



Article

# Running-Time Analysis of Brain Storm Optimization Based on Average Gain Model

Guizhen Mai <sup>1</sup>, Fangqing Liu <sup>2,\*</sup> , Yinghan Hong <sup>1,\*</sup>, Dingrong Liu <sup>2</sup>, Junpeng Su <sup>2</sup>, Xiaowei Yang <sup>2</sup> and Han Huang <sup>2</sup>

<sup>1</sup> School of Physics and Electronic Engineering, Hanshan Normal University, Chaozhou 521041, China; 20200043@hstc.edu.cn

<sup>2</sup> School of Software Engineering, South China University of Technology, Guangzhou 510006, China; liudingrong@gzgs.edu.cn (D.L.); sujp.scut@outlook.com (J.S.); xwyang@scut.edu.cn (X.Y.); hhan@scut.edu.cn (H.H.)

\* Correspondence: peerfog@scut.edu.cn (F.L.); honyinghan@163.com (Y.H.)

**Abstract:** The brain storm optimization (BSO) algorithm has received increased attention in the field of evolutionary computation. While BSO has been applied in numerous industrial scenarios due to its effectiveness and accessibility, there are few theoretical analysis results about its running time. Running-time analysis can be conducted through the estimation of the upper bounds of the expected first hitting time to evaluate the efficiency of BSO. This study estimates the upper bounds of the expected first hitting time on six single individual BSO variants (BSOs with one individual) based on the average gain model. The theoretical analysis indicates the following results. (1) The time complexity of the six BSO variants is  $O(\sqrt{n})$  in equal coefficient linear functions regardless of the presence or absence of the disrupting operator, where  $n$  is the number of the dimensions. Moreover, the coefficient of the upper bounds on the expected first hitting time shows that the single individual BSOs with the disrupting operator require fewer iterations to obtain the target solution than the single individual BSOs without the disrupting operator. (2) The upper bounds on the expected first hitting time of single individual BSOs with the standard normally distributed mutation operator are lower than those of BSOs with the uniformly distributed mutation operator. (3) The upper bounds on the expected first hitting time of single individual BSOs with the  $U(-\frac{1}{2}, \frac{1}{2})$  mutation operator are approximately twice those of BSOs with the  $U(-1, 1)$  mutation operator. The corresponding numerical results are also consistent with the theoretical analysis results.

**Keywords:** brain storm optimization (BSO); running time; average gain model; mutation operator; linear function



**Citation:** Mai, G.; Liu, F.; Hong, Y.; Liu, D.; Su, J.; Yang, X.; Huang, H. Running-Time Analysis of Brain Storm Optimization Based on Average Gain Model. *Biomimetics* **2024**, *9*, 117. <https://doi.org/10.3390/biomimetics9020117>

Academic Editor: Heming Jia

Received: 24 November 2023

Revised: 24 January 2024

Accepted: 31 January 2024

Published: 15 February 2024



**Copyright:** © 2024 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

The swarm intelligence algorithm is one of the nature-inspired optimization algorithms that simulates the behavior of biological groups in nature [1–3]. Over the past two decades, many different types of swarm intelligence algorithms have been proposed, such as particle swarm optimization (PSO) [4,5], ant colony optimization (ACO) [6,7], artificial bee colony (ABC) [8,9], and brain storm optimization (BSO) [10–12]. Different from traditional methods, these algorithms solve problems by simulating the behavior of animal or human groups, with higher flexibility and adaptability.

BSO, a novel swarm intelligence algorithm, is inspired by the human brainstorming process. It is a continuous evolutionary algorithm that simulates the collective behavior of human beings. In recent years, BSOs have seen various practical applications in power systems [13–16], aviation design [17–19], mobile robot path planning [20,21], antenna design [22], financial optimization [23–25], and many other fields [26–30].

In addition, the theoretical analysis of BSO is also very important, especially for the practical application of BSO. The theoretical analysis benefits researchers in enabling them to understand the mechanism of the algorithm in guiding its design, improvement, and

application in practice. The theoretical analysis can be divided into convergence analysis and running-time analysis. Zhou et al. [31], Qiao et al. [32], and Zhang et al. [33] have performed corresponding convergence analyses on BSO. The BSO-CAPSO algorithm, proposed by Alkmini [1], effectively enhances the computational efficiency of BSO through hybridization with chaotic accelerated particle swarm optimization. However, there are few works on the running-time analysis of BSO.

Some theoretical methods have been proposed as general analysis tools to investigate the running time of random heuristic algorithms, including the fitness level method [34], drift analysis method [35], switch analysis method [36], wave model [37], etc. These methods are mainly used to analyze discrete random heuristic algorithms. In contrast, fewer theoretical analysis results have been obtained for continuous random heuristic algorithms [38–40]. However, a large number of practical application problems are continuous. Therefore, the running time of continuous random heuristic algorithms has important research significance. To analyze the running time of continuous random heuristic algorithms, Huang et al. [41] proposed the average gain model.

Huang et al. [41,42] and Zhang et al. [43] used an average gain model to evaluate the expected first hit time of the  $(1 + 1)$ -evolutionary algorithm ( $(1 + 1)$  EA), evolutionary strategy (ES), and covariance matrix adaptation evolution strategy (CMA-ES). The concept of the first hit time refers to the minimum number of iterations required before the algorithm finds an optimal solution [35]. The expected first hit time represents the average number of iterations needed to find the optimal solution, which actually reflects the average time complexity of the algorithm [44]. Wang [45] employed the average gain model to analyze the computational efficiency of the proposed swarm intelligence algorithm and provided theoretical evidence for its effectiveness. Therefore, the expected first hit time is a core metric in runtime analysis. Based on the average gain model, the expected first hit time of BSO is deeply analyzed in this paper.

The core of BSO consists of three key components: clustering, interruption, and update. The mutation operator plays an important role in BSO and is included in the interrupt operation and update operation. Specifically, the mutation operator helps the algorithm to jump out of the local optimal solution and further explore a broader search space by introducing random factors in the search process. For example, Zhan et al. [46] proposed an improved BSO (MBSO) in which the mutation operator employs a novel thought difference strategy (IDS). This strategy takes advantage of the thought differences among individuals in the group and increases the diversity of the group by introducing random factors, thus increasing the probability of the algorithm finding the global optimal solution. In addition, El-Abd [47] improved the step equation of the mutation operator and improved the performance of BSO by adjusting the step size and distribution. This improvement helps the algorithm to balance the local search and the global search better, so that the algorithm can find the global optimal solution more effectively when solving complex optimization problems.

The time complexity of the single individual BSO is analyzed in this paper based on the research process from simple to complex. The single individual BSO without the disrupting operation is the same as the  $(1+1)$  ES [41]. However, the corresponding results can explain the influence of the mutation operator and disrupting operator on the time complexity of BSO. In this paper, we choose the three most classic and representative distributions,  $N(0, 1)$ ,  $U\left(-\frac{1}{2}, \frac{1}{2}\right)$ , and  $U(-1, 1)$ , as the analysis objects for the mutation operator. Therefore, six BSO variants are obtained as the analyzed algorithms based on the combination of three mutation operators and the presence or absence of a disrupting operator.

The remainder of this paper is organized as follows. Section 2 introduces the process of BSO and the mathematical model of running-time analysis for BSO. Section 3 provides the theoretical analysis results for the running-time analysis of three different BSO variants. Section 4 presents the corresponding experimental results to evaluate the theoretical results. Finally, Section 5 concludes the paper.

## 2. Mathematical Model for Running-Time Analysis of BSO

### 2.1. Brain Storm Optimization

BSO was proposed by Shi [48,49] in 2011, and it is simple in concept and easy to implement. BSO can be simplified as follows.

In Steps 4 and 5, the new solution is generated by  $\tilde{x} = x + \Delta$ , where  $x$  is the original individual,  $\tilde{x}$  is the newly generated individual, and  $\Delta$  is a vector generated according to the mutation operator. In this paper, we focus on the running time of BSO with three different mutation operators. The same mutation operators are used to generate new individuals in both Steps 4 and 5. The superposition of different mutation operators is not considered in this work.

To accurately observe the effects of the disrupting and updating operations on the running time of BSO, we select a single individual form of BSO as the analyzed object. The single individual BSO framework simplifies the effect of the population size, which helps to evaluate the effect of the disrupting and updating operations on the running time. Furthermore, following the principle from simple to difficult, the single individual BSO is a suitable starting point for the running-time analysis of BSO. Moreover, the randomness of  $\Delta$  and the design of the operations in Steps 4 and 5 are derived from the mutation operator design of evolutionary programming [50,51]. Therefore, the conclusion of this analysis will have positive implications for the study of similar mutation operators in evolutionary programming algorithms.

### 2.2. Stochastic Process Model of BSO

BSO can be represented as a stochastic process. In this section, we introduce the terminology for the analysis of the running time of BSO.

**Definition 1** (Hill-climbing task). *Given a search space  $S \subseteq R^n$  and a fitness function  $f : S \rightarrow R$ , the hill-climbing task is to find a solution  $\tilde{x}^* \in S$ , where the fitness of  $\tilde{x}^*$  reaches the target value  $H$ , where  $f(\tilde{x}^*) \geq H$ .*

In this paper, we focus on analyzing the BSO running time in the hill-climbing task of a continuous search space.

**Definition 2** (State of BSO). *The state of BSO at the  $t$ -th ( $t = 0, 1, \dots$ ) iteration is defined as  $P_t = \{\vec{\zeta}_1^t, \vec{\zeta}_2^t, \dots, \vec{\zeta}_\lambda^t\}$ , where  $\lambda$  is the size of the population, and  $\vec{\zeta}_1^t, \vec{\zeta}_2^t, \dots, \vec{\zeta}_\lambda^t \in S$ .*

**Definition 3** (State space of BSO). *The set of all possible BSO states is called the state space of BSO, denoted as*

$$\Omega = S^\lambda = \left\{ \left\{ \vec{\zeta}_1, \vec{\zeta}_2, \dots, \vec{\zeta}_\lambda \right\} \mid \vec{\zeta}_k \in S, k = 1, \dots, \lambda \right\}. \tag{1}$$

An optimization problem is a mapping from a decision space to an objective space, and the state space of BSO represents the corresponding decision space.

