing the overall value. For the ith item, let $b_i = 1$ indicate that the item is included in the backpack, and $b_i = 0$ indicate that the item is not included in the backpack. The knapsack problem can be expressed as a constrained optimization problem. Let $x = (b_1, \dots, b_n)$.

$$\max f(x) = \sum_{i=1}^{n} v_i b_i \qquad \text{subject to } \sum_{i=1}^{n} w_i b_i \le C. \tag{25}$$

The first EA is the (1+1) EA using feasibility rules, which is described in Algorithm 1. The (1+1) EA is chosen because it serves as a common baseline in the theoretical analysis of EAs [12], [13]. Its purpose is to avoid complex calculations of transition probabilities, allowing a focus on the analysis itself. We assume that the EA is initialized with an empty knapsack; however, other initialization strategies can also be considered. According to feasibility rules, an infeasible solution will not be accepted because it is worse than the empty knapsack.

Algorithm 1 The (1+1) EA with Feasibility Rules

Specify $X^{[0]} = x$ to be the empty knapsack.
for $t = 1, 2, \dots$ **do**
  Flip each bit of x independently with probability $1/n$ to generate a solution y.
  if both x and y are feasible **then**
    Select the one with the larger objective value f as $X^{[t+1]}$.
  else if both x and y are infeasible **then**
    Select the one with the smaller constraint violation value $\sum_i w_i b_i - C$ as $X^{[t+1]}$;
  else
    Select the feasible one as $X^{[t+1]}$.
  end if
end for

The second EA is the (1+1) EA using greedy repair, which is described in Algorithm 2. Greedy repair transforms an infeasible knapsack into a feasible one by removing the item(s) with the smallest value-to-weight ratio. Therefore, it is sufficient to consider feasible solutions. The fitness function is the objective function $f(x)$.

Algorithm 2 The (1+1) EA with Greedy Repair

Specify $X^{[0]} = x$ to be the empty knapsack.
for $t = 1, 2, \dots$ **do**
  Flip each bit of x independently with probability $1/n$ to generate a solution y.
  while y is infeasible (weight exceeds capacity) **do**
    Select an item with the smallest value-to-weight ratio and remove it from the knapsack.
  end while
  if $f(y) \ge f(x)$ **then**
    $X^{[t+1]} = y$;
  else
    $X^{[t+1]} = x$.
  end if
end for

Table II presents three knapsack problem instances with different optimal solutions. In every instance, there are two high-value, high-weight items and $n-2$ low-value, low-weight items. Item 1 has the highest value-to-weight ratio, exceeding 1, while Item 2 has the lowest ratio, falling below 1. The remaining items all have a value-to-weight ratio equals to 1.

To facilitate analysis, the fitness levels in these knapsack problem instances are expressed in the following form:

$$L_{(b_1,b_2;k)} = \{x = (b_1, \dots, b_n); k = b_3 + \dots + b_n\}.$$
$$L_{(b_1,b_2;[i,j])} = \{x = (b_1, \dots, b_n); i \le b_3 + \dots + b_n \le j\}.$$

Let $(b_1, b_2; k)$ denote a solution in $L_{(b_1,b_2;k)}$ and $L^+_{(b_1,b_2;k)}$ denote the set of feasible solutions with a fitness value larger than $f(b_1, b_2; k)$.

The hitting probability from $(a_1, a_2; i)$ to $L_{(b_1,b_2;k)}$ is denoted as $h_{(a_1,a_2;i),(b_1,b_2;k)}$, and its linear bound coefficient is denoted as $c_{(a_1,a_2;i),(b_1,b_2;k)}$. Similarly, $r_{(a_1,a_2;i),(a_1,a_2;i)^+}$ denotes the conditional probability from $(a_1, a_2; i)$ to $L^+_{(a_1,a_2;i)}$, while $m_{(a_1,a_2;i),(b_1,b_2;k)}$ denotes the mean hitting time from $(a_1, a_2; i)$ to $L_{(b_1,b_2;k)}$. Using the notation, it is convenient to calculate transition probabilities. For example, consider the (1+1) EA with feasibility rules on Instance KP1 and the transition from $L_{(0,0;0)}$ to $L_{(0,0;n-3)}$. This transition happens if and only if bits b_1, b_2 remain unchanged, $n-3$ of the $n-2$ zero-valued bits in $b_3, \dots, b_n$ flips, and the other bits remain unchanged. Therefore, the transition probability

$$p_{(0,0;0),(0,0;n-3)} = \left(1 - \frac{1}{n}\right)^2 \binom{n-2}{n-3} \left(\frac{1}{n}\right)^{n-3} \left(1 - \frac{1}{n}\right).$$

C. Instance KP1

Fig. 2 shows the digraph of the two (1+1) EAs on Instance KP1.

