

Article

# Influence of Binomial Crossover on Approximation Error of Evolutionary Algorithms

Cong Wang<sup>1</sup>, Jun He<sup>2</sup>, Yu Chen<sup>1,\*</sup>  and Xiufen Zou<sup>3,4</sup><sup>1</sup> School of Science, Wuhan University of Technology, Wuhan 430070, China<sup>2</sup> Department of Computer Science, Nottingham Trent University, Clifton Campus, Nottingham NG11 8NS, UK<sup>3</sup> School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China<sup>4</sup> Computational Science Hubei Key Laboratory, Wuhan University, Wuhan 430072, China

\* Correspondence: ychen@whut.edu.cn

**Abstract:** Although differential evolution (DE) algorithms perform well on a large variety of complicated optimization problems, only a few theoretical studies are focused on the working principle of DE algorithms. To make the first attempt to reveal the function of binomial crossover, this paper aims to answer whether it can reduce the approximation error of evolutionary algorithms. By investigating the expected approximation error and the probability of not finding the optimum, we conduct a case study comparing two evolutionary algorithms with and without binomial crossover on two classical benchmark problems: OneMax and Deceptive. It is proven that using binomial crossover leads to the dominance of transition matrices. As a result, the algorithm with binomial crossover asymptotically outperforms that without crossover on both OneMax and Deceptive, and outperforms on OneMax, however, not on Deceptive. Furthermore, an adaptive parameter strategy is proposed which can strengthen the superiority of binomial crossover on Deceptive.

**Keywords:** binomial crossover; differential evolution; fixed-budget analysis; evolutionary computation; approximation error

**MSC:** 90C15

**Citation:** Wang, C.; He, J.; Chen, Y.; Zou, X. Influence of Binomial Crossover on Approximation Error of Evolutionary Algorithms. *Mathematics* **2022**, *10*, 2850. <https://doi.org/10.3390/math10162850>

Academic Editors: Alexandru Agapie, Denis Enachescu, Vlad Stefan Barbu and Bogdan Iftimie

Received: 24 July 2022

Accepted: 9 August 2022

Published: 10 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Evolutionary algorithms (EAs) are a family of randomized search heuristics inspired from biological evolution, and many empirical studies demonstrate that crossovers that combine genes of two parents to generate new offspring could be helpful to the convergence of EAs [1–3]. Meanwhile, theoretical results on runtime analysis validate the promising function of crossover in EAs [4–15], whereas there are also some cases that crossover cannot be helpful [16,17].

By exchanging components of target vectors with donor vectors, differential evolution (DE) algorithms implement crossover operations in a different way. Numerical results show that continuous DE algorithms can achieve competitive performance on a large variety of complicated problems [18–21], and its competitiveness is to great extent attributed to the employed crossover operations [22]. However, the binary differential evolution (BDE) algorithm [23], which simulates the working mechanism of continuous DE, is not as competitive as its continuous counterpart. Analysis of the working principle indicates that the mutation and update strategies result in poor convergence of BDE [24], but there were no theoretical results reported on how crossover influences the performance of discrete-coded DE algorithms.

This paper is dedicated to investigating the influence of binomial crossover by introducing it to the  $(1 + 1)EA$ , excluding the impacts of population and mutation strategies of DE. Although the expected hitting time/runtime is popularly investigated in the theoretical study of randomized search heuristics (RSHs), there is a gap between runtime analysis

and practice because their optimization time to reach an optimum is uncertain and could be even infinite in continuous optimization [25]. Due to this reason, optimization time is seldom used in computer simulation for evaluating the performance of EAs, and their performance is evaluated after running finite generations by solution quality such as the mean and median of the fitness value or approximation error [26]. In theory, solution quality can be measured for a given iteration budget by the expected fitness value [27] or approximation error [28,29], which contributes to the analysis framework named fixed-budget analysis (FBA). An FBA on immune-inspired hypermutations led to theoretical results that are very different from those of runtime analysis but consistent with the empirical results, which demonstrates that the perspective of fixed-budget computations provides valuable information and additional insights for the performance of randomized search heuristics [30].

Accordingly, we evaluate the solution quality of an EA after running finite generations by the expected approximation error and the error tail probability. The former measures the fitness gap between a solution and optimum, and the latter is the probability distribution of the error over error levels, which measures the probability of finding the optimum. An EA is said to outperform another if, for the former EA, its error and tail probability are smaller. Furthermore, an EA is said to asymptotically outperform another if, for the former EA, its error and tail probability are smaller after a sufficiently large number of generations.

The research question of this paper is whether the binomial crossover operator can help reduce the approximation error of EA. As a pioneering work on this topic, we investigate a  $(1+1)EA_C$  that performs the binomial crossover on an individual and an offspring generated by mutation, and compare a  $(1+1)EA$  without crossover and its variant  $(1+1)EA_C$  on two classical problems, OneMax and Deceptive. By splitting the objective space into error levels, the analysis is performed based on the Markov chain models [31,32]. Given the two EAs, the comparison of their performance is drawn from the comparison of their transition probabilities, which are estimated by investigating the bits preferred by evolutionary operations. Under some conditions,  $(1+1)EA_C$  with binomial crossover outperforms  $(1+1)EA$  on OneMax, but not on Deceptive; however, by adding an adaptive parameter mechanism arising from theoretical results,  $(1+1)EA_C$  with binomial crossover outperforms  $(1+1)EA$  on Deceptive too.

This work presents the first study on how binomial crossover influences the expected runtime and tail probability of randomized search heuristics. Meanwhile, we also propose a feasible routine to get adaptive parameter settings of EAs from theoretical results. The rest of this paper is organized as follows. Section 2 reviews related theoretical work. Preliminary contents for our theoretical analysis are presented in Section 3. Then, the influence of the binomial crossover on transition probabilities is investigated in Section 4. Section 5 conducts an analysis of the asymptotic performance of EAs. To reveal how binomial crossover works on the performance of EAs for consecutive iterations, the OneMax problem and the Deceptive problem are investigated in Sections 6 and 7, respectively. Finally, Section 8 presents the conclusions and discussions.

## 2. Related Work

### 2.1. Theoretical Analysis of Crossover in Evolutionary Algorithms

To understand how crossover influences the performance of EAs, Jansen et al. [4] proved that an EA using crossover can reduce the expected optimization time from super-polynomial to a polynomial of small degree on the function Jump. Kötzing et al. [5] investigated crossover-based EAs on the functions OneMax and Jump and showed the potential speedup by crossover when combined with a fitness-invariant bit shuffling operator in terms of optimization time. For a simple GA without shuffling, they found that the crossover probability has a drastic impact on the performance on Jump. Corus and Oliveto [6] obtained an upper bound on the runtime of standard steady-state GAs to hillclimb the OneMax function and proved that the steady-state EAs are 25% faster than their mutation-only counterparts. Their analysis also suggests that larger populations may

be faster than populations of size 2. Dang et al. [7] revealed that the interplay between crossover and mutation may result in a sudden burst of diversity on the Jump test function and reduce the expected optimization time compared to mutation-only algorithms such as  $(1 + 1)$  EA. For royal road functions and OneMax, Sudholt [8] analyzed uniform crossover and  $k$ -point crossover and proved that crossover makes every  $(\mu + \lambda)$  EA at least twice as fast as the fastest EA using only standard bit mutation. Pinto and Doerr [9] provided a simple proof of a crossover-based genetic algorithm (GA) outperforming any mutation-based black-box heuristic on the classic benchmark OneMax. Oliveto et al. [10] obtained a tight lower bound on the expected runtime of the  $(2 + 1)$  GA on OneMax. Lengler and Meier [11] studied the positive effect of using larger population sizes and crossover on Dynamic BinVal.

For non-artificial problems, Lehre and Yao [12] proved that the use of crossover in the  $(\mu + 1)$  steady-state genetic algorithm may reduce the runtime from exponential to polynomial for some instance classes of the problem of computing unique input–output (UIO) sequences. Doerr et al. [13,14] analyzed EAs on the all-pairs shortest path problem. Their results confirmed that the EA with a crossover operator is significantly faster in terms of the expected optimization time. Sutton [15] investigated the closest string problem and proved that a multi-start  $(\mu + 1)$  GA required less randomized fixed-parameter tractable (FPT) time than that with disabled crossover.

However, there is some evidence that crossover is not always helpful. Richter et al. [16] constructed Ignoble Trail functions and proved that mutation-based EAs optimize them more efficiently than GAs with crossover. The later need exponential optimization time. Antipov and Naumov [17] compared crossover-based algorithms on RealJump functions with a slightly shifted optimum, which increases the runtime of all considered algorithms on RealJump. The hybrid GA fails to find the shifted optimum with high probability.

## 2.2. Theoretical Analysis of Differential Evolution Algorithms

Most existing theoretical studies on DE are focused on continuous variants [33]. By estimating the probability density function of generated individuals, Zhou et al. [34] demonstrated that the selection mechanism of DE, which chooses mutually different parents for the generation of donor vectors, sometimes does not work positively on the performance of DE. Zaharie and Micota [35–37] investigated the influence of the crossover rate on both the distribution of the number of mutated components and the probability for a component to be taken from the mutant vector, as well as the influence of mutation and crossover on the diversity of the intermediate population. Wang and Huang [38] attributed the DE to a one-dimensional stochastic model, and investigated how the probability distribution of population is connected to the mutation, selection, and crossover operations of DE. Opara and Arabas [39] compared several variants of the differential mutation using characteristics of their expected mutants' distribution, which demonstrated that the classic mutation operators yield similar search directions and differ primarily by the mutation range. Furthermore, they formalized the contour fitting notion and derived an analytical model that links the differential mutation operator with the adaptation of the range and direction of search [40].

By investigating the expected runtime of BDE, Doerr and Zhang [24] performed a first fundamental analysis on the working principles of discrete-coded DE. It was shown that BDE optimizes the important decision variables, but is hard to find the optima for decision variables with a small influence on the objective function. Since BDE generates trial vectors by implementing a binary variant of binomial crossover accompanied by the mutation operation, it has characteristics significantly different from classic EAs or estimation-of-distribution algorithms.

## 2.3. Fixed-Budget Analysis and Approximation Error

To bridge the wide gap between theory and application, Jasen and Zarges [27] proposed an FBA framework of RSHs, by which the fitness of random local search and  $(1 + 1)$

EA were investigated for given iteration budgets. Under the framework of FBA, Jasen and Zarges [41] analyzed the any-time performance of EAs and artificial immune systems on a proposed dynamic benchmark problem. Nallaperuma et al. [42] considered the well-known traveling salesperson problem (TSP) and derived the lower bounds of the expected fitness gain for a specified number of generations. Based on the Markov chain model of RSHs, Wang et al. [29] constructed a general framework of FBA, by which they found the analytic expression of approximation error instead of asymptotic results of expected fitness values. Doerr et al. [43] built a bridge between runtime analysis and FBA, by which a huge body of work and a large collection of tools for the analysis of the expected optimization time could meet the new challenges introduced by the new fixed-budget perspective.