**Definition 4** (Expected first hitting time). *Let  $\{X_t\}_{t=0}^\infty$  be a stochastic process, where, for any  $t \geq 0$ ,  $X_t \geq 0$  holds. Suppose that  $X_t$  is the Euclidean distance value of the  $t$ -th iteration state of BSO to the target solution, and the target threshold  $\varepsilon > 0$ , the first hitting time [35] of the  $\varepsilon$ -approximation solution, can be defined by*

$$T_\varepsilon = \min\{t \geq 0 : X_t \leq \varepsilon\}. \tag{2}$$

Therefore, the expected first hitting time [44] of BSO can be denoted with  $E(T_\varepsilon | X_0)$ .

The expected first hit time refers to the average number of iterations required for the BSO algorithm to reach the target fitness value. This metric can more accurately measure the performance of an algorithm because it takes into account the probability distribution

of the algorithm over different iterations. Through this metric, we can evaluate the average time complexity of the algorithm in finding the optimal solution, so as to better understand the efficiency of the algorithm.

### 2.3. Running-Time Analysis of BSO Based on Average Gain Model

Inspired by drift analysis [52] and the idea of Riemann integrals [53], Huang et al. [41] proposed the average gain model. Zhang et al. [43] separated this model by introducing the concepts of the supermartingale and stopping time. Based on the former research results [41,43] of the average gain model, Huang et al. [42] proposed an experimental method to estimate the running time of the continuous evolution algorithms.

The expected one-step variation

$$\delta_t = E(X_t - X_{t+1}|H_t), t \geq 0 \tag{3}$$

is called the average gain, where  $\{X_t\}_{t=0}^\infty$  is a stochastic process, and  $H_t = \sigma(X_0, X_1, \dots, X_t)$ . The  $\sigma$ -algebra  $H_t$  contains all the events generated by  $X_0, X_1, \dots, X_t$ . All the information observed from the original population to the  $t$ -th iteration is recorded in  $H_t$ .

Based on Definition 2,  $P_t = \{\vec{\zeta}_1^t, \vec{\zeta}_2^t, \dots, \vec{\zeta}_\lambda^t\}$  is the state of BSO at the  $t$ -th iteration. The process of BSO in solving the hill-climbing task is considered as the gradual process of the stochastic state from the initial population to the population that contains the optimal solution. Let  $f(P_t) = \max\{f(\vec{x}) : \vec{x} \in P_t\}$  be the highest fitness value of individuals in  $P_t$ .  $X_t$  is used to measure the distance of the current population to the population of the target value.  $X_t = f^* - f(P_t)$ , where  $f^*$  is the fitness value of the optimal solution. Obviously,  $\{X_t\}_{t=0}^\infty$  is a non-negative stochastic process.

The state of BSO in the  $(t + 1)$ -th iteration  $P_{t+1}$  only depends on  $P_t$ . In other words, the stochastic process  $\{P_t\}_{t=0}^\infty$  can be modeled by a Markov chain [42]. Similarly,  $\{X_t\}_{t=0}^\infty$  can also be regarded as a Markov chain. In this case, the average gain  $\delta_t = E(X_t - X_{t+1}|H_t)$  can be simplified to  $\delta_t = E(X_t - X_{t+1}|X_t)$ . Based on Th. 2 of [43], the expectation of  $T_\epsilon$  of BSO can be estimated as follows.

**Theorem 1.** Suppose that  $\{X_t\}_{t=0}^\infty$  is a stochastic process associated with BSO, where  $X_t \geq 0$  for all  $t \geq 0$ . Let  $h : [0, A] \rightarrow R^+$  be a monotonically increasing and integrable function. If  $E(X_t - X_{t+1}|X_t) \geq h(X_t)$  and  $X_t > \epsilon > 0$ , it holds for  $T_\epsilon$  that

$$E(T_\epsilon|X_0) \leq 1 + \int_\epsilon^{X_0} \frac{1}{h(x)} x. \tag{4}$$

Theorem 1 shows the upper bounds on the expected first hitting time of BSO based on the average gain model. The average gain  $\delta_t$  plays a key role in analyzing the first hitting time  $T_\epsilon$  of the  $\epsilon$ -approximation solution for BSO. The higher average gain indicates a more efficient iteration of the optimization process.

### 3. Running-Time Analysis of BSO Instances for Equal Coefficient Linear Functions

In this section, we present the theoretical analysis results based on the average gain model to analyze the expected first hitting time of BSO for equal coefficient linear functions. The running time of BSO with three different mutation operators is analyzed from the perspective of whether the disrupting operation exists. In this paper, we refer to the BSO without a disrupting operation as BSO-I, and the BSO with a disrupting operation as BSO-II.

On this basis, the equal coefficient linear functions are selected as the research object [54–56]. These functions are a form of basic continuous optimization problem whose function expression is as follows:

$$f(x_1, x_2, \dots, x_n) = k(x_1 + x_2 + \dots + x_n) = k \sum_{i=1}^n x_i, \tag{5}$$

where  $(x_1, x_2, \dots, x_n) \in S$ . It is assumed that the function starts from the origin and sets the target fitness value to  $na$ , where  $a > 0$ . The objective of optimizing the equal coefficient linear function is to find a solution  $\vec{x}^* \in S$ , such that  $f(\vec{x}^*) \geq na$ .

The mutation operator that obeys the Gaussian distribution and the uniform distribution is selected for the evaluation of BSO. The Gaussian distribution and uniform distribution are common tools for the design of mutation operators [50,51,57], so it is representative to select these two distributions as research cases.

### 3.1. Case Study of BSO without Disrupting Operator

Since the single individual BSO is analyzed in this paper,  $\lambda$  is equal to 1 in the state  $P_t = \{\zeta_1^t, \zeta_2^t, \dots, \zeta_\lambda^t\}$  of BSO. The BSO of a single individual has only one individual, so  $\zeta_1^t$  represents both the optimized individual and the random state of the algorithm. The procedure of single individual BSO can be described as follows when the disrupting operation does not exist (i.e., Step 4 in Algorithm 1 is ignored).

---

#### Algorithm 1 Brain Storm Optimization (BSO)

---

- 1: **Initialization:** Randomly generate  $\lambda$  individuals (potential solutions) to form the initial population  $P = \{\zeta_1, \zeta_2, \dots, \zeta_\lambda\}$  and evaluate the  $\lambda$  individuals;
  - 2: **while** fail to achieve the predetermined maximum number of iterations **do**
  - 3:   **Clustering:** Use clustering algorithms to divide  $\lambda$  individuals into  $m$  clusters;
  - 4:   **Disrupting:** The mutation occurs with a certain probability, and a randomly selected cluster's central individual is replaced by a randomly generated new individual;
  - 5:   **Updating:** Randomly choose one or two clusters to create a new individual; Compare the newly generated individual and the original individual with the same individual index. The better one will be saved as the new individual; Update the whole population; the offspring population is recorded as  $P' = \{\zeta'_1, \zeta'_2, \dots, \zeta'_\lambda\}$ . Evaluate the individuals in  $P'$ ;
  - 6: **end while**
  - 7: Output the most optimal solution discovered.
- 

$\vec{x}^t = (x_1^t, x_2^t, \dots, x_n^t) \in S$ ,  $t = 0, 1, \dots$  is the  $t$ -th generation of the algorithm.

$$X_t = an - f(\vec{x}^t) = an - k(x_1^t + x_2^t + \dots + x_n^t), \tag{6}$$

is defined as the Euclidean distance of the  $t$ -th iteration to the optimal solution. The gain at  $t$ -th is given by

$$\begin{aligned} \eta_t &= X_t - X_{t+1} \\ &= k(x_1^{t+1} + x_2^{t+1} + \dots + x_n^{t+1}) - k(x_1^t + x_2^t + \dots + x_n^t). \end{aligned} \tag{7}$$

#### 3.1.1. When $z_i \sim N(0, 1)$

If the mutation operator obeys the standard normal distribution  $N(0, 1)$ , the distribution function of  $\eta_t$  is as presented by Lemma 1.

**Lemma 1.** For BSO-I, if its mutation operator  $\vec{z}$  obeys  $N(0, 1)$ , the distribution function  $F(u) = P(\eta_t \leq u)$  of the gain  $\eta_t$  is

$$F(u) = \begin{cases} 0, & u < 0 \\ \frac{1}{2}, & u = 0. \\ \frac{1}{\sqrt{2\pi nk}} \int_{-\infty}^u e^{-\frac{t^2}{2nk^2} t}, & u > 0 \end{cases} \tag{8}$$

**Proof.** According to Step 4 and Step 5 of Algorithm 2, the  $(t + 1)$ -th individual is

$$\vec{x}^{t+1} = \begin{cases} \vec{x}^t, & f(\vec{y}^t) \leq f(\vec{x}^t) \\ \vec{y}^t, & f(\vec{y}^t) > f(\vec{x}^t) \end{cases} \tag{9}$$

---

**Algorithm 2** BSO-I

---

- 1: Initialization: Randomly generate an individual  $\vec{x} = (x_1, x_2, \dots, x_n) \in R^n$  based on uniform distribution;
  - 2: **while** stopping criterion is not satisfied **do**
  - 3:    $\vec{y} = \vec{x} + \vec{z}$ , where  $\vec{z}$  is the mutation operator;
  - 4:   **if**  $\vec{y}$  adapts better than  $\vec{x}$  **then**
  - 5:      $\vec{x}$  is substituted for  $\vec{y}$
  - 6:   **end if**
  - 7: **end while**
- Output:**  $\vec{x}$
- 

According to the definition of  $\eta_t$ , where  $t = 0, 1, \dots$ ,

- (1) If  $f(\vec{y}^t) \leq f(\vec{x}^t)$ ,

$$\begin{aligned} \eta_t &= k(x_1^{t+1} + x_2^{t+1} + \dots + x_n^{t+1}) - k(x_1^t + x_2^t + \dots + x_n^t) \\ &= k(x_1^t + x_2^t + \dots + x_n^t) - k(x_1^t + x_2^t + \dots + x_n^t) = 0 \end{aligned} \tag{10}$$

- (2) If  $f(\vec{y}^t) > f(\vec{x}^t)$ ,

$$\begin{aligned} \eta_t &= k(x_1^{t+1} + x_2^{t+1} + \dots + x_n^{t+1}) - k(x_1^t + x_2^t + \dots + x_n^t) \\ &= k(y_1^t + y_2^t + \dots + y_n^t) - k(x_1^t + x_2^t + \dots + x_n^t) \\ &= k[(y_1^t - x_1^t) + (y_2^t - x_2^t) + \dots + (y_n^t - x_n^t)] \\ &= k(z_1^t + z_2^t + \dots + z_n^t) = f(\vec{z}^t) \end{aligned} \tag{11}$$

Since  $z_i \sim N(0, 1)$ ,  $z_1, \dots, z_n$  are independent of each other. All of the  $z_i$  satisfy the additivity of the normal distribution, so  $f(\vec{z}^t)$  obeys the distribution of  $N(0, nk^2)$ .