1) The (1+1) EA Using Feasibility Rules: It is sufficient to consider feasible solutions because infeasible solutions are worse than the empty knapsack. According to the lower bound

TABLE II
KNAPSACK PROBLEM INSTANCES.

ID	item i	1	2	$3,\ldots,n$	capacity C	global optimum	local optimum
KP1	value v_i weight w_i	$n-2$ $n-2-2/3$	$n/2-1/3$ $n-3$	1 1	$n-2$	$L_{(1,0;0)}$ and $L_{(0,0;n-2)}$	$L_{(0,1;1)}$
KP2	value v_i weight w_i	$n-2-1/3$ $n-2-2/3$	$n/2-1/3$ $n-3$	1 1	$n-2$	$L_{(0,0;n-2)}$	$L_{(0,1;1)}$ and $L_{(1,0;0)}$
KP3	value v_i weight w_i	$n-1$ $n-2-2/3$	$n/2-1/3$ $n-3$	1 1	$n-2$	$L_{(1,0;0)}$	$L_{(0,1;1)}$ and $L_{(0,0;n-2)}$

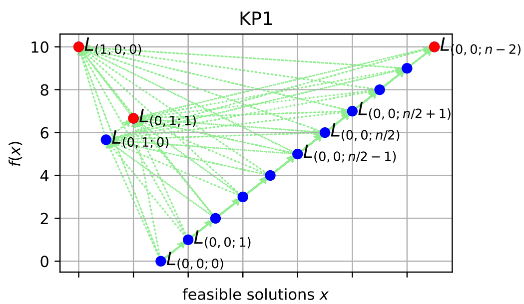


Fig. 2. The digraph of the two (1+1) EAs on Instance KP1. Vertices represent fitness levels (feasible solution area). Arcs represent transitions. $n = 12$.

(8) in Theorem 2, the mean hitting time from the empty knapsack to the global optimal set $L_{(0,0;n-2)} \cup L_{(1,0;0)}$ is

$$m_{(0,0;0),(0,0;n-2)\cup(1,0;0)} \geq \frac{h_{(0,0;0),(0,1;1)}}{p_{(0,1;1),(0,1;1)^+}}. \quad (26)$$

Since $L^+_{(0,1;1)} = L_{(1,0;0)} \cup L_{(0,0;[n/2+2,n-2])}$, the transition probabilities

$$p_{(0,1;1),(1,0;0)} \leq \left(\frac{1}{n}\right)^3,$$

$$p_{(0,1;1),(0,0;[n/2+2,n-2])} \leq \frac{1}{n}\binom{n-3}{n/2+1}\left(\frac{1}{n}\right)^{n/2+1},$$

and then the transition probability

$$p_{(0,1;1),(0,1;1)^+} \leq \left(\frac{1}{n}\right)^3 + \frac{1}{n}\binom{n-3}{n/2+1}\left(\frac{1}{n}\right)^{n/2+1} = O(n^{-3}).$$

Thus, the mean hitting time

$$m_{(0,0;0),(0,0;n-2)\cup(1,0;0)} \geq \Omega(n^3)\, h_{(0,0;0),(0,1;1)}. \quad (27)$$

We compute the hitting probability $h_{(0,0;0),(0,1;1)}$ using the path $L_{(0,0;0)} \to L_{(0,1;1)}$. An intuitive observation is that the probability of hitting $L_{(0,1;1)}$ is $\Omega(\frac{1}{n})$ because of flipping bit b_2 and flipping one of bits in $[b_3, b_n]$. Strictly speaking, the hitting probability

$$h_{(0,0;0),(0,1;1)} \geq p_{(0,0;0),(0,1;1)} \geq \frac{1}{n}\binom{n-2}{1}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-2} = \Omega\left(\frac{1}{n}\right).$$

Thus, the mean hitting time from the empty knapsack to the global optimal set $L_{(0,0;n-2)} \cup L_{(1,0;0)}$ is

$$m_{(0,0;0),(0,0;n-2)\cup(1,0;0)} = \Omega(n^3)\Omega(n^{-1}) = \Omega(n^2). \quad (28)$$