Noting that hypermutations tend to be inferior to typical example functions in terms of runtime, Jansen and Zarges [30] conducted an FBA to explain why artificial immune systems are popular in spite of these proven drawbacks. It was shown that the inversely fitness-proportional mutation (IFPM) and the somatic contiguous hypermutation (CHM) could perform better than the single point mutation on OneMax while FBA is performed by considering different starting points and varied iteration budgets. It indicates that the traditional perspective of expected optimization time may be unable to explain the observed good performance, which is due to the limited length of runs. Therefore, the perspective of fixed-budget computations provides valuable information and additional insights.

### 3. Preliminaries

#### 3.1. Problems

Considering a maximization problem

$$\max f(\mathbf{x}), \quad \mathbf{x} = (x_1, \dots, x_n) \in \{0, 1\}^n,$$

denote its optimal solution by  $\mathbf{x}^*$  and optimal objective value by  $f^*$ . The quality of a solution  $\mathbf{x}$  is evaluated by its approximation error  $e(\mathbf{x}) := |f(\mathbf{x}) - f^*|$ . The error  $e(\mathbf{x})$  takes finite values, called error levels:

$$e(\mathbf{x}) \in \{e_0, e_1, \dots, e_L\}, \quad 0 = e_0 \leq e_1 \leq \dots \leq e_L,$$

where  $L$  is a non-negative integer.  $\mathbf{x}$  is called *at the level  $i$*  if  $e(\mathbf{x}) = e_i, i \in \{0, 1, \dots, L\}$ . The collection of solutions at level  $i$  is denoted by  $\mathcal{X}_i$ .

We investigate the optimization problem in the form

$$\max f(|\mathbf{x}|), \tag{1}$$

where  $|\mathbf{x}| := \sum_{i=1}^n x_i$ . Error levels of (1) take only  $n + 1$  values. Two instances, the unimodal OneMax problem and the multi-modal Deceptive problem, are considered in this paper.

#### Problem 1 (OneMax).

$$\max f(\mathbf{x}) = \sum_{i=1}^n x_i, \quad \mathbf{x} = (x_1, \dots, x_n) \in \{0, 1\}^n.$$

#### Problem 2 (Deceptive).

$$\max f(\mathbf{x}) = \begin{cases} \sum_{i=1}^n x_i, & \text{if } \sum_{i=1}^n x_i > n - 1, \\ n - 1 - \sum_{i=1}^n x_i, & \text{otherwise,} \end{cases} \quad \mathbf{x} = (x_1, \dots, x_n) \in \{0, 1\}^n.$$

For the OneMax problem, both exploration and exploitation are helpful to the convergence of EAs to the optimum, because exploration accelerates the convergence process and exploitation refines the precision of approximation solutions. However, for the Deceptive problem, local exploitation leads to convergence to the local optimum, but it in turn increases the difficulty to jump to the global optimum. That is, exploitation hinders convergence to the global optimum of the Deceptive problem, thus, the performance of EAs is dominantly influenced by their exploration ability.

### 3.2. Evolutionary Algorithms

For the sake of analysis on binomial crossover excluding the influence of population and mutation, the  $(1 + 1)EA$  presented in Algorithm 1 is taken as the baseline algorithm in our study. Its candidate solutions are generated by the bitwise mutation with probability  $p_m$ . The binomial crossover is appended to  $(1 + 1)EA$ , getting  $(1 + 1)EA_C$  which is illustrated in Algorithm 2. The  $(1 + 1)EA_C$  first performs bitwise mutation with probability  $q_m$ , and then applies binomial crossover with rate  $C_R$  to generate a candidate solution for selection.

The EAs investigated in this paper can be modeled as homogeneous Markov chains [31,32]. Given the error vector

$$\tilde{\mathbf{e}} = (e_0, e_1, \dots, e_L)', \tag{2}$$

and the initial distribution

$$\tilde{\mathbf{q}}^{[0]} = (q_0^{[0]}, q_1^{[0]}, \dots, q_L^{[0]})' \tag{3}$$

the transition matrix of  $(1 + 1)EA$  and  $(1 + 1)EA_C$  for the optimization problem (1) can be written in the form

$$\tilde{\mathbf{R}} = (r_{i,j})_{(L+1) \times (L+1)}, \tag{4}$$

where

$$r_{i,j} = \Pr\{\mathbf{x}_{t+1} \in \mathcal{X}_i \mid \mathbf{x}_t \in \mathcal{X}_j\}, \quad i, j = 0, \dots, L.$$

---

#### Algorithm 1 $(1 + 1)EA$

---

- 1: counter  $t = 0$ ;
- 2: randomly generate a solution  $\mathbf{x}_0 = (x_1, \dots, x_n)$ ;
- 3: **while** the stopping criterion is not satisfied **do**
- 4:   generate the mutant  $\mathbf{y}_t = (y_1, \dots, y_n)$  by bitwise mutation:

$$\text{for } i = 1, \dots, n, \quad y_i = \begin{cases} 1 - x_i, & \text{if } rnd_i < p_m, \\ x_i, & \text{otherwise,} \end{cases} \quad rnd_i \sim U[0, 1]; \tag{5}$$

- 5:   **if**  $f(\mathbf{y}) \geq f(\mathbf{x}_t)$  **then**
  - 6:      $\mathbf{x}_{t+1} = \mathbf{y}_t$ ;
  - 7:   **else**
  - 8:      $\mathbf{x}_{t+1} = \mathbf{x}_t$ ;
  - 9:   **end if**
  - 10:    $t = t + 1$ ;
  - 11: **end while**
-

---

**Algorithm 2**  $(1 + 1)EA_C$

---

- 1: counter  $t = 0$ ;
- 2: randomly generate a solution  $\mathbf{x}_0 = (x_1, \dots, x_n)$ ;
- 3: **while** the stopping criterion is not satisfied **do**
- 4:   Generate the mutant  $\mathbf{v} = (v_1, \dots, v_n)$  by bitwise mutation:

$$\text{for } i = 1, \dots, n, \quad v_i = \begin{cases} 1 - x_i, & \text{if } rnd1_i < q_m, \\ x_i, & \text{otherwise,} \end{cases} \quad rnd1_i \sim U[0, 1]; \quad (6)$$

- 5:   set  $rndi \sim U\{1, 2, \dots, n\}$ ;
- 6:   generate the offspring  $\mathbf{y} = (y_1, \dots, y_n)$  by performing binomial crossover on  $\mathbf{v}$ :

$$\text{for } i = 1, \dots, n, \quad y_i = \begin{cases} v_i, & \text{if } i = rndi \text{ or } rnd2_i < C_R, \\ x_i, & \text{otherwise,} \end{cases} \quad rnd2_i \sim U[0, 1]; \quad (7)$$

- 7:   **if**  $f(\mathbf{y}) \geq f(\mathbf{x}_t)$  **then**
  - 8:      $\mathbf{x}_{t+1} = \mathbf{y}$ ;
  - 9:   **else**
  - 10:     $\mathbf{x}_{t+1} = \mathbf{x}_t$ ;
  - 11:   **end if**
  - 12:    $t = t + 1$ ;
  - 13: **end while**
- 

Recalling that the solutions are updated by the elitist selection, we know  $\tilde{\mathbf{R}}$  is an upper triangular matrix that can be partitioned as

$$\tilde{\mathbf{R}} = \begin{pmatrix} 1 & \mathbf{r}_0 \\ \mathbf{0} & \mathbf{R} \end{pmatrix},$$

where  $\mathbf{r}_0$  represents the probabilities to transfer from non-optimal statuses to the optimal status, and  $\mathbf{R}$  is the transition submatrix depicting the transitions between non-optimal statuses.

### 3.3. Transition Probabilities

Transition probabilities can be confirmed by considering generation of a candidate  $\mathbf{y}$  with  $f(\mathbf{y}) \geq f(\mathbf{x})$ , which is achieved if “ $l$  preferred bits” of  $\mathbf{x}$  are changed. If there are multiple solutions that are better than  $\mathbf{x}$ , there could be multiple choices for both the number  $l$  and the location of “ $l$  preferred bits”.

**Example 1.** For the OneMax problem,  $e(\mathbf{x})$  equals to the amount of ‘0’-bits in  $\mathbf{x}$ . Denoting  $e(\mathbf{x}) = j$  and  $e(\mathbf{y}) = i$ , we know  $\mathbf{y}$  replaces  $\mathbf{x}$  if and only if  $j \geq i$ . Then, to generate a candidate  $\mathbf{y}$  replacing  $\mathbf{x}$ , “ $l$  preferred bits” can be confirmed as follows.

- If  $i = j$ , “ $l$  preferred bits” consist of  $l/2$  ‘1’-bits and  $l/2$  ‘0’-bits, where  $l$  is an even number that is not greater than  $\min\{2j, 2(n - j)\}$ .
- While  $i < j$ , “ $l$  preferred bits” could be combinations of  $j - i + k$  ‘0’-bits and  $k$  ‘1’-bits ( $l = j - i + 2k$ ), where  $0 \leq k \leq \min\{i, n - j\}$ . Here,  $k$  is not greater than  $i$ , because  $j - i + k$  could not be greater than  $j$ , the number of ‘0’-bits in  $\mathbf{x}$ . Meanwhile,  $k$  does not exceed  $n - j$ , the number of ‘1’-bits in  $\mathbf{x}$ .

If an EA flips each bit with an identical probability, the probability of flipping  $l$  bits are related to  $l$  and independent of their locations. Denoting the probability of flipping  $l$  bits by  $P(l)$ , we can confirm the connection between the transition probability  $r_{i,j}$  and  $P(l)$ .

As presented in Example 1, transition from level  $j$  to level  $i$  ( $i < j$ ) results from flips of  $j - i + k$  '0'-bits and  $k$  '1'-bits. Then, transition probabilities for OneMax are confirmed as

$$r_{i,j} = \sum_{k=0}^M C_{n-j}^k C_j^{k+(j-i)} P(2k + j - i), \tag{8}$$

where  $M = \min\{n - j, i\}, 0 \leq i < j \leq n$ .

According to definition of the Deceptive problem, we get the following map from  $|x\rangle$  to  $e(x)$ .

$$\begin{array}{l} |x\rangle: \quad 0 \quad 1 \quad \dots \quad n-1 \quad n \\ e(x): \quad 1 \quad 2 \quad \dots \quad n \quad 0 \end{array} \tag{9}$$

Transition from level  $j$  to level  $i$  ( $0 \leq i < j \leq n$ ) is attributed to one of the following cases.

- If  $i \geq 1$ , the amount of '1'-bits decreases from  $j - 1$  to  $i - 1$ . This transition results from a change of  $j - i + k$  '1'-bits and  $k$  '0'-bits, where  $0 \leq k \leq \min\{n - j + 1, i - 1\}$ ;
- if  $i = 0$ , all of  $n - j + 1$  '0'-bits are flipped, and all of its '1'-bits keep unchanged.