Hence, the distribution function of  $\eta_t$  is shown as  $F(u) = P(\eta_t \leq u)$ .

- (1) If  $u < 0$ , according to the definition of  $\eta_t$  where  $\eta_t \geq 0$ , it has  $F(u) = 0$ .
- (2) If  $u = 0$ , the probability density function of  $N(0, n)$  is symmetric in the  $y$  axis, so  $F(u) = P(\eta_t \leq u) = P(\eta_t = 0) = \frac{1}{2}$ .
- (3) If  $u > 0$ ,  $F(u) = P(\eta_t \leq u) = \frac{1}{\sqrt{2\pi nk}} \int_{-\infty}^u e^{-\frac{t^2}{2nk^2} t}$ .

Lemma 1 holds. □

Theorem 2 is presented based on the above proof.

**Theorem 2.** If the mutation operator  $\vec{z}$  of BSO-I obeys  $N(0, 1)$ , the upper bound on the expected first hitting time to reach the target fitness value  $na$  is derived as follows.

$$E(T_\epsilon | X_0) \leq 1 + \sqrt{2\pi n} \frac{a}{k} - \sqrt{\frac{2\pi}{n}} \frac{\epsilon}{k}. \tag{12}$$

**Proof.**

$$\begin{aligned} E(X_t - X_{t+1} | X_t) &= E(\eta_t | X_t) = \int_{-\infty}^{+\infty} u F(u) \\ &= \int_0^{+\infty} u \left( \frac{1}{\sqrt{2\pi nk}} \int_{-\infty}^u e^{-\frac{t^2}{2nk^2} t} \right) = \frac{k\sqrt{n}}{\sqrt{2\pi}} \end{aligned} \tag{13}$$

It is assumed that the algorithm starts from the origin at initialization, where  $\vec{x}^0 = (0, 0, \dots, 0)$ , i.e.,

$$X_0 = na - f(\vec{x}^0) = na - k(0 + 0 \dots + 0) = na \tag{14}$$

According to Theorem 1, the upper bound on the expected first hitting time is derived as

$$E(T_\epsilon | X_0) \leq 1 + \int_\epsilon^{na} \frac{\sqrt{2\pi}}{k\sqrt{n}} x = 1 + \sqrt{2\pi n} \frac{a}{k} - \sqrt{\frac{2\pi}{n}} \frac{\epsilon}{k}. \tag{15}$$

Theorem 2 holds.  $\square$

Theorem 2 indicates that for BSO-I, if its mutation operator  $\vec{z}$  obeys  $N(0, 1)$ , the computational time complexity of BSO-I for the equal coefficient linear function is  $E(T_\epsilon | X_0) = O(\sqrt{n})$ .

### 3.1.2. When $z_i \sim U(-\frac{1}{2}, \frac{1}{2})$

The uniform distribution function does not satisfy additivity like the normal distribution function. The Lindeberg–Levy center limit theorem [58] can provide an idea to find the distribution of  $\eta_t$ . The Lindeberg–Levy center limit theorem is introduced below.

Suppose that  $X_n$  is a sequence of independent and identically distributed random variables with  $E(X_i) = \mu$  and  $(X_i) = \sigma^2 > 0$ ; let

$$Y_n^* = \frac{X_1 + X_2 + \dots + X_n - n\mu}{\sigma\sqrt{n}}, \tag{16}$$

then

$$\lim_{n \rightarrow \infty} P(Y_n^* \leq y) = \Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{t^2}{2}} dt \tag{17}$$

is satisfied for any real number  $y$ .

The Lindeberg–Levy center limit theorem [58] shows that if  $n$  is sufficiently large,  $Y_n^* \sim N(0, 1)$ , it has  $\sum_{i=1}^n X_i \sim N(n\mu, n\sigma^2)$ . Generally, the case of higher dimensions requires more attention in the study of the computational time complexity of algorithms. If the mutation operator obeys  $U(-\frac{1}{2}, \frac{1}{2})$ , the distribution function of  $\eta_t$  can be represented by Lemma 2.

**Lemma 2.** For BSO-I, if its mutation operator  $\vec{z}$  obeys  $U(-\frac{1}{2}, \frac{1}{2})$ , the distribution function  $F(u) = P(\eta_t \leq u)$  of the gain  $\eta_t$  is

$$F(u) = \begin{cases} 0, & u < 0 \\ \frac{1}{2}, & u = 0. \\ \frac{\sqrt{6}}{\sqrt{\pi nk}} \int_{-\infty}^u e^{-\frac{6t^2}{nk^2}} dt, & u > 0 \end{cases} \tag{18}$$

**Proof.** According to the definition of  $\eta_t$ , where  $t = 0, 1, \dots$

- (1) If  $f(\vec{y}^t) \leq f(\vec{x}^t)$ ,  $\eta_t = 0$ .
- (2) If  $f(\vec{y}^t) > f(\vec{x}^t)$ ,  $\eta_t = k(z_1^t + z_2^t + \dots + z_n^t)$ , where  $z_i \sim U(-\frac{1}{2}, \frac{1}{2})$ , and  $z_1, \dots, z_n$  are independent of each other. According to the Lindeberg–Levy center limit theorem,  $\eta_t$  obeys  $N(0, \frac{1}{12}nk^2)$ .

Hence, the distribution function of  $\eta_t$  is  $F(u) = P(\eta_t \leq u)$ .

- (1) If  $u < 0$ ,  $F(u) = 0$ .

- (2) If  $u = 0, F(u) = P(\eta_t = 0) = \frac{1}{2}$ .
- (3) If  $u > 0, F(u) = \frac{\sqrt{6}}{\sqrt{\pi nk}} \int_{-\infty}^u e^{-\frac{6t^2}{nk^2} t}$ .

Lemma 2 holds.  $\square$

Theorem 3 is presented based on the above proof.

**Theorem 3.** For BSO-I, if its mutation operator  $\vec{z}$  obeys  $U\left(-\frac{1}{2}, \frac{1}{2}\right)$ , the upper bound on the expected first hitting time to reach the target fitness value  $na$  is derived as follows.

$$E(T_\epsilon | X_0) \leq 1 + 2\sqrt{6\pi n} \frac{a}{k} - \frac{2\sqrt{6\pi} \epsilon}{\sqrt{n} k}. \tag{19}$$

The proof of this theorem is based on the same principle as Theorem 2. The detailed derivation is given as follows.

**Proof.** According to Lemma 2, we have

$$\begin{aligned} E(X_t - X_{t+1} | X_t) &= E(\eta_t | X_t) = \int_{-\infty}^{+\infty} uF(u) \\ &= \int_0^{+\infty} u \left( \frac{\sqrt{6}}{\sqrt{\pi nk}} \int_{-\infty}^u e^{-\frac{6t^2}{nk^2} t} \right) = \frac{k\sqrt{n}}{2\sqrt{6\pi}}, \end{aligned} \tag{20}$$

and the algorithm starts from the origin at initialization,  $\vec{x}^0 = (0, 0, \dots, 0)$ , i.e.,  $X_t = na$ . According to Theorem 1, the upper bound on the expected first hitting time is derived as

$$E(T_\epsilon | X_0) \leq 1 + \int_\epsilon^{na} \frac{2\sqrt{6\pi}}{k\sqrt{n}} x = 1 + 2\sqrt{6\pi n} \frac{a}{k} - \frac{2\sqrt{6\pi} \epsilon}{\sqrt{n} k}. \quad \square \tag{21}$$

Theorem 3 indicates that if the mutation operator  $\vec{z}$  of BSO-I obeys  $U\left(-\frac{1}{2}, \frac{1}{2}\right)$ , its computational time complexity is  $E(T_\epsilon | X_0) = O(\sqrt{n})$  for the equal coefficient linear function.

### 3.1.3. When $z_i \sim U(-1, 1)$

If the mutation operator obeys  $U(-1, 1)$ , the distribution function of  $\eta_t$  can be represented by Lemma 3.

**Lemma 3.** For BSO-I, if its mutation operator  $\vec{z}$  obeys  $U(-1, 1)$ , the distribution function  $F(u) = P(\eta_t \leq u)$  of the gain  $\eta_t$  is

$$F(u) = \begin{cases} 0, & u < 0 \\ \frac{1}{2}, & u = 0 \\ \frac{\sqrt{3}}{\sqrt{2\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{2nk^2} t}, & u > 0 \end{cases}. \tag{22}$$

The proof of this lemma is based on the same principle as Lemma 2.

**Proof.** According to the definition of  $\eta_t, t = 0, 1, \dots$ ,

- (1) If  $f(\vec{y}^t) \leq f(\vec{x}^t), \eta_t = 0$ .
- (2) If  $f(\vec{y}^t) > f(\vec{x}^t), \eta_t = f(\vec{z}^t)$ .

Since  $z_i \sim U(-1, 1), z_1, \dots, z_n$  are independent of each other, according to the Lindeberg-Levy center limit theorem,  $f(\vec{z}^t)$  obeys  $N(0, \frac{1}{3}nk^2)$ .

Hence, the  $\eta_t$  distribution function  $F(u) = P(\eta_t \leq u)$  is

- (1) If  $u < 0, F(u) = 0$ .
- (2) If  $u = 0, F(u) = P(\eta_t = 0) = \frac{1}{2}$ .
- (3) If  $u > 0, F(u) = \frac{\sqrt{3}}{\sqrt{2\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{2nk^2} t}$ .



□

According to Lemma 3 and Theorem 1, Theorem 4 can be inferred.

**Theorem 4.** For BSO-I, if its mutation operator  $\bar{z}$  obeys  $U(-1, 1)$ , the upper bound on the expected first hitting time to reach the target fitness value  $na$  is derived as

$$E(T_\varepsilon|X_0) \leq 1 + \sqrt{6\pi n} \frac{a}{k} - \sqrt{\frac{6\pi}{n}} \frac{\varepsilon}{k}. \tag{23}$$

The proof of this theorem is based on the same principle as Theorem 2. The proof is given as follows.