2) The (1+1) EA with Greedy Repair: Let S_0 denote the global optimal set $L_{(0,0;n-2)} \cup L_{(1,0;0)}$, and S_1 be the rest of feasible solutions. According to the upper bound (9) in Theorem 2, the mean hitting time from any $x_1 \in S_1$ to S_0 is upper-bounded by

$$\frac{1}{\min_{x_1 \in S_1} p(x_1; S_0)}.$$

For any $x_1 = (0, b_2; k) \in S_1$, the probability of a mutation from $(0, b_2; k)$ to $(1, *; *)$ is $\frac{1}{n}$ (where $*$ represents an arbitrary value). Since Item 1 has the largest value-to-weight ratio, after greedy repair, only item 1 remains and the solution becomes $(1, 0; 0)$. So the probability $p(x_1; S_0)$ is at least $\frac{1}{n}$. Then, the mean hitting time from the empty knapsack to the global optimum $(1, 0; 0)$ is

$$m_{(0,0;0),(1,0;0)} = O(n). \quad (29)$$

By comparing equations (28) and (29), we find that for KP1, the (1+1) EA with greedy repair is faster than that of the (1+1) EA using feasibility rules by a factor of $\Omega(n)$.

D. Instance KP2

Fig. 3 shows the digraph of the two (1+1) EAs on Instance KP2.

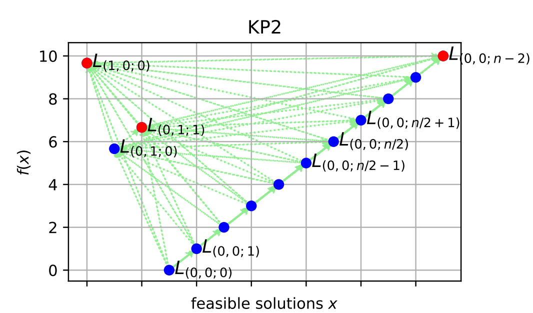


Fig. 3. The digraph of the two (1+1) EAs on Instance KP2. Vertices represent fitness levels (feasible solution area). Arcs represent transitions. $n = 12$.

1) The (1+1) EA Using Feasibility Rules: According to the upper bound (9) in Theorem 2, the mean hitting time from the empty knapsack to the global optimum is

$$m_{(0,0;0),(0,0;n-2)} \leq \sum_{\ell=0}^{n-3} \frac{1}{p_{(0,0;\ell),(0,0;\ell)^+}} + \frac{1}{p_{(0,1;0),(0,1;0)^+}} + \frac{1}{p_{(0,1;1),(0,1;1)^+}} + \frac{h_{(0,0;0),(1,0;0)}}{p_{(1,0;0),(1,0;0)^+}}. \tag{30}$$

The transition probabilities in (30) are estimated as follows. For $\ell = 0, \ldots, n-3$, the transition probabilities

$$p_{(0,0;\ell),(0,0;\ell)^+} \geq p_{(0,0;\ell),(0,0;\ell+1)} \geq \frac{n-2-\ell}{n}\left(1-\frac{1}{n}\right)^{n-1}.$$

$$p_{(0,1;0),(0,1;0)^+} \geq p_{(0,1;0),(0,1;1)} = \frac{n-2}{n}\left(1-\frac{1}{n}\right)^{n-1}.$$

$$p_{(0,1;1),(0,1;1)^+} \geq p_{(0,1;1),(1,0;0)} \geq \left(\frac{1}{n}\right)^3\left(1-\frac{1}{n}\right)^{n-3}.$$

$$p_{(1,0;0),(1,0;0)^+} \geq p_{(1,0;0),(0,0;n-2)} \geq \left(\frac{1}{n}\right)^{n-1}\left(1-\frac{1}{n}\right).$$

Thus, we get the mean hitting time

$$m_{(0,0;0),(0,0;n-2)} \leq O(n^3) + \frac{h_{(0,0;0),(1,0;0)}}{(n-1)n^{-n}}. \tag{31}$$

Intuitively, the hitting probability $h_{(0,0;0),(1,0;0)}$ is no more than the sum of the conditional transition probabilities from $L_{(0,0;\ell)}$ to $L_{(1,0;0)} \cup L_{(0,1;[0;1])}$ where $\ell = 0, \ldots, n-3$. Since these conditional probabilities decreases exponentially fast as ℓ increases, the hitting probability $h_{(0,0;0),(1,0;0)} = O(\frac{1}{n})$. We rigorously prove this using Corollary 3.