Accordingly, transition probabilities for Deceptive are confirmed as

$$r_{i,j} = \begin{cases} \sum_{k=0}^M C_{n-j+1}^k C_{j-1}^{k+(j-i)} P(2k + j - i), & i \geq 1, \\ P(n - j + 1), & i = 0, \end{cases} \tag{10}$$

where  $M = \min\{n - j + 1, i - 1\}$ .

### 3.4. Performance Metrics

To evaluate the performance of EAs, we propose two metrics for a given iteration budget, the expected approximation error (EAE) and the tail probability (TP) of EAs for  $t$  consecutive iterations.

**Definition 1.** Let  $\{x_t, t = 1, 2, \dots\}$  be the individual sequence of an individual-based EA.

(1) The expected approximation error (EAE) after  $t$  consecutive iterations is

$$e^{[t]} = \mathbb{E}[e(x_t)] = \sum_{i=0}^L e_i \Pr\{e(x_t) = e_i\}. \tag{11}$$

(2) Given  $i > 0$ , the tail probability (TP) of the approximation error that  $e(x_t)$  is greater than or equal to  $e_i$  is defined as

$$p^{[t]}(e_i) = \Pr\{e(x_t) \geq e_i\}. \tag{12}$$

EAE is the fitness gap between a solution and the optimum. It measures solution quality after running  $t$  generations. TP is the probability distribution of a found solution over non-optimal levels where  $i > 0$ . The sum of TP is the probability of not finding the optimum.

Given two EAs  $\mathcal{A}$  and  $\mathcal{B}$ , if both EAE and TP of Algorithm  $\mathcal{A}$  are smaller than those of Algorithm  $\mathcal{B}$  for any iteration budget, we say Algorithm  $\mathcal{A}$  outperforms Algorithm  $\mathcal{B}$  on problem (1).

**Definition 2.** Let  $\mathcal{A}$  and  $\mathcal{B}$  be two EAs applied to problem (1).

1. Algorithm  $\mathcal{A}$  outperforms  $\mathcal{B}$ , denoted by  $\mathcal{A} \succsim \mathcal{B}$ , if it holds that

- $e_{\mathcal{A}}^{[t]} - e_{\mathcal{B}}^{[t]} \leq 0, \forall t > 0$ ;
- $p_{\mathcal{A}}^{[t]}(e_i) - p_{\mathcal{B}}^{[t]}(e_i) \leq 0, \forall t > 0, 0 < i < L$ .

2. Algorithm  $\mathcal{A}$  asymptotically outperforms  $\mathcal{B}$  on problem (1), denoted by  $\mathcal{A} \succ^a \mathcal{B}$ , if it holds that
- $\lim_{t \rightarrow \infty} e_{\mathcal{A}}^{[t]} - e_{\mathcal{B}}^{[t]} \leq 0$ ;
  - $\lim_{t \rightarrow +\infty} p_{\mathcal{A}}^{[t]}(e_i) - p_{\mathcal{B}}^{[t]}(e_i) \leq 0$ .

The asymptotic outperformance is weaker than the outperformance.

#### 4. Comparison of Transition Probabilities of Two EAs

In this section, we compare transition probabilities of  $(1 + 1)EA$  and  $(1 + 1)EA_C$ . According to the connection between  $r_{i,j}$  and  $P(l)$ , a comparison of transition probabilities can be conducted by considering the probabilities of flipping “ $l$  preferred bits”.

##### 4.1. Probabilities of Flipping Preferred Bits

Denote probabilities of  $(1 + 1)EA$  and  $(1 + 1)EA_C$  to flip “ $l$  preferred bits” by  $P_1(l, p_m)$  and  $P_2(l, C_R, q_m)$ , respectively. By (5), we know

$$P_1(l, p_m) = (p_m)^l(1 - p_m)^{n-l}. \tag{13}$$

Since the mutation and the binomial crossover in Algorithm 2 are mutually independent, we can get the probability by considering the crossover first. When flipping “ $l$  preferred bits” by  $(1 + 1)EA_C$ , there are  $l + k$  ( $0 \leq k \leq n - l$ ) bits of  $\mathbf{y}$  set as  $v_i$  by (7), the probability of which is

$$P_C(l + k, C_R) = \frac{l + k}{n} (C_R)^{l+k-1} (1 - C_R)^{n-l-k}.$$

If only “ $l$  preferred bits” are flipped, we know,

$$\begin{aligned} P_2(l, C_R, q_m) &= \sum_{k=0}^{n-l} C_{n-l}^k P_C(l + k, C_R) (q_m)^l (1 - q_m)^k \\ &= \frac{1}{n} [l + (n - l)C_R - nq_m C_R] (C_R)^{l-1} (q_m)^l (1 - q_m C_R)^{n-l-1}. \end{aligned} \tag{14}$$

Note that  $(1 + 1)EA_C$  degrades to  $(1 + 1)EA$  when  $C_R = 1$ , and  $(1 + 1)EA$  becomes the random search while  $p_m = 1$ . Thus, we assume that  $p_m, C_R$ , and  $q_m$  are located in  $(0, 1)$ . A fair comparison of transition probabilities is investigated by considering the identical parameter setting

$$p_m = C_R q_m = p, \quad 0 < p < 1. \tag{15}$$

Then, we know  $q_m = p/C_R$ , and Equation (14) implies

$$P_2(l, C_R, p/C_R) = \frac{1}{n} \left[ (n - l) + \frac{l - np}{C_R} \right] p^l (1 - p)^{n-l-1}. \tag{16}$$

Subtracting (13) from (16), we have

$$\begin{aligned} P_2(l, C_R, p/C_R) - P_1(l, p) &= \left\{ \frac{1}{n} \left[ (n - l) + \frac{l - np}{C_R} \right] - (1 - p) \right\} p^l (1 - p)^{n-l-1} \\ &= \left( \frac{1}{C_R} - 1 \right) \left( \frac{l}{n} - p \right) p^l (1 - p)^{n-l-1}. \end{aligned} \tag{17}$$

From the fact that  $0 < C_R < 1$ , we conclude that  $P_2(l, C_R, p/C_R)$  is greater than  $P_1(l, p)$  if and only if  $l > np$ . That is, the introduction of the binomial crossover in  $(1 + 1)EA$  leads to the enhancement of the exploration ability of  $(1 + 1)EA_C$ . We get the following theorem for the case that  $p \leq \frac{1}{n}$ .

**Theorem 1.** While  $0 < p \leq \frac{1}{n}$ , it holds for all  $1 \leq l \leq n$  that  $P_1(l, p) \leq P_2(l, C_R, p/C_R)$ .

**Proof.** The result can be obtained directly from Equation (17) by setting  $p \leq \frac{1}{n}$ .  $\square$

For the popular setting where the mutation probability of (1+1)EA is set as  $1/n$ , the introduction of binomial crossover does increase the ability to generate new candidate solutions. Then, we investigate how this improvement contributes to change of transition probabilities.

#### 4.2. Comparison of Transition Probabilities

To validate that algorithm  $\mathcal{A}$  is more efficient than algorithm  $\mathcal{B}$ , it is assumed that the probability of  $\mathcal{A}$  to transfer to promising statuses could be not smaller than that of  $\mathcal{B}$ .

**Definition 3.** Let  $\mathcal{A}$  and  $\mathcal{B}$  be two EAs with an identical initialization mechanism.  $\tilde{\mathbf{A}} = (a_{i,j})$  and  $\tilde{\mathbf{B}} = (b_{i,j})$  are the transition matrices of  $\mathcal{A}$  and  $\mathcal{B}$ , respectively. It is said that  $\tilde{\mathbf{A}}$  dominates  $\tilde{\mathbf{B}}$ , denoted by  $\tilde{\mathbf{A}} \succeq \tilde{\mathbf{B}}$ , if it holds that

1.  $a_{i,j} \geq b_{i,j}, \quad \forall 0 \leq i < j \leq L;$
2.  $a_{i,j} > b_{i,j}, \quad \exists 0 \leq i < j \leq L.$

Denote the transition probabilities of (1 + 1)EA and (1 + 1)EA<sub>C</sub> by  $p_{i,j}$  and  $s_{i,j}$ , respectively. For the OneMax problem and Deceptive problem, we get the relation of transition dominance on the premise that  $p_m = C_R q_m = p \leq \frac{1}{n}$ .

**Theorem 2.** For (1 + 1)EA and (1 + 1)EA<sub>C</sub>, denote their transition matrices by  $\tilde{\mathbf{P}}$  and  $\tilde{\mathbf{S}}$ , respectively. On the condition that  $p_m = C_R q_m = p \leq \frac{1}{n}$ , it holds for problem (1) that  $\tilde{\mathbf{S}} \succeq \tilde{\mathbf{P}}$ .

**Proof.** Denote the collection of all solutions at level  $k$  by  $\mathcal{S}(k), k = 0, 1, \dots, n$ . We prove the result by considering the transition probability

$$r_{i,j} = \Pr\{\mathbf{y} \in \mathcal{S}(i) \mid \mathbf{x} \in \mathcal{S}(j)\}, \quad (i < j).$$

Since the function values of solutions are merely related to the number of ‘1’-bits, the probability to generate a solution  $\mathbf{y} \in \mathcal{S}(i)$  by performing mutation on  $\mathbf{x} \in \mathcal{S}(j)$  depends on the Hamming distance  $l = H(\mathbf{x}, \mathbf{y})$ . Given  $\mathbf{x} \in \mathcal{S}_j, \mathcal{S}(i)$  is partitioned as  $\mathcal{S}(i) = \cup_{l=1}^L \mathcal{S}_l(i)$ , where  $\mathcal{S}_l(i) = \{\mathbf{y} \in \mathcal{S}(i) \mid H(\mathbf{x}, \mathbf{y}) = l\}$ , and  $L$  is a positive integer that is smaller than or equal to  $n$ .