**Proof.** According to Lemma 3, we have

$$\begin{aligned} E(X_t - X_{t+1}|X_t) &= E(\eta_t|X_t) = \int_{-\infty}^{+\infty} uF(u) \\ &= \int_0^{+\infty} u \left( \frac{\sqrt{3}}{\sqrt{2\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{2nk^2} t} \right) = \frac{k\sqrt{n}}{\sqrt{6\pi}}, \end{aligned} \tag{24}$$

and the algorithm starts from the origin at initialization,  $\bar{x}^0 = (0, 0, \dots, 0)$ , i.e.,  $X_t = na$ . According to Theorem 1, the upper bound on the expected first hitting time is derived as

$$E(T_\varepsilon|X_0) \leq 1 + \int_\varepsilon^{na} \frac{\sqrt{6\pi}}{k\sqrt{n}} x = 1 + \sqrt{6\pi n} \frac{a}{k} - \sqrt{\frac{6\pi}{n}} \frac{\varepsilon}{k}. \quad \square \tag{25}$$

Theorem 4 indicates that if the mutation operator  $\bar{z}$  of BSO-I obeys  $U(-1, 1)$ , its computational time complexity for the equal coefficient linear function is  $E(T_\varepsilon|X_0) = O(\sqrt{n})$ .

The time complexity of BSO-I with three different mutation operators is  $O(\sqrt{n})$ . In the next section, we will discuss the running time of BSO considering the case with a disrupting operation.

### 3.2. Case Study of BSO with Disrupting Operator

Based on the average gain model, this section analyzes the upper bounds of the expected first hit time in three BSO cases. When interference operations are added to the single individual BSO, the algorithm process can be simplified as follows.

In Algorithm 3, the disrupting operations, which are executed with a small probability, are shown in Steps 3 to 6. Let  $A = \{Pb'|Pb' < Pb\}$  indicate that replacement occurs, while  $\bar{A} = \{Pb'|Pb' \geq Pb\}$  indicates that no replacement occurs.

To highlight the effect of each mutation operator on the algorithm, we choose the same mutation operators of BSO-I in Steps 5 and 8 to generate new individuals. As a result,  $b_i = x_i + \Delta x_i$ , where mutation operator parameter  $\Delta x_i$  and  $z_i$  follow the same distribution.

Here,  $\bar{x}^t = (x_1^t, x_2^t, \dots, x_n^t) \in S$  is still the  $t$ -th individual of the algorithm. We have  $X_t = an - f(\bar{x}_t) = an - k(x_1^t + x_2^t, \dots + x_n^t)$ , and the corresponding gain at  $t$  is given by  $\eta_t = X_t - X_{t+1} = k(x_1^{t+1} + x_2^{t+1} + \dots + x_n^{t+1}) - k(x_1^t + x_2^t + \dots + x_n^t)$ .

#### 3.2.1. When $z_i \sim N(0, 1)$

- (1) If  $Pb' \geq Pb$ , it is the same as the result of the case with no disrupting operation in Section 3.1, and the average gain is

$$E(X_t - X_{t+1}|X_t, \bar{A}) = E(\eta_t|X_t, \bar{A}) = \frac{k\sqrt{n}}{\sqrt{2\pi}}. \tag{26}$$

- (2) If  $Pb' < Pb$  and the mutation operator obeys  $N(0, 1)$ , the distribution function of  $\eta_t$  is represented by Lemma 4.

**Algorithm 3** BSO-II

```

1: Initialization: Randomly generate an individual  $\vec{x} = (x_1, x_2, \dots, x_n) \in R^n$  based on
   uniform distribution;
2: while stopping criterion is not satisfied do
3:   Randomly generate a value  $Pb'$  from 0 to 1 based on uniform distribution;
4:   if  $Pb'$  is smaller than a pre-determined probability  $Pb$  then
5:     replace  $\vec{x}$  with a randomly generated individual  $\vec{b} = (b_1, b_2, \dots, b_n)$  based on
     uniform distribution;
6:   end if
7:   if  $Pb' < Pb$  then
8:      $\vec{y} = \vec{b} + \vec{z}$ , where  $\vec{z}$  is a mutation operator;
9:   else
10:     $\vec{y} = \vec{x} + \vec{z}$ ;
11:  end if
12:  if  $\vec{y}$  has better fitness than  $\vec{x}$  then
13:    replace  $\vec{x}$  with  $\vec{y}$ 
14:  end if
15: end while
Output:  $\vec{x}$ 

```

**Lemma 4.** For BSO-II, if its mutation operator  $\vec{z}$  obeys  $N(0, 1)$  and  $Pb' < Pb$ , the distribution function of the gain  $\eta_t$  is  $F(u) = P(\eta_t \leq u)$ .

$$F(u) = \begin{cases} 0, & u < 0 \\ \frac{1}{2}, & u = 0. \\ \frac{1}{2k\sqrt{\pi n}} \int_{-\infty}^u e^{-\frac{t^2}{4nk^2} t}, & u > 0 \end{cases} \tag{27}$$

**Proof.** According to Step 12 and Step 13 of Algorithm 3, the  $(t + 1)$ -th individual is  $\vec{x}^{t+1} = \begin{cases} \vec{x}^t, & f(\vec{y}^t) \leq f(\vec{x}^t) \\ \vec{y}^t, & f(\vec{y}^t) > f(\vec{x}^t) \end{cases}$ .

According to the definition of  $\eta_t, t = 0, 1, \dots$ ,

- (1) If  $f(\vec{y}^t) \leq f(\vec{x}^t), \eta_t = 0$ .
- (2) If  $f(\vec{y}^t) > f(\vec{x}^t),$

$$\begin{aligned} \eta_t &= k(x_1^{t+1} + x_2^{t+1} + \dots + x_n^{t+1}) - k(x_1^t + x_2^t + \dots + x_n^t) \\ &= k[(b_1^t + b_2^t + \dots + b_n^t) + (z_1^t + z_2^t + \dots + z_n^t)] - k(x_1^t + x_2^t + \dots + x_n^t) \\ &= k[(x_1^t + x_2^t + \dots + x_n^t) + (\Delta x_1^t + \Delta x_2^t + \dots + \Delta x_n^t)] \\ &\quad + k(z_1^t + z_2^t + \dots + z_n^t) - k(x_1^t + x_2^t + \dots + x_n^t) \\ &= k[(\Delta x_1^t + \Delta x_2^t + \dots + \Delta x_n^t) + (z_1^t + z_2^t + \dots + z_n^t)] \end{aligned} \tag{28}$$

Since  $z_i \sim N(0, 1), \Delta x_i \sim N(0, 1)$ , and  $z_1, \dots, z_n$  are independent of each other,  $\Delta x_1, \Delta x_2, \dots, \Delta x_n$  are also independent of each other. All of  $z_i$  and  $\Delta x_i$  satisfy the additivity of the normal distribution. As a result,  $\eta_t$  obeys  $N(0, 2nk^2)$ .

Hence, the distribution function of  $\eta_t$  is  $F(u) = P(\eta_t \leq u)$ .

- (1) If  $u < 0, F(u) = 0$ .
- (2) If  $u = 0, F(u) = P(\eta_t \leq u) = P(\eta_t = 0) = \frac{1}{2}$ .
- (3) If  $u > 0, F(u) = P(\eta_t \leq u) = \frac{1}{2k\sqrt{\pi n}} \int_{-\infty}^u e^{-\frac{t^2}{4nk^2} t}$ .

Lemma 4 holds.  $\square$

Based on the above proofs, we can conclude that

$$E(X_t - X_{t+1}|X_t, A) = E(\eta_t|X_t, A) = \int_0^{+\infty} u \left( \frac{1}{2k\sqrt{\pi n}} \int_{-\infty}^u e^{-\frac{t^2}{4nk^2}} t \right) = \frac{k\sqrt{n}}{\sqrt{\pi}} \quad (29)$$

Assume that the probability  $Pb$  is equal to 0.2 [59]; Theorem 5 can be obtained according to Theorem 1.

**Theorem 5.** For BSO-II, if its mutation operator  $\bar{z}$  obeys  $N(0, 1)$ , the upper bound on the expected first hitting time to reach the target fitness value  $na$  is derived as follows.

$$E(T_\varepsilon|X_0) \leq 1 + \frac{5\sqrt{2\pi n} a}{4 + \sqrt{2} k} - \frac{5\sqrt{2\pi}}{4\sqrt{n} + \sqrt{2n}} \frac{\varepsilon}{k}. \quad (30)$$

**Proof.**

$$\begin{aligned} E(X_t - X_{t+1}|X_t) &= P(\bar{A})E(X_t - X_{t+1}|X_t, \bar{A}) + P(A)E(X_t - X_{t+1}|X_t, A) \\ &\text{(full expectation formula)} \\ &= (1 - 0.2) \times E(X_t - X_{t+1}|X_t, \bar{A}) + 0.2 \times E(X_t - X_{t+1}|X_t, A) \quad (31) \\ &= 0.8 \times \frac{k\sqrt{n}}{\sqrt{2\pi}} + 0.2 \times \frac{k\sqrt{n}}{\sqrt{\pi}} = \frac{4k\sqrt{n} + k\sqrt{2n}}{5\sqrt{2\pi}} \end{aligned}$$

The algorithm starts from the origin at initialization,  $\bar{x}^0 = (0, 0, \dots, 0)$ , i.e.,  $X_t = na$ . According to Theorem 1, the upper bound on the expected first hitting time is derived as

$$\begin{aligned} E(T_\varepsilon|X_0) &\leq 1 + \int_\varepsilon^{na} \frac{5\sqrt{2\pi}}{4k\sqrt{n} + k\sqrt{2n}} x \\ &= 1 + \frac{5\sqrt{2\pi n} a}{4 + \sqrt{2} k} - \frac{5\sqrt{2\pi}}{4\sqrt{n} + \sqrt{2n}} \frac{\varepsilon}{k} \quad (32) \end{aligned}$$

Theorem 5 holds.  $\square$

According to the proof of Theorem 5, if the mutation operator  $\bar{z}$  of BSO-II obeys  $N(0, 1)$ , its computational time complexity for the equal coefficient linear function is  $E(T_\varepsilon|X_0) = O(\sqrt{n})$ .