We choose the path $L_{(0,0;0)} \to L_{(0,0;1)} \to L_{(0,0;2)} \to \cdots \to L_{(0,0;n-3)} \to L_{(1,0;0)}$ to calculate the hitting probability $h_{(0,0;0),(1,0;0)}$. According to Corollary 3, the hitting probability

$$\begin{aligned} h_{(0,0;0),(1,0;0)} &\leq \sum_{\ell=0}^{n-3} r_{(0,0;\ell),(1,0;0)\cup(0,1;[0,1])} \\ &= \sum_{\ell=0}^{n-3} \frac{p_{(0,0;\ell),(1,0;0)} + p_{(0,0;\ell),(0,1;0)} + p_{(0,0;\ell),(0,1;1)}}{p_{(0,0;\ell),(0,0;\ell)^+}} \\ &\leq \sum_{\ell=0}^{n-3} \frac{p_{(0,0;\ell),(1,0;0)} + p_{(0,0;\ell),(0,1;0)} + p_{(0,0;\ell),(0,1;1)}}{p_{(0,0;\ell),(0,0;\ell+1)}}. \end{aligned} \tag{32}$$

The transition probabilities

$$p_{(0,0;\ell),(1,0;0)} \leq \left(\frac{1}{n}\right)^{\ell+1}.$$

$$p_{(0,0;\ell),(0,1;0)} \leq \left(\frac{1}{n}\right)^{\ell+1}.$$

$$p_{(0,0;\ell),(0,1;1)} \leq \begin{cases} \frac{1}{n}, & \ell \leq 1, \\ \frac{1}{n}\binom{\ell}{\ell-1}\left(\frac{1}{n}\right)^{\ell}, & \ell \geq 2. \end{cases}$$

$$p_{(0,0;\ell),(0,0;\ell+1)} \geq \frac{n-2-\ell}{n}\left(1-\frac{1}{n}\right)^{n-1}.$$

Substituting them into (32), we get the hitting probability

$$\begin{aligned} h_{(0,0;0),(0,1;0)} &\leq O\left(\frac{1}{n}\right) + \sum_{\ell=2}^{n-3} \frac{e}{n-2-\ell}\left(\frac{2}{n^\ell} + \frac{\ell}{n^\ell}\right) \\ &= O\left(\frac{1}{n}\right). \end{aligned} \tag{33}$$

Inserting (33) into (31), we get the mean hitting time $m_{(0,0;0),(0,0;n-2)}$ is upper-bounded by

$$m_{(0,0;0),(0,0;n-2)} = \frac{O(n^{-1})}{(n-1)n^{-n}}. \tag{34}$$

2) The (1+1) EA Using Greedy Repair: According to the lower bound (8) in Theorem 2, the mean hitting time from the empty knapsack to the global optimum is

$$m_{(0,0;0),(0,0;n-2)} \geq \frac{h_{(0,0;0),(1,0;0)}}{p_{(1,0;0),(1,0;0)^+}}. \tag{35}$$

Since $L^+_{(1,0;0)} = L_{(0,0;n-2)}$, the transition probability

$$p_{(1,0;0),(0,0;n-2)} \leq \left(1-\frac{1}{n}\right)\left(\frac{1}{n}\right)^{n-1}. \tag{36}$$

Then, the mean hitting time

$$m_{(0,0;0),(0,0;n-2)} \geq \frac{h_{(0,0;0),(1,0;0)}}{(n-1)n^{-n}}. \tag{37}$$

Intuitively, the chain follows the path $(0,0;0) \to (0,0;1) \to \cdots \to (0,0;n/2-1)$ with probability $\Omega(1)$. For each vertex on this path, the mutation probability from $(0,0;\ell)$ (where $\ell = 0, \ldots, n/2-1$) to $(1,*;*)$ is $\frac{1}{n}$. After greedy repair, the solution becomes $(1,0;0)$. Thus, the hitting probability $h_{(0,0;0),(1,0;0)}$ is no less than the sum of $\Omega(\frac{1}{n})$ over $\ell = 0, \ldots, n/2-1$. Then the hitting probability $h_{(0,0;0),(1,0;0)} = \Omega(1)$. We rigorously prove this using Theorem 4.

According to (17) in Theorem 4, we have a lower bound on the hitting probability as

$$\begin{aligned} h_{(0,0;0),(1,0;0)} &\geq c_{(0,0;0),(1,0;0)} \\ &= r_{(0,0;0),(1,0;0)} + \sum_{\ell=1}^{n/2-1} c_{(0,0;0),(0,0;\ell)}\, r_{(0,0;\ell),(1,0;0)}. \end{aligned} \tag{38}$$

We omit the terms with $\ell \geq \frac{n}{2}$ from the summation, as excluding them still yields a valid lower bound.