Accordingly, the probability to transfer from level  $j$  to  $i$  is confirmed as

$$r_{i,j} = \sum_{l=1}^L \Pr\{\mathbf{y} \in \mathcal{S}_l(i) \mid \mathbf{x} \in \mathcal{S}(j)\} = \sum_{l=1}^L |\mathcal{S}_l(i)| P(l),$$

where  $|\mathcal{S}_l(i)|$  is the size of  $\mathcal{S}_l(i), P(l)$  the probability to flip “ $l$  preferred bits”. Then,

$$p_{i,j} = \sum_{l=1}^L \Pr\{\mathbf{y} \in \mathcal{S}_l(j) \mid \mathbf{x}\} = \sum_{l=1}^L |\mathcal{S}_l(j)| P_1(l, p), \tag{18}$$

$$s_{i,j} = \sum_{l=1}^L \Pr\{\mathbf{y} \in \mathcal{S}_l(j) \mid \mathbf{x}\} = \sum_{l=1}^L |\mathcal{S}_l(j)| P_2(l, C_R, p/C_R). \tag{19}$$

Since  $p \leq 1/n$ , Theorem 1 implies that

$$P_1(l, p) \leq P_2(l, C_R, p/C_R), \quad \forall 1 \leq l \leq n.$$

Combining it with (18) and (19) we know

$$p_{i,j} \leq s_{i,j}, \quad \forall 0 \leq i < j \leq n. \tag{20}$$

Then, we get the result by Definition 2.  $\square$

**Example 2 (Comparison of transition probabilities for the OneMax problem).** Let  $p_m = C_R q_m = p \leq \frac{1}{n}$ . By (8), we have

$$p_{i,j} = \sum_{k=0}^M C_{n-j}^k C_j^{k+(j-i)} P_1(2k + j - i, p), \tag{21}$$

$$s_{i,j} = \sum_{k=0}^M C_{n-j}^k C_j^{k+(j-i)} P_2(2k + j - i, C_R, p/C_R). \tag{22}$$

where  $M = \min\{n - j, i\}$ . Since  $p \leq 1/n$ , Theorem 1 implies that

$$P_1(2k + j - i, p) \leq P_2(2k + j - i, C_R, p/C_R),$$

and by (21) and (22) we have  $p_{i,j} \leq s_{i,j}, \forall 0 \leq i < j \leq n$ .

**Example 3 (Comparison of transition probabilities for the Deceptive problem).** Let  $p_m = C_R q_m = p \leq \frac{1}{n}$ . Equation (10) implies that

$$p_{i,j} = \begin{cases} \sum_{k=0}^M C_{n-j+1}^k C_{j-1}^{k+(j-i)} P_1(2k + j - i, p), & i > 0, \\ P_1(n - j + 1, p), & i = 0, \end{cases} \tag{23}$$

$$s_{i,j} = \begin{cases} \sum_{k=0}^M C_{n-j+1}^k C_{j-1}^{k+(j-i)} P_2(2k + j - i, C_R, \frac{p}{C_R}), & i > 0, \\ P_2(n - j + 1, C_R, p/C_R), & i = 0, \end{cases} \tag{24}$$

where  $M = \min\{n - j + 1, i - 1\}$ . Similar to the analysis of Example 2, we get the conclusion that  $p_{i,j} \leq s_{i,j}, \forall 0 \leq i < j \leq n$ .

The results demonstrate that when  $p \leq 1/n$ , the introduction of binomial crossover leads to transition dominance of  $(1 + 1)EA_C$  over  $(1 + 1)EA$ . In the following section, we would like to answer if transition dominance leads to outperformance of  $(1 + 1)EA_C$  over  $(1 + 1)EA$ .

### 5. Analysis of Asymptotic Performance

In this section, we will prove that  $(1 + 1)EA_C$  asymptotically outperforms  $(1 + 1)EA$  using the average convergence rate [25,32].

**Definition 4.** The average convergence rate (ACR) of an EA for  $t$  generation is

$$R_{EA}(t) = 1 - \left( e^{[t]} / e^{[0]} \right)^{1/t}. \tag{25}$$

**Lemma 1** ([32], Theorem 1). Let  $\mathbf{R}$  be the transition submatrix associated with a convergent EA. Under random initialization (i.e., the EA may start at any initial state with a positive probability), it holds

$$\lim_{t \rightarrow +\infty} R_{EA}(t) = 1 - \rho(\mathbf{R}), \tag{26}$$

where  $\rho(\mathbf{R})$  is the spectral radius of  $\mathbf{R}$ .

Lemma 1 presents the asymptotic characteristics of the ACR, by which we get the result on the asymptotic performance of EAs.

**Proposition 1.** If  $\tilde{\mathbf{A}} \succeq \tilde{\mathbf{B}}$ , there exists  $T > 0$  such that

1.  $e_{\mathbf{A}}^{[t]} \leq e_{\mathbf{B}}^{[t]}, \forall t > T;$

$$2. \quad p_{\mathcal{A}}^{[t]}(e_i) \leq p_{\mathcal{B}}^{[t]}(e_i), \forall t > T, 1 \leq i \leq L.$$

**Proof.** By Lemma 1, we know  $\forall \epsilon > 0$ , there exists  $T > 0$  such that

$$e^{[0]}(\rho(\mathbf{R}) - \epsilon)^t < e^{[t]} < e^{[0]}(\rho(\mathbf{R}) + \epsilon)^t, \quad t > T. \tag{27}$$

From the fact that the transition submatrix  $\mathbf{R}$  of an RSH is upper triangular, we conclude

$$\rho(\mathbf{R}) = \max\{r_{1,1}, \dots, r_{L,L}\}. \tag{28}$$

Denote

$$\tilde{\mathbf{A}} = (a_{i,j}) = \begin{pmatrix} 1 & \mathbf{a}_0 \\ \mathbf{0} & \mathbf{A} \end{pmatrix}, \quad \tilde{\mathbf{B}} = (b_{i,j}) = \begin{pmatrix} 1 & \mathbf{b}_0 \\ \mathbf{0} & \mathbf{B} \end{pmatrix}.$$

While  $\tilde{\mathbf{A}} \succeq \tilde{\mathbf{B}}$ , it holds

$$a_{j,j} = 1 - \sum_{i=0}^{j-1} a_{i,j} < 1 - \sum_{i=0}^{j-1} b_{i,j} = b_{j,j}, \quad 1 \leq j \leq L.$$

Then, Equation (28) implies that

$$\rho(\mathbf{A}) < \rho(\mathbf{B}).$$

Applying it to (27) for  $\epsilon < \frac{1}{2}(\rho(\mathbf{B}) - \rho(\mathbf{A}))$ , we have

$$e_{\mathcal{A}}^{[t]} < e^{[0]}(\rho(\mathbf{A}) + \epsilon)^t < e^{[0]}(\rho(\mathbf{B}) - \epsilon)^t < e_{\mathcal{B}}^{[t]}, \tag{29}$$

which proves the first conclusion.

Noting that the tail probability  $p^{[t]}(e_i)$  can be taken as the expected approximation error of an optimization problem with an error vector

$$\mathbf{e} = (\underbrace{0, \dots, 0}_i, 1, \dots, 1)',$$

by (29) we have

$$p_{\mathcal{A}}^{[t]}(e_i) \leq p_{\mathcal{B}}^{[t]}(e_i), \quad \forall t > T, 1 \leq i \leq L.$$

The second conclusion is proven.  $\square$

By Definition 2 and Proposition 1, we get the following theorem for comparing the asymptotic performance of  $(1 + 1)EA$  and  $(1 + 1)EA_C$ .

**Theorem 3.** *If  $C_R = C_R q_m = p \leq \frac{1}{n}$ , the  $(1 + 1)EA_C$  asymptotically outperforms  $(1 + 1)EA$  on problem (1).*

**Proof.** The proof can be completed by applying Theorem 2 and Proposition 1.  $\square$

On condition that  $C_R = C_R q_m = p \leq \frac{1}{n}$ , Theorem 3 indicates that after sufficiently many number of iterations,  $(1 + 1)EA_C$  can performs better on problem (1) than  $(1 + 1)EA$ . A further question is whether  $(1 + 1)EA_C$  outperforms  $(1 + 1)EA$  for  $t < +\infty$ . We answer the question in next sections.

### 6. Comparison of the Two EAs on OneMax

In this section, we show that the outperformance introduced by binomial crossover can be obtained for the uni-modal OneMax problem based on the following lemma [29].

**Lemma 2** ([29], Theorem 3). *Let*

$$\tilde{\mathbf{e}} = (e_0, e_1, \dots, e_L)', \quad \tilde{\mathbf{v}} = (v_0, v_1, \dots, v_L)',$$

where  $0 \leq e_{i-1} \leq e_i, i = 1, \dots, L, v_i > 0, i = 0, 1, \dots, L$ . If transition matrices  $\tilde{\mathbf{R}}$  and  $\tilde{\mathbf{S}}$  satisfy

$$s_{j,j} \geq r_{j,j}, \quad \forall 1 \leq j \leq L, \tag{30}$$

$$\sum_{l=0}^{i-1} (r_{l,j} - s_{l,j}) \geq 0, \quad \forall 0 \leq i < j \leq L, \tag{31}$$

$$\sum_{l=0}^i (s_{l,j-1} - s_{l,j}) \geq 0, \quad \forall 0 \leq i < j-1 < L, \tag{32}$$

it holds

$$\tilde{\mathbf{e}}' \tilde{\mathbf{R}}^t \tilde{\mathbf{v}} \leq \tilde{\mathbf{e}}' \tilde{\mathbf{S}}^t \tilde{\mathbf{v}}.$$

For the EAs investigated in this study, conditions (30)–(32) are satisfied thanks to the monotonicity of transition probabilities.

**Lemma 3.** *When  $p \leq 1/n$  ( $n \geq 3$ ),  $P_1(l, p)$  and  $P_2(l, C_R, p/C_R)$  are monotonously decreasing in  $l$ .*

**Proof.** When  $p \leq 1/n$ , Equations (13) and (14) imply that

$$\frac{P_1(l+1, p)}{P_1(l, p)} = \frac{p}{1-p} \leq \frac{1}{n-1}, \tag{33}$$

$$\frac{P_2(l+1, C_R, p/C_R)}{P_2(l, C_R, p/C_R)} = \frac{(l+1)(1-C_R)+nC_R(1-p/C_R)}{l(1-C_R)+nC_R(1-p/C_R)} \frac{p}{1-p} \leq \frac{l+1}{l} \frac{p}{1-p} \leq \frac{l+1}{l} \frac{1}{n-1}, \tag{34}$$

all of which are not greater than 1 when  $n \geq 3$ . Thus,  $P_1(l, p)$  and  $P_2(l, C_R, p/C_R)$  are monotonously decreasing in  $l$ . □

**Lemma 4.** *For the OneMax problem,  $p_{i,j}$  and  $s_{i,j}$  are decreasing in  $j$ .*

**Proof.** We validate the monotonicity of  $p_{i,j}$  for (1 + 1)EA, and that of  $s_{i,j}$  can be confirmed in a similar way.