### 3.2.2. When $z_i \sim U\left(-\frac{1}{2}, \frac{1}{2}\right)$

- (1) If  $Pb' \geq Pb$ , the result is the same as the case in Section 3.1 with no disrupting operation. The average gain is

$$E(X_t - X_{t+1}|X_t, \bar{A}) = E(\eta_t|X_t, \bar{A}) = \frac{k\sqrt{n}}{2\sqrt{6\pi}}. \quad (33)$$

- (2) If  $Pb' < Pb$  and the mutation operator obeys  $U\left(-\frac{1}{2}, \frac{1}{2}\right)$ , the distribution function of  $\eta_t$  is represented by Lemma 5.

**Lemma 5.** If the mutation operator of BSO-II  $\bar{z}$  obeys  $U\left(-\frac{1}{2}, \frac{1}{2}\right)$  and  $Pb' < Pb$ , the distribution function of the gain  $\eta_t$  is  $F(u) = P(\eta_t \leq u)$ .

$$F(u) = \begin{cases} 0, & u < 0 \\ \frac{1}{2}, & u = 0. \\ \frac{\sqrt{3}}{\sqrt{\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{nk^2}} t, & u > 0 \end{cases} \quad (34)$$

The proof of this lemma is based on the same principle as Lemma 4. The detailed derivation is given as follows.

**Proof.** According to the definition of  $\eta_t, t = 0, 1, \dots,$

- (1) If  $f(\bar{y}^t) \leq f(\bar{x}^t), \eta_t = 0.$
- (2) If  $f(\bar{y}^t) > f(\bar{x}^t), \eta_t = k[(\Delta x_1^t + \Delta x_2^t + \dots + \Delta x_n^t) + (z_1^t + z_2^t + \dots + z_n^t)]$

Since  $z_i \sim U\left(-\frac{1}{2}, \frac{1}{2}\right), \Delta x_i \sim U\left(-\frac{1}{2}, \frac{1}{2}\right),$  and  $z_1, \dots, z_n$  are independent of each other,  $\Delta x_1, \Delta x_2, \dots, \Delta x_n$  are also independent of each other.  $f(\bar{z}^t)$  obeys  $N\left(0, \frac{1}{6}nk^2\right)$  according to the Lindeberg–Levy center limit theorem.

Hence, the  $\eta_t$  distribution function  $F(u) = P(\eta_t \leq u)$  is

- (1) If  $u < 0, F(u) = 0.$
  - (2) If  $u = 0, F(u) = P(\eta_t \leq u) = P(\eta_t = 0) = \frac{1}{2}.$
  - (3) If  $u > 0, F(u) = P(\eta_t \leq u) = \frac{\sqrt{3}}{\sqrt{\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{nk^2} t}.$
- 

Theorem 6 can be concluded based on Lemma 5 and Theorem 1.

**Theorem 6.** *If the mutation operator of BSO-II  $\bar{z}$  obeys  $U\left(-\frac{1}{2}, \frac{1}{2}\right),$  the upper bound on the expected first hitting time to reach the target fitness value  $na$  is derived as follows.*

$$E(T_\epsilon | X_0) \leq 1 + \frac{10\sqrt{6\pi n} a}{4 + \sqrt{2} k} - \frac{10\sqrt{6\pi} \epsilon}{4\sqrt{n} + \sqrt{2n} k}. \tag{35}$$

The proof of this theorem is based on the same principle as Theorem 5. The detailed derivation is presented as follows.

**Proof.** According to Lemma 5, we have

$$E(X_t - X_{t+1} | X_t, A) = E(\eta_t | X_t, A) = \int_0^{+\infty} u \left( \frac{\sqrt{3}}{\sqrt{\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{nk^2} t} \right) = \frac{k\sqrt{n}}{2\sqrt{3\pi}}. \tag{36}$$

Suppose that the probability  $Pb = 0.2;$  according to Theorem 1, the following conclusions can be drawn:

$$\begin{aligned} & E(X_t - X_{t+1} | X_t) \\ &= P(\bar{A})E(X_t - X_{t+1} | X_t, \bar{A}) + P(A)E(X_t - X_{t+1} | X_t, A) \\ &= (1 - 0.2) \times E(X_t - X_{t+1} | X_t, \bar{A}) + 0.2 \times E(X_t - X_{t+1} | X_t, A) \\ &= 0.8 \times \frac{k\sqrt{n}}{2\sqrt{6\pi}} + 0.2 \times \frac{k\sqrt{n}}{2\sqrt{3\pi}} = \frac{4k\sqrt{n} + k\sqrt{2n}}{10\sqrt{6\pi}} \end{aligned} \tag{37}$$

The algorithm starts from the origin at initialization,  $\bar{x}^0 = (0, 0, \dots, 0),$  i.e.,  $X_t = na.$  According to Theorem 1, the upper bound on the expected first hitting time is derived as

$$E(T_\epsilon | X_0) \leq 1 + \int_\epsilon^{na} \frac{10\sqrt{6\pi}}{4\sqrt{n} + \sqrt{2n}} x = 1 + \frac{10\sqrt{6\pi n} a}{4 + \sqrt{2} k} - \frac{10\sqrt{6\pi} \epsilon}{4\sqrt{n} + \sqrt{2n} k}. \quad \square \tag{38}$$

Theorem 6 indicates that for BSO-II, if its mutation operator  $\bar{z}$  obeys  $U\left(-\frac{1}{2}, \frac{1}{2}\right),$  the computational time complexity of BSO-II for the equal coefficient linear function is  $E(T_\epsilon | X_0) = O(\sqrt{n}).$

### 3.2.3. When $z_i \sim U(-1, 1)$

- (1) If  $Pb' \geq Pb,$  the result is the same as the case in Section 3.1 with no disrupting operation. The average gain is

$$E(X_t - X_{t+1} | X_t, \bar{A}) = E(\eta_t | X_t, \bar{A}) = \frac{k\sqrt{n}}{\sqrt{6\pi}}. \tag{39}$$

- (2) If  $Pb' < Pb$ , and the mutation operator obeys  $U(-1, 1)$ , the distribution function of  $\eta_t$  is represented by Lemma 6.

**Lemma 6.** *If the mutation operator of BSO-II  $\bar{z}$  obeys  $U(-1, 1)$  and  $Pb' < Pb$ , its distribution function  $F(u) = P(\eta_t \leq u)$  of the gain  $\eta_t$  is*

$$F(u) = \begin{cases} 0, & u < 0 \\ \frac{1}{2}, & u = 0. \\ \frac{\sqrt{3}}{\sqrt{4\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{4nk^2} t}, & u > 0 \end{cases} \quad (40)$$

The proof of Lemma 6 is based on the same principle as Lemma 4. The proof is given as follows, which is used to support the proof of Theorem 7.

**Proof.** According to the definition of  $\eta_t, t = 0, 1, \dots$ ,

- (1) If  $f(\bar{y}^t) \leq f(\bar{x}^t), \eta_t = 0$ .
- (2) If  $f(\bar{y}^t) > f(\bar{x}^t), \eta_t = k[(\Delta x_1^t + \Delta x_2^t + \dots + \Delta x_n^t) + (z_1^t + z_2^t + \dots + z_n^t)]$

Since  $z_i \sim U(-1, 1), \Delta x_i \sim U(-1, 1)$ , and  $z_1, \dots, z_n$  are independent of each other,  $\Delta x_1, \Delta x_2, \dots, \Delta x_n$  are also independent of each other.  $f(\bar{z}^t)$  obeys  $N(0, \frac{2}{3}nk^2)$  according to the Lindeberg–Levy center limit theorem.

Hence, the  $\eta_t$  distribution function  $F(u) = P(\eta_t \leq u)$  is

- (1) If  $u < 0, F(u) = 0$ .
  - (2) If  $u = 0, F(u) = P(\eta_t \leq u) = P(\eta_t = 0) = \frac{1}{2}$ .
  - (3) If  $u > 0, F(u) = P(\eta_t \leq u) = \frac{\sqrt{3}}{\sqrt{4\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{4nk^2} t}$ .
- 

**Theorem 7.** *If the mutation operator of BSO-II  $\bar{z}$  obeys  $U(-1, 1)$ , the upper bound on the expected first hitting time to reach the target fitness value  $na$  is derived as*

$$E(T_\epsilon | X_0) \leq 1 + \frac{5\sqrt{6\pi n} a}{4 + \sqrt{2} k} - \frac{5\sqrt{6\pi}}{4\sqrt{n} + \sqrt{2n} k} \frac{\epsilon}{k}. \quad (41)$$

The proof of this theorem is based on the same principle as Theorem 5. The proof is given as follows.

**Proof.** According to Lemma 6, we have

$$E(X_t - X_{t+1} | X_t, A) = E(\eta_t | X_t, A) = \int_0^{+\infty} u \left( \frac{\sqrt{3}}{\sqrt{4\pi nk}} \int_{-\infty}^u e^{-\frac{3t^2}{4nk^2} t} \right) = \frac{k\sqrt{n}}{\sqrt{3\pi}}. \quad (42)$$

Suppose that the probability  $Pb = 0.2$ ; according to Theorem 1, the following conclusions can be drawn:

$$\begin{aligned} & E(X_t - X_{t+1} | X_t) \\ &= P(\bar{A})E(X_t - X_{t+1} | X_t, \bar{A}) + P(A)E(X_t - X_{t+1} | X_t, A) \\ &= (1 - 0.2) \times E(X_t - X_{t+1} | X_t, \bar{A}) + 0.2 \times E(X_t - X_{t+1} | X_t, A) \\ &= 0.8 \times \frac{k\sqrt{n}}{\sqrt{6\pi}} + 0.2 \times \frac{k\sqrt{n}}{\sqrt{3\pi}} = \frac{4k\sqrt{n} + k\sqrt{2n}}{5\sqrt{6\pi}}, \end{aligned} \quad (43)$$

The algorithm starts from the origin at initialization,  $\bar{x}^0 = (0, 0, \dots, 0)$ , i.e.,  $X_t = na$ . According to Theorem 1, the upper bound on the expected first hitting time is derived as

$$E(T_\epsilon | X_0) \leq 1 + \int_\epsilon^{na} \frac{5\sqrt{6\pi}}{4\sqrt{n} + \sqrt{2n} k} x = 1 + \frac{5\sqrt{6\pi n} a}{4 + \sqrt{2} k} - \frac{5\sqrt{6\pi}}{4\sqrt{n} + \sqrt{2n} k} \frac{\epsilon}{k}. \quad \square \quad (44)$$

Theorem 7 indicates that if the mutation operator of BSO-II  $\vec{z}$  obeys  $U(-1, 1)$ , its computational time complexity for the equal coefficient linear function is  $E(T_\epsilon|X_0) = O(\sqrt{n})$ .