The conditional transition probability $r_{(0,0;\ell),(1,0;0)}$ (where $0 \leq \ell < n/2$) is calculated as follows. The mutation probability from $(0,0;\ell)$ to $(1,*;*)$ is $(1-\frac{1}{n})\frac{1}{n}$ (where $*$ represents an arbitrary value). Since Item 1 has the largest value-to-weight ratio, after greedy repair, only Item 1 remains. The solution becomes $(1,0;0)$. Thus, we get

$$r_{(0,0;\ell),(1,0;0)} \geq p_{(0,0;\ell),(1,0;0)} = \Omega\left(\frac{1}{n}\right).$$

Then the lower bound coefficient

$$c_{(0,0;0),(1,0;0)} = \Omega\left(\frac{1}{n}\right) + \Omega\left(\frac{1}{n}\right)\sum_{\ell=1}^{n/2-1} c_{(0,0;0),(0,0;\ell)}. \tag{39}$$

The lower bound coefficient $c_{(0,0;0),(0,0;\ell)}$ is calculated using Corollary 2. By (20) in Corollary 2, we assign

$$\begin{aligned} c_{(0,0;0),(0,0;\ell)} &= \prod_{j=0}^{\ell-1} r_{(0,0;j),(0,0;(j,\ell])} = \prod_{j=0}^{\ell-1} \frac{p_{(0,0;j),(0,0;(j,\ell])}}{p_{(0,0;j),(0,0;j)^+}} \\ &= \prod_{j=0}^{\ell-1} \frac{p_{(0,0;j),(0,0;(j,\ell])}}{p_{(0,0;j),(0,0;(j,n-2])} + p_{(0,0;j),(1,0;0)} + p_{(0,0;j),(0,1;[0,1])}} \\ &= \prod_{j=0}^{\ell-1} \frac{1}{1 + \frac{p_{(0,0;j),(0,0;[\ell+1,n-2])} + p_{(0,0;j),(1,0;0)} + p_{(0,0;j),(0,1;[0,1])}}{p_{(0,0;j),(0,0;(j,\ell])}}}. \end{aligned} \tag{40}$$

For $j \le \ell - 1 < n/2$, the transition probabilities

$$p_{(0,0;j),(0,0;[\ell+1,n-2])} \le \binom{n-2-j}{\ell+1-j} \left(\frac{1}{n}\right)^{\ell+1-j},$$
$$p_{(0,0;j),(0,1;[0,1])} \le \frac{1}{n}.$$

$$p_{(0,0;j),(1,0;0)} \le \frac{1}{n},$$
$$p_{(0,0;j),(0,0;(j,\ell])} \ge \binom{n-2-j}{1} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1}.$$

Substituting them into (40), we get for $\ell < n/2$,

$$\begin{aligned} c_{(0,0;0),(0,0;\ell)} &\ge \prod_{j=0}^{\ell-1} \frac{1}{1 + \frac{e}{(\ell+1-j)!} + \frac{2e}{n-2-j}} \\ &\ge \prod_{j=0}^{n/2-1} \frac{1}{1 + \frac{e}{(n/2-j)!} + \frac{2e}{n-2-j}} = \Omega(1). \end{aligned}$$

We omit the proof of $\Omega(1)$ in the product, which one can refer to [27, Lemma 2] for details. Then we get

$$c_{(0,0;0),(1,0;0)} = \Omega\left(\frac{1}{n}\right) + \Omega\left(\frac{1}{n}\right) \sum_{\ell=1}^{n/2-1} \Omega(1) = \Omega(1).$$

Thus, the mean hitting time to the global optimal set $L_{(0,0;n-2)}$ is

$$m_{(0,0;0),(0,0;n-2)} \ge \frac{\Omega(1)}{(n-1)n^{-n}}. \tag{41}$$

By comparing (34) and (41), we observe that for KP2, the (1+1) EA using greedy repair is slower than that of the (1+1) EA using feasibility rules by a factor $O(n^{-1})$.

E. Instance KP3

Fig. 4 shows the digraph of the two (1+1) EAs on Instance KP3.