Let  $0 \leq i < j < n$ . By (21) we know

$$p_{i,j+1} = \sum_{k=0}^M C_{n-j-1}^k C_{j+1}^{i-k} P_1(2k+j+1-i, p), \tag{35}$$

$$p_{i,j} = \sum_{k=0}^M C_{n-j}^k C_j^{i-k} P_1(2k+j-i, p), \tag{36}$$

where  $M = \min\{n-j-1, i\}$ . Moreover, (33) implies that

$$\frac{C_{j+1}^{i-k} P_1(2k+j+1-i, p)}{C_j^{i-k} P_1(2k+j-i, p)} = \frac{j+1}{(j+1)-(i-k)} \frac{p}{1-p} \leq \frac{j+1}{2} \frac{1}{n-1} < 1,$$

and we know

$$C_{j+1}^{i-k} P_1(2k+j+1-i, p) < C_j^{i-k} P_1(2k+j-i, p). \tag{37}$$

Note that

$$\min\{n-j-1, i\} \geq \min\{n-j, i\}, \quad C_{n-j-1}^k < C_{n-j}^k. \tag{38}$$

From (35)–(38) we conclude that

$$p_{i,j+1} < p_{i,j}, \quad 0 \leq i < j < n.$$

Similarly, we can validate that

$$s_{i,j+1} < s_{i,j}, \quad 0 \leq i < j < n.$$

In conclusion,  $p_{i,j}$  and  $s_{i,j}$  are monotonously decreasing in  $j$ .  $\square$

**Theorem 4.** On condition that  $p_m = C_R q_m = p \leq \frac{1}{n}$ , it holds for the OneMax problem that

$$(1 + 1)EA_C \succsim (1 + 1)EA.$$

**Proof.** Given the initial distribution  $\tilde{\mathbf{q}}^{[0]}$  and transition matrix  $\tilde{\mathbf{R}}$ , the level distribution at iteration  $t$  is confirmed by

$$\tilde{\mathbf{q}}^{[t]} = \tilde{\mathbf{R}}^t \tilde{\mathbf{q}}^{[0]}. \tag{39}$$

Denote

$$\tilde{\mathbf{e}} = (e_0, e_1, \dots, e_L)', \quad \tilde{\mathbf{o}}_i = (\underbrace{0, \dots, 0}_i, 1, \dots, 1)'$$

By premultiplying (39) with  $\tilde{\mathbf{e}}$  and  $\tilde{\mathbf{o}}_i$ , respectively, we get

$$e^{[t]} = \tilde{\mathbf{e}}' \tilde{\mathbf{R}}^t \tilde{\mathbf{q}}^{[0]}, \tag{40}$$

$$p^{[t]}(e_i) = \Pr\{e(\mathbf{x}_t)\} \geq e_i\} = \tilde{\mathbf{o}}_i' \tilde{\mathbf{R}}^t \tilde{\mathbf{q}}^{[0]}. \tag{41}$$

Meanwhile, by Theorem 2 we have

$$q_{j,j} \leq s_{j,j} \leq p_{j,j}, \tag{42}$$

$$\sum_{l=0}^{i-1} (q_{l,j} - s_{l,j}) \geq 0, \quad \sum_{l=0}^{i-1} (s_{l,j} - p_{l,j}) \geq 0, \quad \forall i < j, \tag{43}$$

and Lemma 4 implies

$$\sum_{l=0}^i (s_{l,j-1} - s_{l,j}) \geq 0, \quad \sum_{l=0}^i (p_{l,j-1} - p_{l,j}) \geq 0 \quad \forall i < j - 1. \tag{44}$$

Then, (42)–(44) validate satisfaction of conditions (30)–(32), and by Lemma 2 we know

$$\begin{aligned} \tilde{\mathbf{e}}' \tilde{\mathbf{S}}^t \tilde{\mathbf{q}}^{[0]} &\leq \tilde{\mathbf{e}}' \tilde{\mathbf{P}}^t \tilde{\mathbf{q}}^{[0]}, & \forall t > 0; \\ \tilde{\mathbf{o}}_i' \tilde{\mathbf{S}}^t \tilde{\mathbf{q}}^{[0]} &\leq \tilde{\mathbf{o}}_i' \tilde{\mathbf{P}}^t \tilde{\mathbf{q}}^{[0]}, & \forall t > 0, 1 \leq i < n. \end{aligned}$$

Then, we get the conclusion by Definition 2.  $\square$

The above theorem demonstrates that the dominance of transition matrices introduced by the binomial crossover operator leads to the outperformance of  $(1 + 1)EA_C$  on the uni-modal problem OneMax.

### 7. Comparison of the Two EAs on Deceptive

In this section, we show that the outperformance of  $(1 + 1)EA_C$  over  $(1 + 1)EA$  may not always hold on Deceptive. Then, we propose an adaptive strategy of parameter setting arising from the theoretical analysis, with which  $(1 + 1)EA_C$  performs better in terms of tail probability.

7.1. Numerical Demonstration for Inconsistency between the Transition Dominance and the Algorithm Outperformance

For the Deceptive problem, we first present a counterexample to show even if the transition matrix of an EA dominates another EA, we cannot conclude that the former EA outperforms the latter.

**Example 4.** We construct two artificial Markov chains as the models of two EAs. Let  $EA_{\mathcal{R}}$  and  $EA_{\mathcal{S}}$  be two EAs starting with an identical initial distribution

$$\mathbf{p}^{[0]} = \left( \frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \right)^t,$$

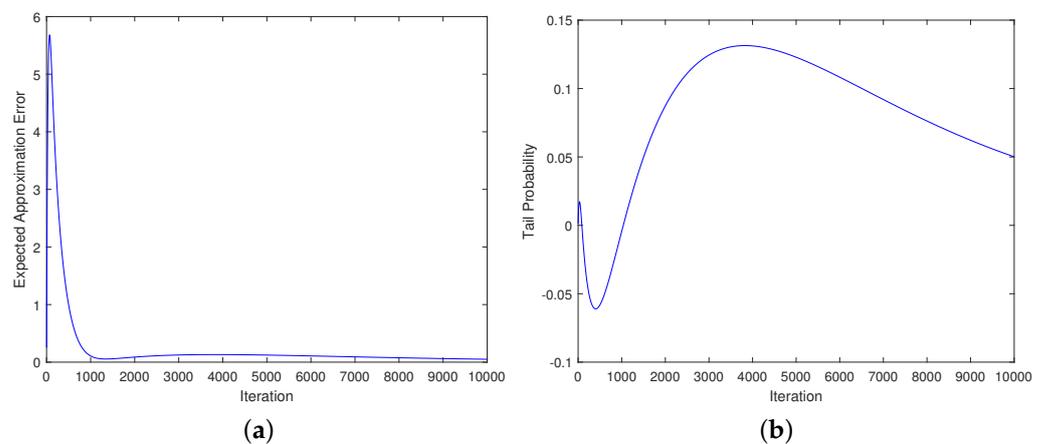
and the respective transition matrices are

$$\tilde{\mathbf{R}} = \begin{pmatrix} 1 & \frac{1}{n^3} & \frac{2}{n^3} & \dots & \frac{n}{n^3} \\ & 1 - \frac{1}{n^3} & \frac{1}{n^2} & & \\ & & 1 - \frac{1}{n^2} - \frac{2}{n^3} & \ddots & \\ & & & \ddots & \frac{n-1}{n^2} \\ & & & & 1 - \frac{1}{n} \end{pmatrix}$$

and

$$\tilde{\mathbf{S}} = \begin{pmatrix} 1 & \frac{2}{n^3} & \frac{4}{n^3} & \dots & \frac{2n}{n^3} \\ & 1 - \frac{2}{n^3} & \frac{1}{n^2} + \frac{1}{2n} & & \\ & & 1 - \frac{n^2+2n+8}{2n^3} & \ddots & \\ & & & \ddots & \frac{n-1}{n^2} + \frac{n-1}{2n} \\ & & & & 1 - \frac{n^2+n+2}{2n^2} \end{pmatrix}.$$

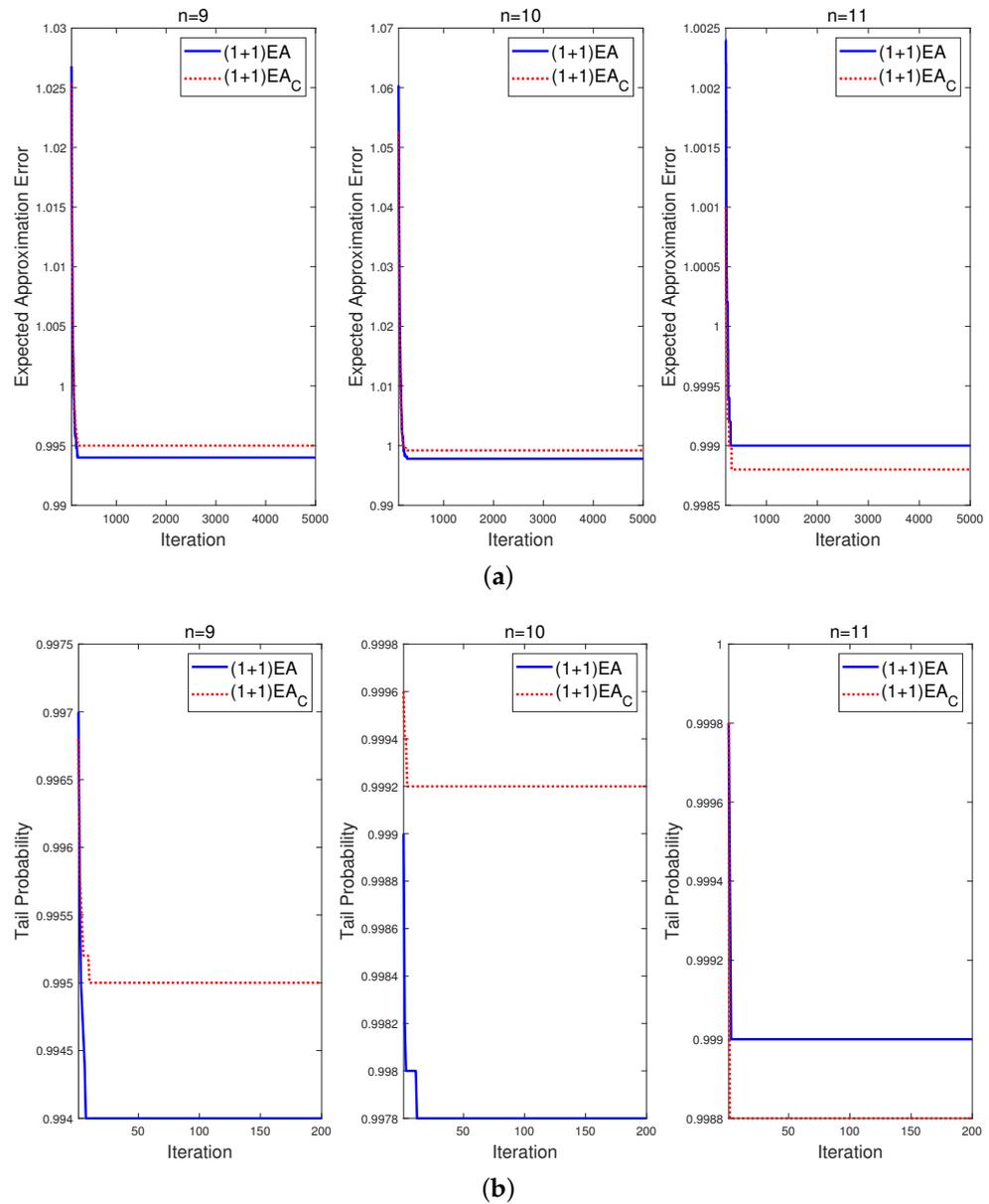
Obviously, it holds  $\tilde{\mathbf{S}} \succeq \tilde{\mathbf{R}}$ . Through computer simulation, we get the curve of EAE difference of the two EAs in Figure 1a and the curve of TPs difference between the two EAs in Figure 1b. From Figure 1b, it is clear that  $EA_{\mathcal{R}}$  does not always outperform  $EA_{\mathcal{S}}$  because the difference of TPs is negative at the early stage of the iteration process but later positive.