### 3.3. Summary

Overall, we summarize the theoretical analysis results of the running-time analysis of the single individual BSO in solving the  $n$ -dimensional equal coefficient linear function in six different situations. The theoretical analysis results are shown in Table 1. The time complexity of the single individual BSO in these six cases is  $O(\sqrt{n})$ . However, the coefficients in the display expressions are different.

**Table 1.** Analysis of the running time of BSO in six different situations.

Algorithm	Mutation Operator $\vec{z}$	Display Expression for $E(T_\epsilon X_0)$	Time Complexity $T(n)$
BSO-I	$N(0, 1)$	$E(T_\epsilon X_0) \leq 1 + \sqrt{2\pi n} \frac{a}{k} - \sqrt{\frac{2\pi}{n}} \frac{\epsilon}{k}$	$O(\sqrt{n})$
	$U\left(-\frac{1}{2}, \frac{1}{2}\right)$	$E(T_\epsilon X_0) \leq 1 + 2\sqrt{6\pi n} \frac{a}{k} - \frac{2\sqrt{6\pi}}{\sqrt{n}} \frac{\epsilon}{k}$	$O(\sqrt{n})$
	$U(-1, 1)$	$E(T_\epsilon X_0) \leq 1 + \sqrt{6\pi n} \frac{a}{k} - \sqrt{\frac{6\pi}{n}} \frac{\epsilon}{k}$	$O(\sqrt{n})$
BSO-II	$N(0, 1)$	$E(T_\epsilon X_0) \leq 1 + \frac{5\sqrt{2\pi n}}{4+\sqrt{2}} \frac{a}{k} - \frac{5\sqrt{2\pi}}{4\sqrt{n}+\sqrt{2n}} \frac{\epsilon}{k}$	$O(\sqrt{n})$
	$U\left(-\frac{1}{2}, \frac{1}{2}\right)$	$E(T_\epsilon X_0) \leq 1 + \frac{10\sqrt{6\pi n}}{4+\sqrt{2}} \frac{a}{k} - \frac{10\sqrt{6\pi}}{4\sqrt{n}+\sqrt{2n}} \frac{\epsilon}{k}$	$O(\sqrt{n})$
	$U(-1, 1)$	$E(T_\epsilon X_0) \leq 1 + \frac{5\sqrt{6\pi n}}{4+\sqrt{2}} \frac{a}{k} - \frac{5\sqrt{6\pi}}{4\sqrt{n}+\sqrt{2n}} \frac{\epsilon}{k}$	$O(\sqrt{n})$

Table 1 shows the correlation between the expected first hitting time, the dimension  $n$ , the slope  $k$ , and the parameter  $a$ . The upper bounds on the expected first hitting time of BSO-II are lower than those of BSO-I. This means that the performance of BSO-II is better than that of BSO-I in solving the equal coefficient linear function. The disrupting operation in BSO helps to reduce the running time of the algorithm. Moreover, the upper bounds on the expected first hitting time of the algorithm using the standard normal distribution mutation operator are lower than those of the algorithms with the uniform distribution mutation operator. In addition, the upper bounds on the expected first hitting time of the algorithm using the  $U\left(-\frac{1}{2}, \frac{1}{2}\right)$  mutation operator are approximately two times higher than those of the algorithm using the  $U(-1, 1)$  mutation operator.

## 4. Experimental Results

In Section 3, we obtain the theoretical analysis results of the expected first hitting time of single individual BSOs through the average gain model. To verify the correctness of the analysis results, numerical experiments are presented in this section.

As the number of samples increases, the arithmetic mean will gradually approach true mathematical expectations based on the Wiener–Khinchine theorem of large numbers [58]. The Wiener–Khinchine theorem of large numbers is introduced as follows.

Suppose that  $X_1, X_2, \dots$  is a sequence of independent and identically distributed random variables with  $E(X_i) = \mu$ ; for any  $\epsilon > 0$ , the following equation will hold.

$$\lim_{n \rightarrow \infty} P\left(\left|\frac{1}{n} \sum_{i=1}^n X_i - \mu\right| < \epsilon\right) = 1. \tag{45}$$

The Wiener–Khinchine theorem of large numbers indicates that if the number of samples is sufficiently large, the mathematical expectations are approximately equal to the mean of samples  $X_1, X_2, \dots, X_n$ . Therefore, we use the arithmetic mean of the first hitting time of multiple experiments to estimate the actual expected first hitting time.

In this section, the parameters of the proposed approach are set as follows. The fixing error is  $\epsilon = 1 \times 10^{-8}$ , the initial individual is  $x^0 = (x_1^0, x_2^0, \dots, x_n^0) = (0, 0, \dots, 0)$ , the slope

is  $k = 1$ , and the target fitness parameter is  $a = 10$ . The problem dimension  $n$  is set from 10 to 280. BSO-I and BSO-II are conducted on the  $n$ -dimensional equal coefficient linear function for 300 runs for each dimension. The termination criterion for each experiment is that the error of the optimal solution should be below a predefined threshold  $\varepsilon$ . Table 2 shows the numerical results of the practical expected first hitting time  $E(\widehat{T_\varepsilon|X_0})$  and the theoretical time upper bound, where  $E(\widehat{T_\varepsilon|X_0}) = \frac{\sum_{i=1}^{300} T_{\varepsilon i}}{300}$ , and  $T_{\varepsilon i}$  is the first hitting time of the  $\varepsilon$ -approximation solution at the  $i$ -th run.

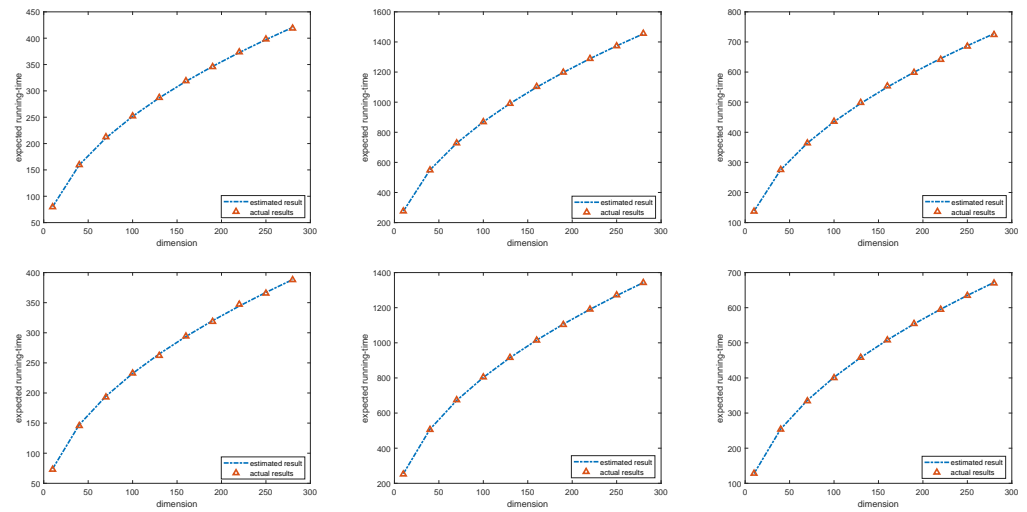
**Table 2.** Comparison of the estimation of the expected first hitting time and the theoretical upper bounds.

Algorithm	$\bar{z}$	$n$	10	40	70	100	130	160	190	220	250	280
BSO-I	$N(0,1)$	$1 + \sqrt{2\pi n} \frac{a}{k} - \sqrt{\frac{2\pi}{n}} \frac{\varepsilon}{k}$	80.27	159.53	210.72	251.66	286.80	318.07	346.51	372.79	397.33	420.44
		$E(\widehat{T_\varepsilon X_0})$	79.65	<b>159.57</b>	<b>212.51</b>	<b>252.09</b>	<b>287.14</b>	316.84	345.61	<b>373.66</b>	<b>398.08</b>	418.75
	$U(-\frac{1}{2}, \frac{1}{2})$	$1 + 2\sqrt{6\pi n} \frac{a}{k} - \frac{2\sqrt{6\pi}}{\sqrt{n}} \frac{\varepsilon}{k}$	275.59	550.17	727.49	869.32	991.04	1099.35	1197.90	1288.93	1373.94	1453.98
		$E(\widehat{T_\varepsilon X_0})$	<b>275.81</b>	548.63	<b>727.97</b>	868.67	990.57	<b>1102.96</b>	<b>1198.00</b>	1288.83	1373.06	<b>1455.95</b>
	$U(-1,1)$	$1 + \sqrt{6\pi n} \frac{a}{k} - \sqrt{\frac{6\pi}{n}} \frac{\varepsilon}{k}$	138.29	275.59	364.24	435.16	496.02	550.17	599.45	644.96	687.47	727.49
		$E(\widehat{T_\varepsilon X_0})$	137.27	275.45	363.95	<b>435.81</b>	<b>498.34</b>	<b>553.30</b>	598.41	641.36	685.46	723.91
BSO-II	$N(0,1)$	$1 + \frac{5\sqrt{2\pi n}}{4+\sqrt{2}} \frac{a}{k} - \frac{5\sqrt{2\pi}}{4\sqrt{n}+\sqrt{2n}} \frac{\varepsilon}{k}$	74.20	147.40	194.68	232.49	264.93	293.81	320.08	344.35	367.01	388.35
		$E(\widehat{T_\varepsilon X_0})$	72.97	145.55	193.03	<b>232.77</b>	262.05	<b>294.26</b>	318.51	<b>347.10</b>	365.51	387.87
	$U(-\frac{1}{2}, \frac{1}{2})$	$1 + \frac{10\sqrt{6\pi n}}{4+\sqrt{2}} \frac{a}{k} - \frac{10\sqrt{6\pi}}{4\sqrt{n}+\sqrt{2n}} \frac{\varepsilon}{k}$	254.58	508.16	671.91	802.89	915.30	1015.32	1106.33	1190.40	1268.90	1342.82
		$E(\widehat{T_\varepsilon X_0})$	251.80	505.49	<b>674.53</b>	<b>804.58</b>	<b>915.61</b>	1014.22	1103.21	<b>1190.82</b>	<b>1271.88</b>	1342.12
	$U(-1,1)$	$1 + \frac{5\sqrt{6\pi n}}{4+\sqrt{2}} \frac{a}{k} - \frac{5\sqrt{6\pi}}{4\sqrt{n}+\sqrt{2n}} \frac{\varepsilon}{k}$	127.79	254.58	336.45	401.95	458.15	508.16	553.67	595.70	634.95	671.91
		$E(\widehat{T_\varepsilon X_0})$	<b>128.10</b>	253.61	334.30	400.20	458.07	507.69	<b>554.34</b>	594.77	634.45	669.75

The points larger than the theoretical upper bounds are highlighted in boldface.