1) The (1+1) EA Using Feasibility Rules: According to the lower bound (8) in Theorem 2, the mean hitting time from the empty knapsack to the global optimum $L_{(1,0;0)}$ is

$$m_{(0,0;0),(1,0;0)} \ge \frac{h_{(0,0;0),(0,0;n-2)}}{p_{(0,0;n-2),(0,0;n-2)^+}}. \tag{42}$$

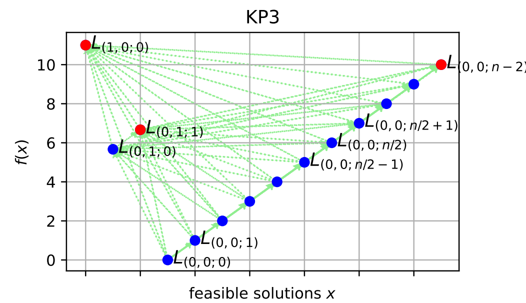


Fig. 4. The digraph of the two (1+1) EAs on Instance KP3. Vertices represent fitness levels (feasible solution area). Arcs represent transitions. $n = 12$.

Since $L^+_{(0,0;n-2)} = L_{(1,0;0)}$, the transition probability

$$p_{(0,0;n-2),(1,0;0)} = \left(1 - \frac{1}{n}\right)\left(\frac{1}{n}\right)^{n-1}.$$

Thus, the mean hitting time

$$m_{(0,0;0),(1,0;0)} \ge \Omega(n^{n-1}) h_{(0,0;0),(0,0;n-2)}. \tag{43}$$

Intuitively, the chain follows the path $L_{(0,0;0)} \to L_{(0,0;1)} \to \cdots \to L_{(0,0;n-2)}$ and reaches $L_{(0,0;n-2)}$ with positive probability. We use Corollary 2 to prove that the hitting probability $h_{(0,0;0),(0,0;n-2)} = \Omega(1)$. By (20), we get

$$\begin{aligned} & h_{(0,0;0),(0,0;n-2)} \\ \ge & \prod_{j=0}^{n-3} r_{(0,0;j),(0,0;(j,n-2])} = \prod_{j=0}^{n-3} \frac{p_{(0,0;j),(0,0;(j,n-2])}}{p_{(0,0;j),(0,0;j)^+}} \\ \ge & \prod_{j=0}^{n-3} \frac{p_{(0,0;j),(0,0;(j,n-2])}}{p_{(0,0;j),(0,0;(j,n-2])} + p_{(0,0;j),(0,1;[0,1])} + p_{(0,0,j),(1,0;0)}} \\ = & \prod_{j=0}^{n-3} \frac{1}{1 + \frac{p_{(0,0;j),(0,1;0)} + p_{(0,0;j),(0,1;1)} + p_{(0,0,j),(1,0;0)}}{p_{(0,0;j),(0,0;(j,n-2])}}}. \end{aligned}$$

The transition probabilities

$$p_{(0,0;j),(0,1;0)} \le \left(\frac{1}{n}\right)^{j+1}.$$

$$p_{(0,0;j),(0,1;1)} \le \begin{cases} \frac{1}{n}, & j \le 1, \\ \frac{1}{n}\binom{j}{j-1}\left(\frac{1}{n}\right)^j, & j \ge 2. \end{cases}$$

$$p_{(0,0;j),(1,0;0)} \le \left(\frac{1}{n}\right)^{1+j},$$

$$p_{(0,0;j),(0,0;[j+1,n-2])} \ge \binom{n-2-j}{1} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1}.$$

Then we get

$$\begin{aligned} & h_{(0,0;0),(0,0;j)} \\ \ge & \frac{1}{(1 + \frac{3e}{n-2-j})^2} \prod_{j=2}^{n-3} \frac{1}{1 + \frac{2e}{n^j(n-2-j)} + \frac{ej}{n^j(n-2-j)}} = \Omega(1). \end{aligned}$$

We omit the proof of the bound $\Omega(1)$ in the product, which one can refer to [27, Lemma 2] for details. By substituting the above bound $\Omega(1)$ into (43), we get the mean hitting time to the optimal solution as follows:

$$m_{(0,0;0),(1,0;0)} = \Omega(n^{n-1}). \tag{44}$$

2) The (1+1) EA with Greedy Repair: Let $S_0 = L_{(1,0;0)}$ represent the global optimal set, with S_1 containing all other feasible solutions. Following the same analysis applied to the (1+1) EA with greedy repair on Instance KP1, we get that the mean hitting time from an empty knapsack to the global optimal solution is given by

$$m_{(0,0;0),(1,0;0)} = O(n). \tag{45}$$

By comparing (34) and (41), we observe that for KP3, the (1+1) EA using greedy repair is faster than that of the (1+1) EA using feasibility rules by a factor $\Omega(n^{n-2})$. Table III demonstrates that neither greedy repair nor feasibility rules can dominate the other.