**Figure 1.** Simulation results on the difference of EAEs and TPs for the counterexample. (a) Difference of expected approximation errors (EAEs). (b) Difference of tail probabilities (TPs).

Now we turn to discuss  $(1 + 1)EA$  and  $(1 + 1)EA_C$  on Deceptive. We demonstrate  $(1 + 1)EA_C$  may not outperform  $(1 + 1)EA$  over all generations although the transition matrix of  $(1 + 1)EA_C$  dominates that of  $(1 + 1)EA$ .

**Example 5.** In  $(1 + 1)EA$  and  $(1 + 1)EA_C$ , set  $p_m = C_R q_m = 1/n$ . For  $(1 + 1)EA_C$ , let  $q_m = \frac{1}{2}$ ,  $C_R = \frac{2}{n}$ . The numerical simulation results of EAEs and TPs for 5000 independent runs are depicted in Figure 2. It is shown that when  $n \geq 9$ , both EAEs and TPs of  $(1 + 1)EA$  could be smaller than those of  $(1 + 1)EA_C$ . This indicates that the dominance of the transition matrix does not always guarantee the outperformance of the corresponding algorithm.



**Figure 2.** Numerical comparison for  $(1 + 1)EA$  and  $(1 + 1)EA_C$  applied to the Deceptive problem, where  $n$  refers to the problem dimension. (a) Numerical comparison of expected approximation errors (EAEs). (b) Numerical comparison of tail probabilities (TPs).

With  $p_m = C_R q_m = p \leq 1/n$ , although the binomial crossover leads to transition dominance of  $(1 + 1)EA_C$  over  $(1 + 1)EA$ , the enhancement of exploitation plays a governing role in the iteration process. Thus, the imbalance of exploration and exploitation leads to poor performance of  $(1 + 1)EA_C$  at some stage of the iteration process. As shown in the previous two examples, the outperformance of  $(1 + 1)EA_C$  cannot be drawn from the dominance of transition matrices.

The fitness landscape of Deceptive confirms that global convergence of EAs on Deceptive is principally attributed to the direct transition from level  $j$  to level 0, quantified

by the transition probability  $r_{0,j}$ . By investigating the impact of binomial crossover on the transition probability  $r_{0,j}$ , we arrive at an adaptive strategy for the regulation of the mutation rate and the crossover rate, by which performance of both  $(1 + 1)EA$  and  $(1 + 1)EA_C$  are enhanced.

7.2. Comparisons on the Probabilities to Transfer from Non-Optimal Statuses to the Optimal Status

A comparison between  $p_{0,j}$  and  $s_{0,j}$  is performed by investigating their monotonicity. Substituting (13) and (14) into (23) and (24), respectively, we have

$$p_{0,j} = P_1(n - j + 1, p_m) = (p_m)^{n-j+1}(1 - p_m)^{j-1}, \tag{45}$$

$$s_{0,j} = P_3(n - j + 1, C_R, q_m) = \frac{1}{n} [(j - 1)(1 - C_R) + nC_R(1 - q_m)] C_R^{n-j} (q_m)^{n-j+1} (1 - q_m C_R)^{j-2}. \tag{46}$$

We first investigate the maximum values of  $p_{0,j}$  to get the ideal performance of  $(1 + 1)EA$  on the Deceptive problem.

**Theorem 5.** While

$$p_m^* = \frac{n - j + 1}{n}, \tag{47}$$

$p_{0,j}$  gets its maximum values  $p_{0,j}^{max} = \left(\frac{n-j+1}{n}\right)^{n-j+1} \left(\frac{j-1}{n}\right)^{j-1}$ .

**Proof.** By (45), we know

$$\frac{\partial}{\partial p_m} p_{0,j} = (n - j + 1 - np_m) p_m^{n-j} (1 - p_m)^{j-2}.$$

While  $p_m = \frac{n-j+1}{n}$ ,  $p_{0,j}$  gets its maximum value

$$p_{0,j}^{max} = P_1\left(n - j + 1, \frac{n - j + 1}{n}\right) = \left(\frac{n - j + 1}{n}\right)^{n-j+1} \left(\frac{j - 1}{n}\right)^{j-1}.$$

□

Influence of the binomial crossover on  $s_{0,j}$  is investigated on condition that  $p_m = q_m$ . By regulating  $C_R$ , we compare  $p_{0,j}$  with the maximum value  $s_{0,j}^{max}$  of  $s_{0,j}$ .

**Theorem 6.** On condition that  $p_m = q_m$ , the following results hold.

1.  $p_{0,1} = s_{0,1}^{max}$ .
2. If  $q_m > \frac{n-1}{n}$ ,  $p_{0,2} < s_{0,2}^{max}$ ; otherwise,  $p_{0,2} = s_{0,2}^{max}$ .
3.  $\forall j \in \{3, \dots, n - 1\}$ ,  $p_{0,j} \leq s_{0,j}^{max}$  if  $q_m > \frac{n-j}{n-1}$ ; otherwise,  $s_{0,j}^{max} = p_{0,j}$ .
4. if  $q_m > \frac{1}{n}$ ,  $p_{0,n} < s_{0,n}^{max}$ ; otherwise,  $s_{0,n}^{max} = p_{0,n}$ .

**Proof.** Note that  $(1 + 1)EA_C$  degrades to  $(1 + 1)EA$  when  $C_R = 1$ . Then, if the maximum value  $s_{0,j}^{max}$  of  $s_{0,j}$  is obtained by setting  $C_R = 1$ , we have  $s_{0,j}^{max} = p_{0,j}$ ; otherwise, it holds  $s_{0,j}^{max} > p_{0,j}$ .

1. For the case that  $j = 1$ , Equation (46) implies

$$s_{0,1} = q_m^n (C_R)^{n-1}.$$

Obviously,  $s_{0,1}$  is monotonously increasing in  $C_R$ . It gets the maximum value while  $C_R^* = 1$ . Then, by (45) we get  $s_{0,1}^{max} = p_{0,1}$ .

2. While  $j = 2$ , by (46) we have

$$\frac{\partial s_{0,2}}{\partial C_R} = \frac{n-1}{n} q_m^{n-1} (C_R)^{n-3} (n-2+(1-nq_m)C_R).$$

- If  $0 < q_m \leq \frac{n-1}{n}$ ,  $s_{0,2}$  is monotonously increasing in  $C_R$ , and gets its maximum value while  $C_R^* = 1$ . For this case, we know  $s_{0,2}^{max} = p_{0,2}$ .
- While  $\frac{n-1}{n} < q_m < 1$ ,  $s_{0,2}$  gets its maximum value  $s_{0,2}^{max}$  by setting

$$C_R^* = \frac{n-2}{nq_m-1}. \tag{48}$$

Then, we have  $s_{0,2}^{max} > p_{0,2}$ .

3. For the case that  $3 \leq j \leq n-1$ , we denote

$$s_{0,j} = \frac{n-j+1}{n} q_m^{n-j+1} I_1 + \frac{(j-1)(1-q_m)}{n} q_m^{n-j+1} I_2,$$

where

$$I_1 = (C_R)^{n-j} (1-q_m C_R)^{j-1},$$

$$I_2 = (C_R)^{n-j+1} (1-q_m C_R)^{j-2}.$$

Then,

$$\frac{\partial I_1}{\partial C_R} = (C_R)^{n-j-1} (1-q_m C_R)^{j-2} (n-j-(n-1)q_m C_R),$$

$$\frac{\partial I_2}{\partial C_R} = (C_R)^{n-j} \left(1 - \frac{C_R}{n}\right)^{j-3} (n-j+1-(n-1)q_m C_R).$$

- While  $0 < q_m \leq \frac{n-j}{n-1}$ , both  $I_1$  and  $I_2$  are increasing in  $C_R$ . For this case,  $s_{0,j}$  gets its maximum value when  $C_R^* = 1$ , and we have  $s_{0,j}^{max} = p_{0,j}$ .
- If  $\frac{n-j+1}{n-1} \leq q_m \leq 1$ ,  $I_1$  gets its maximum value when  $C_R = \frac{n-j}{(n-1)q_m}$ , and  $I_2$  gets its maximum value when  $C_R = \frac{n-j+1}{(n-1)q_m}$ . Then,  $s_{0,j}$  get its maximum value  $s_{0,j}^{max}$  at some

$$C_R^* \in \left( \frac{n-j}{(n-1)q_m}, \frac{n-j+1}{(n-1)q_m} \right). \tag{49}$$

Accordingly, we know  $s_{0,j}^{max} > p_{0,j}$ .

- If  $\frac{n-j}{n-1} < q_m < \frac{n-j+1}{n-1}$ ,  $I_1$  gets its maximum value when  $C_R = \frac{n-j}{(n-1)q_m}$ , and  $I_2$  is monotonously increasing in  $C_R$ . Then,  $s_{0,j}$  get its maximum value  $s_{0,j}^{max}$  at some

$$C_R^* \in \left( \frac{n-j}{(n-1)q_m}, 1 \right], \tag{50}$$

and we know  $s_{0,j}^{max} > p_{0,j}$ .

4. While  $j = n$ , Equation (46) implies that

$$\frac{\partial s_{0,n}}{\partial C_R} = (n-1)(1-q_m C_R)^{n-3} (1-2q_m-(n-1-nq_m)q_m C_R).$$

Denoting

$$g(q_m, C_R) = 1-2q_m-(n-1-nq_m)q_m C_R,$$

we can confirm the sign of  $\partial s_{0,n} / \partial C_R$  by considering

$$\frac{\partial}{\partial C_R} g(q_m, C_R) = -(n - 1 - nq_m)q_m.$$

- While  $0 < q_m \leq \frac{n-1}{n}$ ,  $g(q_m, C_R)$  is monotonously decreasing in  $C_R$ , and its minimum value is

$$g(q_m, 1) = (nq_m - 1)(q_m - 1).$$

The maximum value of  $g(q_m, C_R)$  is

$$g(q_m, 0) = 1 - 2q_m.$$

- (a) If  $0 < q_m \leq \frac{1}{n}$ , we have

$$g(q_m, C_R) \geq g(q_m, 1) > 0.$$

Thus,  $\frac{\partial s_{0,n}}{\partial C_R} \geq 0$ , and  $s_{0,n}$  is increasing in  $C_R$ . For this case,  $s_{0,n}$  get its maximum value when  $C_R^* = 1$ , and we have  $s_{0,n}^{max} = p_{0,n}$ .

