

## Article

# An Optimization Algorithm Inspired by the Phase Transition Phenomenon for Global Optimization Problems with Continuous Variables

Zijian Cao <sup>1,2</sup> and Lei Wang <sup>1,\*</sup>

<sup>1</sup> School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China; bosscao@163.com or caozijian@xatu.edu.cn

<sup>2</sup> School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China

\* Correspondence: wangleeei@163.com

Received: 8 September 2017; Accepted: 9 October 2017; Published: 20 October 2017

**Abstract:** In this paper, we propose a novel nature-inspired meta-heuristic algorithm for continuous global optimization, named the phase transition-based optimization algorithm (PTBO). It mimics three completely different kinds of motion characteristics of elements in three different phases, which are the unstable phase, the meta-stable phase, and the stable phase. Three corresponding operators, which are the stochastic operator of the unstable phase, the shrinkage operator in the meta-stable phase, and the vibration operator of the stable phase, are designed in the proposed algorithm. In PTBO, the three different phases of elements dynamically execute different search tasks according to their phase in each generation. It makes it such that PTBO not only has a wide range of exploration capabilities, but also has the ability to quickly exploit them. Numerical experiments are carried out on twenty-eight functions of the CEC 2013 benchmark suite. The simulation results demonstrate its better performance compared with that of other state-of-the-art optimization algorithms.

**Keywords:** phase transition; nature-inspired; continuous optimization; global optimization

## 1. Introduction

Nature-inspired optimization algorithms have become of increasing interest to many researchers in optimization fields in recent years [1]. After half a century of development, nature-inspired optimization algorithms have formed a great family. It not only has a wide range of contact with biological, physical, and other basic science, but also involves many fields such as artificial intelligence, artificial life, and computer science.

Due to the differences in natural phenomena, these optimization algorithms can be roughly divided into three types. These are algorithms based on biological evolution, algorithms based on swarm behavior, and algorithms based on physical phenomena. Typical biological evolutionary algorithms are Evolutionary Strategies (ES) [2], Evolutionary Programming (EP) [3], the Genetic Algorithm (GA) [4–6], Genetic Programming (GP) [7], Differential Evolution (DE) [8], the Backtracking Search Algorithm (BSA) [9], Biogeography-Based Optimization (BBO) [10,11], and the Differential Search Algorithm (DSA) [12].

In the last decade, swarm intelligence, as a branch of intelligent computation models, has been gradually rising [13]. Swarm intelligence algorithms mainly simulate biological habits or behavior, including foraging behavior, search behavior and migratory behavior, brooding behavior, and mating behavior. Inspired by these phenomena, researchers have designed many intelligent algorithms, such as Ant Colony Optimization (ACO) [14], Particle Swarm Optimization (PSO) [15], Bacterial Foraging (BFA) [16], Artificial Bee Colony (ABC) [17–19], Group Search Optimization (GSO) [20],

Cuckoo Search (CS) [21,22], Seeker Optimization (SOA) [23], the Bat Algorithm (BA) [24], Bird Mating Optimization (BMO) [25], Brain Storm Optimization [26,27] and the Grey Wolf Optimizer (GWO) [28].

In recent years, in addition to the above two kinds of algorithms, intelligent algorithms simulating physical phenomenon have also attracted a great deal of researchers' attention, such as the Gravitational Search Algorithm (GSA) [29], the Harmony Search Algorithm (HSA) [30], the Water Cycle Algorithm (WCA) [31], the Intelligent Water Drops Algorithm (IWDA) [32], Water Wave Optimization (WWO) [33], and States of Matter Search (SMS) [34]. The classical optimization algorithm about a physical phenomenon is simulated annealing, which is based on the annealing process of metal [35,36].

Though these meta-heuristic algorithms that have many advantages over traditional algorithms, especially in NP hard problems, have been proposed to tackle many challenging complex optimization problems in science and industry, there is no single approach that is optimal for all optimization problems [37]. In other words, an approach may be suitable for solving these problems, but it is not suitable for solving those problems. This is especially true as global optimization problems have become more and more complex, from simple uni-modal functions to hybrid rotated shifted multimodal functions [38]. Hence, more innovative and effective optimization algorithms are always needed.