TABLE III
COMPARISON OF ALGORITHM 1 AND ALGORITHM 2.

	KP1	KP2	KP3
$\frac{\text{mean hitting time of Algorithm 1}}{\text{mean hitting time of Algorihm 2}}$	$\Omega(n)$	$O(n^{-1})$	$\Omega(n^{n-2})$

VII. Conclusions

This paper investigates the computation of coefficients in the linear drift function for elitist EAs. First, we provide a new interpretation of the linear bound coefficients, where each coefficient corresponds to a hitting probability at a specific fitness level. This transforms the task of estimating the hitting time into one of estimating the hitting probability. Second, we propose a new drift analysis method for estimating hit probability. This method improves the drift analysis method with new explicit expressions for estimating the hitting time.

The proposed method can estimate both lower and upper bounds on the hitting time, which is useful for comparing the hitting time of two EAs. To demonstrate this, it is applied to compare two EAs with feasibility rules and greedy repair for solving the knapsack problem. The results show that neither constraint handling technique consistently outperforms the other across various instances. However, in certain special cases, using greedy repair can significantly reduce the hitting time from exponential to polynomial.

Future research will aim to extend this framework to the analysis of more combinatorial optimization problems, such as vertex cover and maximum satisfiability problems. However, there are inherent limitations to the linear drift function. Specifically, it may not provide tight time bounds for fitness functions that do not follow a level-based structure. Also, it is not available for non-elitist EAs.

Acknowledgments

S. Y. Chong received research funding under Shenzhen Pengcheng Peacock Special Post. X. Yao was supported by the National Natural Science Foundation of China (Grant No. 62250710682), an internal grant from Lingnan University, the Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386).

References

[1] B. Doerr and F. Neumann, "A survey on recent progress in the theory of evolutionary algorithms for discrete optimization," *ACM Transactions on Evolutionary Learning and Optimization*, vol. 1, no. 4, pp. 1–43, 2021.

[2] B. Doerr, D. Johannsen, and C. Winzen, "Multiplicative drift analysis," *Algorithmica*, vol. 4, no. 64, pp. 673–697, 2012.

[3] T. Kötzing and M. S. Krejca, "First-hitting times under drift," *Theoretical Computer Science*, vol. 796, pp. 51–69, 2019.

[4] J. Lengler, "Drift analysis," in *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*. Springer, 2020, pp. 89–131.

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

[6] 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.

[7] B. Doerr and L. A. Goldberg, "Adaptive drift analysis," *Algorithmica*, vol. 65, no. 1, pp. 224–250, 2013.

[8] B. Mitavskiy, J. Rowe, and C. Cannings, "Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, no. 2, pp. 243–284, 2009.

[9] D. Johannsen, "Random combinatorial structures and randomized search heuristics," Ph.D. dissertation, Universität des Saarlandes Saarbrücken, 2010.

[10] J. He and Y. Zhou, "Drift Analysis with Fitness Levels for Elitist Evolutionary Algorithms," *Evolutionary Computation*, pp. 1–25, 03 2024.

[11] I. Wegener, "Methods for the analysis of evolutionary algorithms on pseudo-boolean functions," in *Evolutionary optimization*. Springer, 2003, pp. 349–369.

[12] D. Sudholt, "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, 2012.

[13] B. Doerr and T. Kötzing, "Lower bounds from fitness levels made easy," *Algorithmica*, vol. 86, no. 2, pp. 367–395, 2024.

[14] Z. Yang, H. Qiu, L. Gao, D. Xu, and Y. Liu, "A general framework of surrogate-assisted evolutionary algorithms for solving computationally expensive constrained optimization problems," *Information Sciences*, vol. 619, pp. 491–508, 2023.

[15] D. Molina-Pérez, E. Mezura-Montes, E. A. Portilla-Flores, E. Vega-Alvarado, and B. Calva-Yañez, "A differential evolution algorithm for solving mixed-integer nonlinear programming problems," *Swarm and Evolutionary Computation*, vol. 84, p. 101427, 2024.

[16] R. Wang and Z. Zhang, "Set theory-based operator design in evolutionary algorithms for solving knapsack problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1133–1147, 2021.

[17] L. Wang, Y. He, X. Wang, Z. Zhou, H. Ouyang, and S. Mirjalili, "A novel discrete differential evolution algorithm combining transfer function with modulo operation for solving the multiple knapsack problem," *Information Sciences*, vol. 680, p. 121170, 2024.

[18] J. He and X. Yao, "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.