- (b) If  $\frac{1}{n} < q_m \leq \frac{1}{2}$ ,  $s_{0,n}$  gets the maximum value  $s_{0,n}^{max}$  when

$$C_R^* = \frac{1 - 2q_m}{q_m(n - 1 - nq_m)}.$$

Thus,  $s_{0,n}^{max} > p_{0,n}$ .

- (c) If  $\frac{1}{2} < q_m \leq \frac{n-1}{n}$ ,  $g(q_m, 0) < 0$ , and then,  $s_{0,n}$  is decreasing in  $C_R$ . Then, its maximum value is obtained by setting  $C_R^* = 0$ , and we know  $s_{0,n}^{max} > p_{0,n}$ .

- While  $\frac{n-1}{n} < q_m \leq 1$ ,  $g(q_m, C_R)$  is increasing in  $C_R$ , and its maximum value is

$$g(q_m, 1) = (nq_m - 1)(q_m - 1) < 0.$$

Then,  $s_{0,n}$  is monotonously decreasing in  $C_R$ , and its maximum value is obtained by setting  $C_R^* = 0$ . Accordingly, we know  $s_{0,n}^{max} > p_{0,n}$ .

In summary,  $s_{0,n}^{max} > p_{0,n}$  while  $q_m > \frac{1}{n}$ ; otherwise,  $s_{0,n}^{max} = p_{0,n}$ .

□

Theorems 5 and 6 present the “best” settings to maximize the transition probabilities from non-optimal statuses to the optimal level, by which we get a parameter adaptive strategy that greatly enhances the exploration of compared EAs.

### 7.3. Parameter Adaptive Strategy to Enhance Exploration of EAs

Since the level index  $j$  is equal to the Hamming distance between  $\mathbf{x}$  and  $\mathbf{x}^*$ , improvement of level index  $j$  is bounded by reduction of the Hamming distance obtained by replacing  $\mathbf{x}$  with  $\mathbf{y}$ . Then, while the local exploitation leads to a transition from level  $j$  to a non-optimal level  $i$ , the practically adaptive strategy of parameters can be obtained according to the Hamming distance between  $\mathbf{x}$  and  $\mathbf{y}$ .

When  $(1 + 1)EA$  is located at the solution  $\mathbf{x}$  at status  $j$ , Equation (47) implies that the “best” setting of mutation rate is  $p_m^*(j) = \frac{n-j+1}{n}$ . Once it transfers to solution  $\mathbf{y}$  at status  $i$  ( $i < j$ ), the “best” setting changes to  $p_m^*(i) = \frac{n-i+1}{n}$ . Then, the difference of “best” settings is  $\frac{j-i}{n}$ , bounded from above by  $\frac{H(\mathbf{x}, \mathbf{y})}{n}$ . Accordingly, the mutation rate of  $(1 + 1)EA$  can be updated to

$$p'_m = p_m + \frac{H(\mathbf{x}, \mathbf{y})}{n}. \tag{51}$$

For  $(1 + 1)EA_C$ , the parameter  $q_m$  is adapted using the strategy consistent to that of  $p_m$  to focus on influence of  $C_R$ . That is,

$$q'_m = q_m + \frac{H(\mathbf{x}, \mathbf{y})}{n}. \tag{52}$$

Since  $s_{0,j}$  demonstrates different monotonicity for varied levels, one cannot get an identical strategy for the adaptive setting of  $C_R$ . As a compromise, we would like to consider the case that  $3 \leq j \leq n - 1$ , which is obtained by random initialization with overwhelming probability.

According to the proof of Theorem 6, we know  $C_R$  should be set as great as possible for the case  $q_m \in (0, \frac{n-j}{n-1}]$ ; while  $q_m \in (\frac{n-j}{n-1}, 1]$ ,  $C_R^*$  is located in intervals whose boundary values are  $\frac{n-j}{(n-1)q_m}$  and  $\frac{n-j+1}{(n-1)q_m}$ , given by (49) and (50), respectively. Then, while  $q_m$  is updated by (52), the update strategy of  $C_R$  can be confirmed to satisfy that

$$C'_R q'_m = C_R q_m + \frac{H(\mathbf{x}, \mathbf{y})}{n-1}.$$

Accordingly, the adaptive setting of  $C_R$  could be

$$C'_R = \left( C_R q_m + \frac{H(\mathbf{x}, \mathbf{y})}{n-1} \right) / q'_m, \tag{53}$$

where  $q'_m$  is updated by (52).

By incorporating the adaptive strategy (51) to  $(1 + 1)EA$ , we compare the performance of its adaptive variant with the adaptive  $(1 + 1)EA_C$  that regulates its mutation rate and crossover rate by (52) and (53), respectively. For 13–20 dimensional Deceptive problems, numerical simulation of the tail probability is implemented by 10,000 independent runs. The initial value of  $p_m$  is set as  $\frac{1}{n}$ . To investigate the sensitivity of the adaptive strategy on initial values of  $q_m$ , the mutation rate  $q_m$  in  $(1 + 1)EA_C$  is initialized with values  $\frac{1}{\sqrt{n}}$ ,  $\frac{3}{2\sqrt{n}}$  and  $\frac{2}{\sqrt{n}}$ , and the corresponding variants are denoted by  $(1 + 1)EA_C^1$ ,  $(1 + 1)EA_C^2$  and  $(1 + 1)EA_C^3$ , respectively.

The converging curves of averaged TPs are illustrated in Figure 3. Compared to the EAs with fixed parameters during the evolution process, the performance of the adaptive EAs on Deceptive has been significantly improved. Furthermore, we also note that the converging curves of adaptive  $(1 + 1)EA_C$  are not sensitive to the initial mutation rate. Although transition dominance does not necessarily lead to outperformance of  $(1 + 1)EA_C$  over  $(1 + 1)EA$ , the proposed adaptive strategy can greatly enhance global exploration of  $(1 + 1)EA_C$  to a large extent, and consequently, we get the improved adaptive  $(1 + 1)EA_C$  that is not sensitive to initial mutation rates.

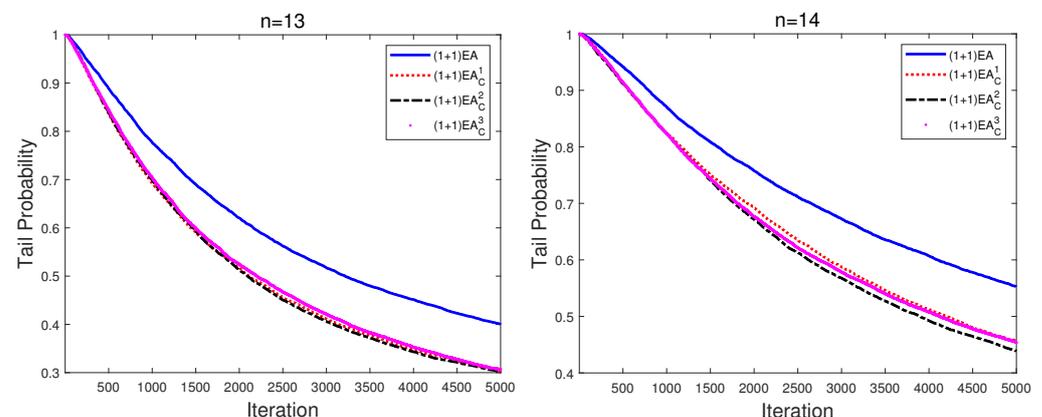
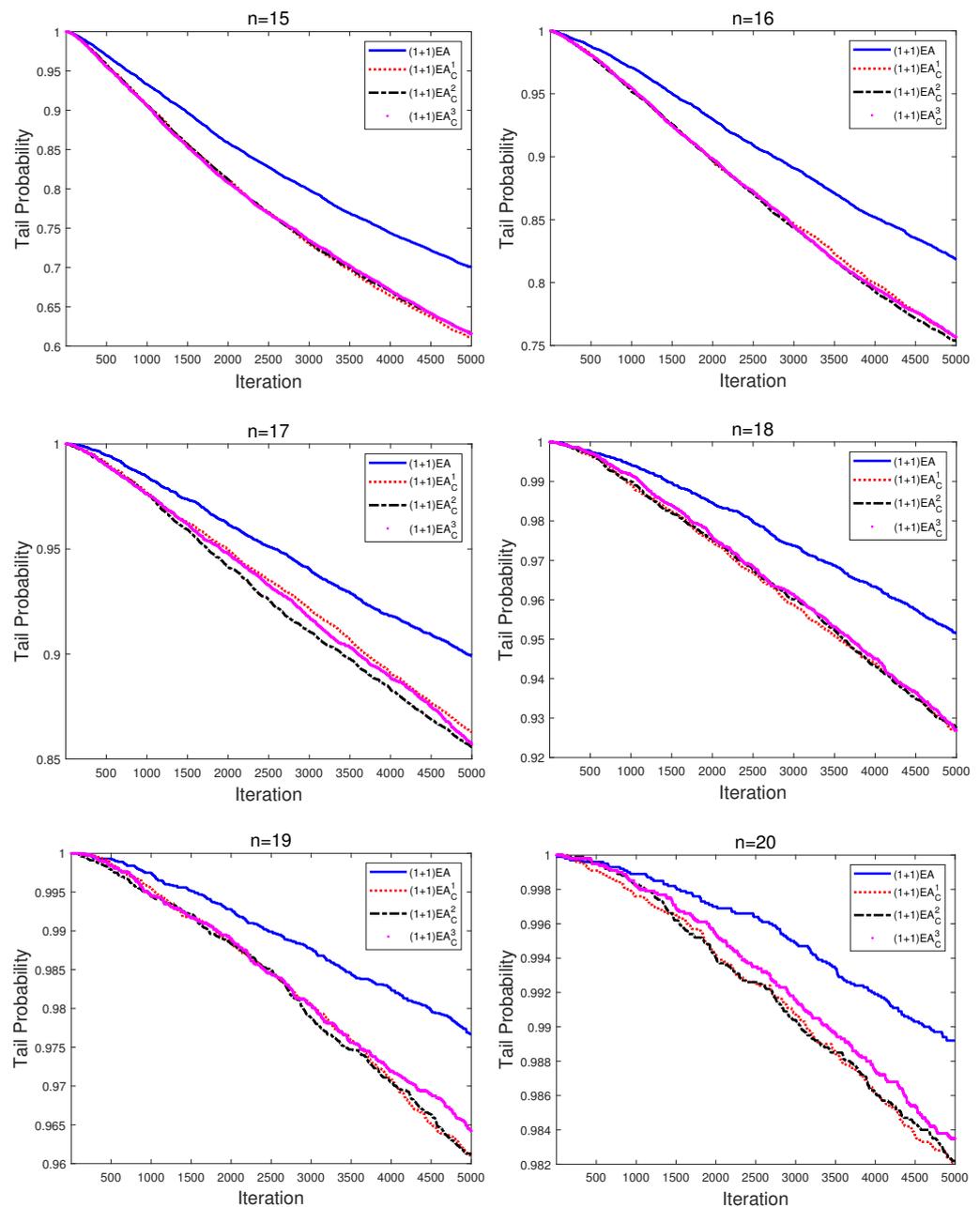


Figure 3. Cont.