As shown in Table 2, the experimental results strongly fit the theoretical upper bounds, indicating that the error between the theoretical upper bounds and the actual value is within  $\varepsilon$ . The points larger than the theoretical upper bounds are highlighted in boldface. The arithmetic mean of multiple experiments is used to estimate the expected first hitting time. In the real case, only the arithmetic mean of 300 experiments is used to estimate the expected first hitting time, which allows a certain statistical error. According to the central limit theorem, the results obtained from 300 independent experiments follow a normal distribution. The null hypothesis  $H_0$  means that the mean value of the expected first hitting time in 300 experiments is less than or equal to the corresponding theoretical upper bounds. The corresponding significance level  $\alpha$  is 0.05 with the  $T$  testing. Moreover, as shown in Figure 1, the actual expected running time is followed with the estimated result based on our proof. All the detailed results are shown in Table 3.

Table 3 provides the numerical results, where  $h$  represents the hypothetical result,  $p$  represents the  $p$ -value of the test, and  $ci$  is the confidence interval. As shown in the  $T$  testing of Table 3,  $h = 0$  and  $p > \alpha$ . The null hypothesis  $H_0$  is accepted at the significance level  $\alpha = 0.05$ . Therefore, the analytic expression of the running time of BSO obtained based on the average gain model can characterize the actual upper bounds of the running time of BSO in these six BSO variants.



**Figure 1.** The curve of the expected and the actual running time. The six figures, arranged from top to bottom and left to right, depict the three distributions corresponding to BSO-I and the three distributions corresponding to BSO-II, respectively.

**Table 3.** Statistical results of hypothesis testing.

Algorithm	$\vec{z}$	$n$	10	40	70	100	130	160	190	220	250	280	
BSO-I	$N(0,1)$	$h$	0	0	0	0	0	0	0	0	0	0	
		$p$	0.80	0.48	0.06	0.38	0.41	0.81	0.72	0.31	0.34	0.83	
		$ci$	78.41	157.86	210.57	249.84	284.76	314.52	343.03	370.82	395.13	415.87	
	$U(-\frac{1}{2}, \frac{1}{2})$	$h$	0	0	0	0	0	0	0	0	0	0	
		$p$	0.44	0.79	0.42	0.60	0.57	0.10	0.49	0.51	0.61	0.27	
		$ci$	273.43	545.49	724.19	864.51	986.22	1098.22	1193.50	1283.96	1367.75	1450.62	
	$U(-1,1)$	$h$	0	0	0	0	0	0	0	0	0	0	
		$p$	0.85	0.54	0.57	0.36	0.10	0.07	0.69	0.95	0.81	0.94	
		$ci$	135.64	273.05	361.29	432.88	495.38	549.76	595.05	637.79	681.73	720.20	
	BSO-II	$N(0,1)$	$h$	0	0	0	0	0	0	0	0	0	0
			$p$	0.95	0.96	0.93	0.42	0.99	0.38	0.83	0.06	0.81	0.61
			$ci$	71.71	143.86	191.16	230.52	259.89	291.83	315.84	344.26	362.73	385.06
$U(-\frac{1}{2}, \frac{1}{2})$		$h$	0	0	0	0	0	0	0	0	0	0	
		$p$	0.98	0.92	0.12	0.23	0.45	0.65	0.86	0.44	0.18	0.59	
		$ci$	249.47	502.38	670.91	800.79	911.60	1009.53	1098.54	1185.87	1266.44	1337.20	
$U(-1,1)$		$h$	0	0	0	0	0	0	0	0	0	0	
		$p$	0.37	0.77	0.92	0.84	0.52	0.60	0.37	0.66	0.59	0.83	
		$ci$	126.53	251.47	331.78	397.34	455.24	504.80	551.07	591.02	630.94	665.97	



## 5. Conclusions

The running time of six BSO variants for the equal coefficient linear function is analyzed in this paper based on the average gain model. The additivity of the normal distribution and the Lindeberg–Levy center limit theorem are applied to deal with the superposition of the normal distribution mutation operator and the uniform distribution mutation operator, respectively. Furthermore, the full expectation formula is utilized to deal with the problem of individual replacement with a certain probability in the disrupting operator. This paper also concludes the upper bounds on the expected first hitting time of the single individual BSO in equal coefficient linear functions.

The analysis results show that the time complexity of BSO-I and BSO-II is  $O(\sqrt{n})$  in the equal coefficient linear function. However, their coefficients are different. In the linear function with equal coefficients, the upper bound of the expected first hit time of BSO-II is smaller than that of BSO-I. In addition, the single individual BSO using the standard normally distributed mutation operator expects a lower upper bound on the first hit time than the corresponding algorithm using the uniformly distributed mutation operator. The upper bounds on the expected first hitting time of single individual BSOs with the  $U(-\frac{1}{2}, \frac{1}{2})$  mutation operator are approximately twice those of BSOs with the  $U(-1, 1)$  mutation operator.

In our future work, we will analyze the running time of the population-based BSO in the equal coefficient linear function. The running time of population-based BSOs in practical optimization problems is also an important topic. Moreover, it is crucial to extend our research to practical optimization problems that are encountered in real-world applications with complex constraints, non-linear relationships, or high-dimensional spaces.

**Author Contributions:** Conceptualization, G.M. and F.L.; methodology, G.M., F.L. and Y.H.; formal analysis, G.M. and Y.H.; investigation, D.L., J.S., X.Y. and H.H.; writing original draft preparation, F.L., D.L. and J.S.; writing review and editing, G.M. and Y.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China (62276103), Natural Science Foundation of Guangdong Province (2022A1515110058, 2022A1515011551), Guangdong Provincial Department of Education (No.2021KTSCX070,2021KCXTD038,2023KCXTD002), Guangzhou Science and Technology Planning Project(2023A04J1684), Doctor Starting Fund of Hanshan Normal University, China (No. QD20190628,QD2021201), Scientific Research Talents Fund of Hanshan Normal University, China (No. Z19113),The quality of teaching construction project of Hanshan Normal University (E22022,E23045), Research Platform Project of Hanshan Normal University (PNB221102), The quality of teaching construction project of Guangdong Provincial Department of Education in 2023 (No.350) and Guangdong Provincial Key Laboratory of Data Science and Intelligent Education (2022KSYS003).

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

**Data Availability Statement:** Data are contained within the article.

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

## References

1. Michaloglou, A.; Tsitsas, N.L. A Brain Storm and Chaotic Accelerated Particle Swarm Optimization Hybridization. *Algorithms* **2023**, *16*, 208. [[CrossRef](#)]
2. Slowik, A.; Kwasnicka, H. Nature inspired methods and their industry applications-swarm intelligence algorithms. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1004–1015. [[CrossRef](#)]
3. Xue, Y.; Jiang, J.; Zhao, B.; Ma, T. A self-adaptive artificial bee colony algorithm based on global best for global optimization. *Soft Comput.* **2018**, *22*, 2935–2952. [[CrossRef](#)]
4. Liu, X.; Zhan, Z.; Gao, Y.; Zhang, J.; Kwong, S.; Zhang, J. Coevolutionary Particle Swarm Optimization With Bottleneck Objective Learning Strategy for Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **2019**, *23*, 587–602. [[CrossRef](#)]
5. Zhao, Q.; Li, C. Two-Stage Multi-Swarm Particle Swarm Optimizer for Unconstrained and Constrained Global Optimization. *IEEE Access* **2020**, *8*, 124905–124927. [[CrossRef](#)]