[19] T. Chen, J. He, G. Chen, and X. Yao, "Choosing selection pressure for wide-gap problems," *Theoretical Computer Science*, vol. 411, no. 6, pp. 926–934, 2010.

[20] J. Jägersküpper, "Algorithmic analysis of a basic evolutionary algorithm for continuous optimization," *Theoretical Computer Science*, vol. 379, no. 3, pp. 329–347, 2007.

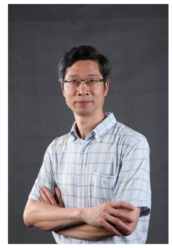
[21] S. Y. Yuen and B. K.-S. Cheung, "Bounds for probability of success of classical genetic algorithm based on Hamming distance," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 1–18, 2006.

[22] T. Kötzing, "Concentration of first hitting times under additive drift," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 1391–1398.
[23] J. He and X. Yao, "Towards an analytic framework for analysing the computation time of evolutionary algorithms," *Artificial Intelligence*, vol. 145, no. 1-2, pp. 59–97, 2003.
[24] J. He and G. Lin, "Average convergence rate of evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 316–321, 2016.
[25] S. Y. Chong, P. Tiňo, J. He, and X. Yao, "A new framework for analysis of coevolutionary systems—directed graph representation and random walks," *Evolutionary Computation*, vol. 27, no. 2, pp. 195–228, 2019.
[26] S. Y. Chong, P. Tiňo, and J. He, "Coevolutionary systems and pagerank," *Artificial Intelligence*, vol. 277, p. 103164, 2019.
[27] J. He, S. Y. Chong, and X. Yao, "Fast estimations of hitting time of elitist evolutionary algorithms from fitness levels," *arXiv*, 2023, doi.org/10.48550/arXiv.2311.10502.
[28] J. Norris, *Markov Chains*. Cambridge: Cambridge University Press, 1997.
[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.
[30] J. He and X. Yao, "Average drift analysis and population scalability," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 426–439, 2017.
[31] M. Mahrach, G. Miranda, C. León, and E. Segredo, "Comparison between single and multi-objective evolutionary algorithms to solve the knapsack problem and the travelling salesman problem," *Mathematics*, vol. 8, no. 11, p. 2018, 2020.
[32] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1996.
[33] Y. Zhou and J. He, "A runtime analysis of evolutionary algorithms for constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 608–619, 2007.

Jun He received his Bachelor of Science degree in Mathematics, his Master of Science degree in Mathematics, and his PhD in Computer Science from Wuhan University, China, in 1989, 1992, and 1995, respectively. He is currently an Associate Professor of Computer Science at the School of Science and Technology, Nottingham Trent University, UK.

His research interests include computational intelligence, evolutionary computation, global optimization, explainable artificial intelligence and its applications in healthcare..

Siang Yew Chong is a Research Associate Professor at the Department of Computer Science and Engineering, Southern University of Science and Technology, China, and a Shenzhen Pengcheng Peacock scholar. He is also a member of the Centre of Artificial Intelligence and Optimization, School of Computer Science, University of Nottingham Ningbo China. His main research interests are in Coevolutionary and Evolutionary Computation that employ methods from Machine Learning, Evolutionary Game Theory, and Dynamical Systems. Recently, he coauthored the monograph "Coevolutionary Computation and Its Applications" with Xin Yao published under Springer Nature (2025).



Xin Yao obtained his Ph.D. in 1990 from the University of Science and Technology of China (USTC), Hefei, China; MSc in 1985 from North China Institute of Computing Technologies, Beijing, China; and BSc in 1982 from USTC. He is the Vice President (Research and Innovation) and Tong Tin Sun Chair Professor of Machine Learning at Lingnan University, Hong Kong, China. He is a Fellow of IEEE and Hong Kong Academy of Engineering. He served as the President (2014-15) of IEEE Computational Intelligence Society (CIS) and the Editor-in-Chief (2003-08) of IEEE Transactions on Evolutionary Computation. His major research interests include evolutionary computation, machine learning and AI ethics. His work won the 2001 IEEE Donald G. Fink Prize Paper Award; 2010, 2016 and 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Awards; 2011 IEEE Transactions on Neural Networks Outstanding Paper Award; 2010 BT Gordon Radley Award for Best Author of Innovation (Finalist); and other best paper awards at conferences. He received the 2012 Royal Society Wolfson Research Merit Award, 2013 IEEE CIS Evolutionary Computation Pioneer Award, and 2020 IEEE Frank Rosenblatt Award.