**Figure 3.** Numerical comparison on tail probabilities (TPs) of adaptive  $(1+1)EA$  and  $(1+1)EA_C$  applied to the Deceptive problem, where  $n$  is the problem dimension.  $(1+1)EA_C^1$ ,  $(1+1)EA_C^2$ , and  $(1+1)EA_C^3$  are three variants of  $(1+1)EA_C$  with  $q_m$  initialized as  $\frac{1}{\sqrt{n}}$ ,  $\frac{3}{2\sqrt{n}}$ , and  $\frac{2}{\sqrt{n}}$ , respectively.

### 8. Conclusions and Discussions

Under the framework of fixed-budget analysis, we conduct a pioneering analysis of the influence of binomial crossover on the approximation error of EAs. The performance of EAs after running finite generations is measured by two metrics: the expected value of the approximation error and the error tail probability, by which we make a case study by comparing the performance of  $(1+1)EA$  and  $(1+1)EA_C$  with binomial crossover.

Starting from the comparison of the probability of flipping “1 preferred bits”, it is proven that under proper conditions, incorporation of binomial crossover leads to the dominance of transition probabilities, that is, the probability of transferring to any promising status is improved. Accordingly, the asymptotic performance of  $(1+1)EA_C$  is superior to that of  $(1+1)EA$ .

It is found that the dominance of transition probability guarantees that  $(1 + 1)EA_C$  outperforms  $(1 + 1)EA$  on OneMax in terms of both expected approximation error and tail probability. However, this dominance does not lead to the outperformance on Deceptive. This means that using binomial crossover may improve the performance on some problems but not on other problems.

For Deceptive, an adaptive strategy of parameter setting is proposed based on the monotonicity analysis of transition probabilities. Numerical simulations demonstrate that it can significantly improve the exploration ability of both  $(1 + 1)EA_C$  and  $(1 + 1)EA$ , and superiority of binomial crossover is further strengthened by the adaptive strategy. Thus, a problem-specific adaptive strategy is helpful for improving the performance of EAs.

Our future work will focus on a further study for the adaptive setting of crossover rate in population-based EAs on more complex problems, as well as the development of adaptive EAs improved by the introduction of binomial crossover.

**Author Contributions:** Conceptualization, J.H. and X.Z.; formal analysis, C.W.; writing—original draft preparation, C.W.; writing—review and editing, Y.C. and J.H.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Fundamental Research Funds for the Central Universities grant number WUT:2020IB006.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tam, H.H.; Leung, M.F.; Wang, Z.; Ng, S.C.; Cheung, C.C.; Lui, A.K. Improved adaptive global replacement scheme for MOEA/D-AGR. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 2153–2160.
2. Tam, H.H.; Ng, S.C.; Lui, A.K.; Leung, M.F. Improved activation schema on automatic clustering using differential evolution algorithm. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, Spain, 5–8 June 2017; pp. 1749–1756.
3. Gao, W.; Li, G.; Zhang, Q.; Luo, Y.; Wang, Z. Solving nonlinear equation systems by a two-phase evolutionary algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 5652–5663. [\[CrossRef\]](#)
4. Jansen, T.; Wegener, I. The analysis of evolutionary algorithms—A proof that crossover really can help. *Algorithmica* **2002**, *34*, 47–66. [\[CrossRef\]](#)
5. Kötzing, T.; Sudholt, D.; Theile, M. How crossover helps in pseudo-boolean optimization. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; pp. 989–996.
6. Corus, D.; Oliveto, P.S. Standard steady state genetic algorithms can hillclimb faster than mutation-only evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2017**, *22*, 720–732. [\[CrossRef\]](#)
7. Dang, D.C.; Friedrich, T.; Kötzing, T.; Krejca, M.S.; Lehre, P.K.; Oliveto, P.S.; Sudholt, D.; Sutton, A.M. Escaping local optima using crossover with emergent diversity. *IEEE Trans. Evol. Comput.* **2017**, *22*, 484–497. [\[CrossRef\]](#)
8. Sudholt, D. How crossover speeds up building block assembly in genetic algorithms. *Evol. Comput.* **2017**, *25*, 237–274. [\[CrossRef\]](#)
9. Pinto, E.C.; Doerr, C. A simple proof for the usefulness of crossover in black-box optimization. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Coimbra, Portugal, 8–12 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 29–41.
10. Oliveto, P.S.; Sudholt, D.; Witt, C. A tight lower bound on the expected runtime of standard steady state genetic algorithms. In Proceedings of the the 2020 Genetic and Evolutionary Computation Conference, Cancun, Mexico, 8–12 July 2020; pp. 1323–1331.
11. Lengler, J.; Meier, J. Large population sizes and crossover help in dynamic environments. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Leiden, The Netherlands, 5–9 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 610–622.
12. Lehre, P.K.; Yao, X. Crossover can be constructive when computing unique input output sequences. In Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning, Melbourne, Australia, 7–10 December 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 595–604.

13. Doerr, B.; Happ, E.; Klein, C. Crossover can provably be useful in evolutionary computation. *Theor. Comput. Sci.* **2012**, *425*, 17–33. [[CrossRef](#)]
14. Doerr, B.; Johannsen, D.; Kötzing, T.; Neumann, F.; Theile, M. More effective crossover operators for the all-pairs shortest path problem. *Theor. Comput. Sci.* **2013**, *471*, 12–26. [[CrossRef](#)]
15. Sutton, A.M. Fixed-parameter tractability of crossover: Steady-state GAs on the closest string problem. *Algorithmica* **2021**, *83*, 1138–1163. [[CrossRef](#)]
16. Richter, J.N.; Wright, A.; Paxton, J. Ignoble trails—where crossover is provably harmful. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Dortmund, Germany, 13–17 September 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 92–101.
17. Antipov, D.; Naumov, S. The effect of non-symmetric fitness: The analysis of crossover-based algorithms on RealJump functions. In Proceedings of the the 16th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, Virtual, 6–8 September 2021; pp. 1–15.
18. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [[CrossRef](#)]
19. Das, S.; Mullick, S.S.; Suganthan, P. Recent advances in differential evolution—An updated survey. *Swarm Evol. Comput.* **2016**, *27*, 1–30. [[CrossRef](#)]
20. Sepesy Maučec, M.; Brest, J. A review of the recent use of differential evolution for large-scale global optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite. *Swarm Evol. Comput.* **2019**, *50*, 100428. [[CrossRef](#)]
21. Pant, M.; Zaheer, H.; Garcia-Hernandez, L.; Abraham, A. Differential evolution: A review of more than two decades of research. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103479. [[CrossRef](#)]
22. Lin, C.; Qing, A.; Feng, Q. A comparative study of crossover in differential evolution. *J. Heuristics* **2011**, *17*, 675–703. [[CrossRef](#)]
23. Gong, T.; Tuson, A.L. Differential evolution for binary encoding. In *Soft Computing in Industrial Applications*; Saad, A., Dahal, K., Sarfraz, M., Roy, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 251–262.
24. Doerr, B.; Zheng, W. Working principles of binary differential evolution. *Theor. Comput. Sci.* **2020**, *801*, 110–142. [[CrossRef](#)]
25. Chen, Y.; He, J. Average convergence rate of evolutionary algorithms in continuous optimization. *Inf. Sci.* **2021**, *562*, 200–219. [[CrossRef](#)]
26. Xu, T.; He, J.; Shang, C. Helper and equivalent objectives: Efficient approach for constrained optimization. *IEEE Trans. Cybern.* **2022**, *52*, 240–251. [[CrossRef](#)] [[PubMed](#)]
27. Jansen, T.; Zarges, C. Performance analysis of randomised search heuristics operating with a fixed budget. *Theor. Comput. Sci.* **2014**, *545*, 39–58. [[CrossRef](#)]
28. He, J. An analytic expression of relative approximation error for a class of evolutionary algorithms. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 4366–4373.
29. Wang, C.; Chen, Y.; He, J.; Xie, C. Error analysis of elitist randomized search heuristics. *Swarm Evol. Comput.* **2021**, *63*, 100875. [[CrossRef](#)]
30. Jansen, T.; Zarges, C. Reevaluating Immune-Inspired Hypermutations Using the Fixed Budget Perspective. *IEEE Trans. Evol. Comput.* **2014**, *18*, 674–688. [[CrossRef](#)]
31. He, J.; Yao, X. Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artif. Intell.* **2003**, *145*, 59–97. [[CrossRef](#)]
32. He, J.; Lin, G. Average convergence rate of evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2016**, *20*, 316–321. [[CrossRef](#)]
33. Opara, K.R.; Arabas, J. Differential evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2019**, *44*, 546–558. [[CrossRef](#)]
34. Zhou, Y.; Yi, W.; Gao, L.; Li, X. Analysis of mutation vectors selection mechanism in differential evolution. *Appl. Intell.* **2016**, *44*, 904–912. [[CrossRef](#)]
35. Zaharie, D. Influence of crossover on the behavior of differential evolution algorithms. *Appl. Soft Comput.* **2009**, *9*, 1126–1138. [[CrossRef](#)]
36. Zaharie, D. Statistical properties of differential evolution and related random search algorithms. In *COMPSTAT 2008: Proceedings in Computational Statistics*; Brito, P., Ed.; Physica: Heidelberg, Germany, 2008; pp. 473–485.
37. Zaharie, D.; Micota, F. Revisiting the analysis of population variance in differential evolution algorithms. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, Spain, 5–8 June 2017; pp. 1811–1818.
38. Wang, L.; Huang, F.Z. Parameter analysis based on stochastic model for differential evolution algorithm. *Appl. Math. Comput.* **2010**, *217*, 3263–3273. [[CrossRef](#)]
39. Opara, K.R.; Arabas, J. Comparison of mutation strategies in differential evolution—A probabilistic perspective. *Swarm Evol. Comput.* **2018**, *39*, 53–69. [[CrossRef](#)]
40. Opara, K.R.; Arabas, J. The contour fitting property of differential mutation. *Swarm Evol. Comput.* **2019**, *50*, 100441. [[CrossRef](#)]
41. Jansen, T.; Zarges, C. Evolutionary algorithms and artificial immune systems on a bi-stable dynamic optimisation problem. In Proceedings of the 16th Annual Conference on Genetic and Evolutionary Computation, Vancouver, BC, Canada, 12–16 July 2014; pp. 975–982.

- 
42. Nallaperuma, S.; Neumann, F.; Sudholt, D. Expected fitness gains of randomized search heuristics for the traveling salesperson problem. *Evol. Comput.* **2017**, *25*, 673–705. [[CrossRef](#)]
  43. Doerr, B.; Jansen, T.; Witt, C.; Zarges, C. A method to derive fixed budget results from expected optimisation times. In Proceedings of the the 15th Annual Conference on Genetic and Evolutionary Computation, Amsterdam, The Netherlands, 6–10 July 2013; pp. 1581–1588.