6. Yu, X.; Chen, W.; Gu, T.; Yuan, H.; Zhang, H.; Zhang, J. ACO-A\*: Ant Colony Optimization Plus A\* for 3-D Traveling in Environments With Dense Obstacles. *IEEE Trans. Evol. Comput.* **2019**, *23*, 617–631. [[CrossRef](#)]
7. Lyu, Z.; Wang, Z.; Duan, D.; Lin, L.; Li, J.; Yang, Y.; Chen, Y.; Li, Y. Tilting Path Optimization of Tilt Quad Rotor in Conversion Process Based on Ant Colony Optimization Algorithm. *IEEE Access* **2020**, *8*, 140777–140791. [[CrossRef](#)]
8. Zhang, L.; Wang, S.; Zhang, K.; Zhang, X.; Sun, Z.; Zhang, H.; Chipecane, M.T.; Yao, J. Cooperative Artificial Bee Colony Algorithm With Multiple Populations for Interval Multiobjective Optimization Problems. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1052–1065. [[CrossRef](#)]
9. Kumar, D.; Mishra, K. Co-variance guided artificial bee colony. *Appl. Soft Comput.* **2018**, *70*, 86–107. [[CrossRef](#)]
10. Cheng, S.; Qin, Q.; Chen, J.; Shi, Y. Brain storm optimization algorithm: A review. *Artif. Intell. Rev.* **2016**, *46*, 445–458. [[CrossRef](#)]
11. Cheng, S.; Sun, Y.; Chen, J.; Qin, Q.; Chu, X.; Lei, X.; Shi, Y. A comprehensive survey of brain storm optimization algorithms. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastián, Spain, 5–8 June 2017; pp. 1637–1644.
12. Cheng, S.; Lei, X.; Hui, L.; Zhang, Y.; Shi, Y. Generalized pigeon-inspired optimization algorithms. *Sci. China Inf. Sci.* **2019**, *62*, 120–130. [[CrossRef](#)]
13. Xiong, G.; Shi, D.; Zhang, J.; Zhang, Y. A binary coded brain storm optimization for fault section diagnosis of power systems. *Electr. Power Syst. Res.* **2018**, *163*, 441–451. [[CrossRef](#)]
14. Wang, Z.; He, J.; Xu, Y.; Crossley, P.; Zhang, D. Multi-objective optimisation method of power grid partitioning for wide-area backup protection. *IET Gener. Transm. Distrib.* **2018**, *12*, 696–703. [[CrossRef](#)]
15. Ogawa, S.; Mori, H. A Hierarchical Scheme for Voltage and Reactive Power Control with Predator-Prey Brain Storm Optimization. In Proceedings of the 2019 20th International Conference on Intelligent System Application to Power Systems (ISAP), New Delhi, India, 10–14 December 2019; pp. 1–6. [[CrossRef](#)]
16. Matsumoto, K.; Fukuyama, Y. Voltage and Reactive Power Control by Parallel Modified Brain Storm Optimization. In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Fukuoka, Japan, 19–21 February 2020; pp. 553–558. [[CrossRef](#)]
17. Soyinka, O.K.; Duan, H. Optimal Impulsive Thrust Trajectories for Satellite Formation via Improved Brainstorm Optimization. In *Proceedings of the Advances in Swarm Intelligence*; Tan, Y., Shi, Y., Niu, B., Eds.; Springer: Cham, Switzerland, 2016; pp. 491–499.
18. Li, J.; Duan, H. Simplified brain storm optimization approach to control parameter optimization in F/A-18 automatic carrier landing system. *Aerosp. Sci. Technol.* **2015**, *42*, 187–195. [[CrossRef](#)]
19. Zhang, C.; Xu, X.; Shi, Y.; Deng, Y.; Li, C.; Duan, H. Binocular Pose Estimation for UAV Autonomous Aerial Refueling via Brain Storm Optimization. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 254–261. [[CrossRef](#)]
20. Tuba, E.; Strumberger, I.; Zivkovic, D.; Bacanin, N.; Tuba, M. Mobile Robot Path Planning by Improved Brain Storm Optimization Algorithm. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [[CrossRef](#)]
21. Li, G.; Zhang, D.; Shi, Y. An Unknown Environment Exploration Strategy for Swarm Robotics Based on Brain Storm Optimization Algorithm. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 1044–1051. [[CrossRef](#)]
22. Aldhafeeri, A.; Rahmat-Samii, Y. Brain Storm Optimization for Electromagnetic Applications: Continuous and Discrete. *IEEE Trans. Antennas Propag.* **2019**, *67*, 2710–2722. [[CrossRef](#)]
23. Sun, Y. A Hybrid Approach by Integrating Brain Storm Optimization Algorithm with Grey Neural Network for Stock Index Forecasting. *Abstr. Appl. Anal.* **2014**, *2014*, 1–10. [[CrossRef](#)]
24. Xiong, G.; Shi, D. Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects. *Energy* **2018**, *157*, 424–435. [[CrossRef](#)]
25. Wu, Y.; Wang, X.; Xu, Y.; Fu, Y. Multi-objective Differential-Based Brain Storm Optimization for Environmental Economic Dispatch Problem. In *Proceedings of the Brain Storm Optimization Algorithms: Concepts, Principles and Applications*; Cheng, S., Shi, Y., Eds.; Springer: Cham, Switzerland, 2019; pp. 79–104.
26. Ma, X.; Jin, Y.; Dong, Q. A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting. *Appl. Soft Comput.* **2017**, *54*, 296–312. [[CrossRef](#)]
27. Liang, J.J.; Wang, P.; Yue, C.T.; Yu, K.; Li, Z.H.; Qu, B. Multi-objective Brainstorm Optimization Algorithm for Sparse Optimization. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [[CrossRef](#)]
28. Fu, Y.; Tian, G.; Fathollahi-Fard, A.M.; Ahmadi, A.; Zhang, C. Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint. *J. Clean. Prod.* **2019**, *226*, 515–525. [[CrossRef](#)]
29. Pourpanah, F.; Shi, Y.; Lim, C.P.; Hao, Q.; Tan, C.J. Feature selection based on brain storm optimization for data classification. *Appl. Soft Comput.* **2019**, *80*, 761–775. [[CrossRef](#)]
30. Peng, S.; Wang, H.; Yu, Q. Multi-Clusters Adaptive Brain Storm Optimization Algorithm for QoS-Aware Service Composition. *IEEE Access* **2020**, *8*, 48822–48835. [[CrossRef](#)]
31. Zhou, Z.; Duan, H.; Shi, Y. Convergence analysis of brain storm optimization algorithm. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 3747–3752. [[CrossRef](#)]

32. Qiao, Y.; Huang, Y.; Gao, Y. The Global Convergence Analysis of Brain Storm Optimization. *NeuroQuantology* **2018**, *16*, 6. [[CrossRef](#)]
33. Zhang, Y.; Huang, H.; Hongyue, W.; Hao, Z. Theoretical analysis of the convergence property of a basic pigeon-inspired optimizer in a continuous search space. *Sci. China Inf. Sci.* **2019**, *62*, 86–94. [[CrossRef](#)]
34. Sudholt, Dirk. A New Method for Lower Bounds on the Running Time of Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **2013**, *17*, 418–435. [[CrossRef](#)]
35. He, J.; Yao, X. Average Drift Analysis and Population Scalability. *IEEE Trans. Evol. Comput.* **2017**, *21*, 426–439. [[CrossRef](#)]
36. Yu, Y.; Qian, C.; Zhou, Z.H. Switch Analysis for Running Time Analysis of Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **2015**, *19*, 777–792. [[CrossRef](#)]
37. Yuanxiang, LI.; Zhenglong, XIANG.; Daomin, JI. Wave models and dynamical analysis of evolutionary algorithms. *Sci. China Inf. Sci.* **2019**, *62*, 53–68.
38. Lehre, P.K.; Witt, C. Concentrated Hitting Times of Randomized Search Heuristics with Variable Drift. In *Proceedings of the Algorithms and Computation*; Ahn, H.K., Shin, C.S., Eds.; Springer: Cham, Switzerland, 2014; pp. 686–697.
39. Witt, C. Fitness levels with tail bounds for the analysis of randomized search heuristics. *Inf. Process. Lett.* **2014**, *114*, 38–41. [[CrossRef](#)]
40. Yu, Y.; Qian, C. Running time analysis: Convergence-based analysis reduces to switch analysis. In Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015; pp. 2603–2610. [[CrossRef](#)]
41. Huang, H.; Xu, W.; Zhang, Y.; Lin, Z.; Hao, Z. Runtime analysis for continuous (1+1) evolutionary algorithm based on average gain model. *Sci. China Inf. Sci.* **2014**, *44*, 811–824.
42. Huang, H.; Su, J.; Zhang, Y.; Hao, Z. An Experimental Method to Estimate Running Time of Evolutionary Algorithms for Continuous Optimization. *IEEE Trans. Evol. Comput.* **2020**, *24*, 275–289. [[CrossRef](#)]
43. Zhang, Y.; Huang, H.; Hao, Z.; Hu, G. First hitting time analysis of continuous evolutionary algorithms based on average gain. *Clust. Comput.* **2016**, *19*, 1323–1332.
44. Yu, Y.; Zhou, Z.H. A new approach to estimating the expected first hitting time of evolutionary algorithms. *Artif. Intell.* **2006**, *172*, 1809–1832. [[CrossRef](#)]
45. Wang, Y. Application of data mining based on swarm intelligence algorithm in financial support of livestock and poultry breeding insurance. *Soft Comput.* **2023**. [[CrossRef](#)]
46. Zhan, Z.; Zhang, J.; Shi, Y.; Liu, H. A modified brain storm optimization. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–8.
47. El-Abd, M. Brain storm optimization algorithm with re-initialized ideas and adaptive step size. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 2682–2686.
48. Shi, Y. Brain Storm Optimization Algorithm. In *Proceedings of the Advances in Swarm Intelligence*; Tan, Y., Shi, Y., Chai, Y., Wang, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 303–309.
49. Shi, Y. An Optimization Algorithm Based on Brainstorming Process. *Int. J. Swarm Intell. Res.* **2011**, *2*, 35–62. [[CrossRef](#)]
50. Yao, X.; Xu, Y. Recent advances in evolutionary computation. *J. Comput. Sci. Technol.* **2006**, *21*, 1–18. [[CrossRef](#)]
51. Agapie, A.; Agapie, M.; Zbaganu, G. Evolutionary algorithms for continuous-space optimisation. *Int. J. Syst. Sci.* **2013**, *44*, 502–512. [[CrossRef](#)]
52. He, J.; Yao, X. Drift analysis and average time complexity of evolutionary algorithms. *Artif. Intell.* **2001**, *127*, 57–85. [[CrossRef](#)]
53. Hassler, U. Riemann Integrals. In *Proceedings of the Stochastic Processes and Calculus: An Elementary Introduction with Applications*; Springer: Cham, Switzerland, 2016; pp. 179–197.
54. Jägersküpfer, J. Combining Markov-chain analysis and drift analysis: The (1+1) evolutionary algorithm on linear functions reloaded. *Algorithmica* **2011**, *59*, 409–424. [[CrossRef](#)]
55. Witt, C. Tight Bounds on the Optimization Time of a Randomized Search Heuristic on Linear Functions. *Comb. Probab. Comput.* **2013**, *22*, 294–318. [[CrossRef](#)]
56. Hao, Z.; Huang, H.; Zhang, X.; Tu, K. A Time Complexity Analysis of ACO for Linear Functions. In Proceedings of the Simulated Evolution and Learning, 6th International Conference, SEAL 2006, Hefei, China, 15–18 October 2006.
57. Jägersküpfer, J. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theor. Comput. Sci.* **2007**, *379*, 329–347. [[CrossRef](#)]
58. Feller, W. *An Introduction to Probability Theory and Its Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2008; Volume 2.
59. Zhan, Z.h.; Chen, W.n.; Lin, Y.; Gong, Y.j.; Li, Y.l.; Zhang, J. Parameter investigation in brain storm optimization. In Proceedings of the 2013 IEEE Symposium on Swarm Intelligence (SIS), Singapore, 16–19 April 2013; pp. 103–110. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.