A natural phenomenon provides a rich source of inspiration to researchers to develop a diverse range of optimization algorithms with different degrees of usefulness and practicability. In this paper, a new meta-heuristic optimization algorithm inspired by the phase transition phenomenon of elements in a natural system for continuous global optimization is proposed, which is termed phase transition-based optimization (PTBO). From a statistical mechanics point of view, a phase transition is a non-trivial macroscopic form of collective behavior in a system composed of a number of elements that follow simple microscopic laws [39]. Phase transitions play an important role in the probabilistic analysis of combinatorial optimization problems, and are successfully applied in Random Graph, Satisfiability, and Traveling Salesman Problems [40]. However, there are few algorithms with the mechanism of phase transition for solving continuous global optimization problems in the literature.

Phase transition is ubiquitous in nature or in social life. For example, at atmospheric pressure, water boils at a critical temperature of 100 °C. When the temperature is lower than 100 °C, water is a liquid, while above 100 °C, it is a gas. Besides this, there are also a lot of examples of phase transition. Everybody knows that magnets attract nails made out of iron. However, the attraction force disappears when the temperature of the nail is raised above 770 °C. When the temperature is above 770 °C, the nail enters a paramagnetic phase. Moreover, it is well-known that mercury is a good conductor for its weak resistance. Nevertheless, the electrical resistance of mercury falls abruptly down to zero when the temperature passes through 4.2 K (kelvin temperature). From a macroscopical view, a phase transition is the transformation of a system from one phase to another one, depending on the values of control parameters, such as temperature, pressure, and other outside interference.

From the above examples, we can see that each system or matter has different phases and related critical points for those phases. We can also observe that all phase transitions have a common law, and we may think that phase transition is the competitive result of two kinds of tendencies: stable order and unstable disorder. In complex system theory, phase transition is related to the self-organized process by which a system transforms from disorder to order. From this point of view, phase transition implicates a search process of optimization. The thermal motion of an element is a source of disorder [41]. The interaction of elements is the cause of order.

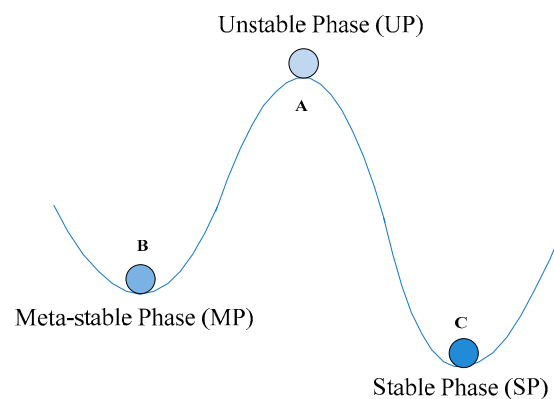
In the proposed PTBO algorithm, the degree of order or disorder is described by stability. We used the value of an objective function to depict the level of stability. For the sake of generality, we extract three kinds of phases in the process of transition from disorder to order, i.e., an unstable phase, a meta-stable phase [42] and a stable phase. From a microscopic viewpoint, the diverse motion characteristics of elements in different phases provide us with novel inspiration to develop a new meta-heuristic algorithm for solving complex continuous optimization problems.

This paper is organized as follows. In Section 2, we briefly introduce the prerequisite preparation for the phase transition-based optimization algorithm. In Section 3, the phase transition-based optimization algorithm and its operators are described. An analysis of PTBO and a comparative study are presented in Section 4. In Section 5, the experimental results with those of other state-of-the-art algorithms are demonstrated. Finally, Section 6 draws conclusions.

## 2. Prerequisite Preparation

### 2.1. Fundamental Concepts

Nature itself is a large complex system. In a natural system, the motion of elements transforming from an unstable phase (disorder) to a stable phase (order) is an eternal natural law. In this paper, we uniformly call a molecule in water, an atom in iron, an electron in mercury, etc., an element. The phase of an element in a system may be divided into an unstable phase, a meta-stable phase, and a stable phase. Figure 1 shows three possible positions of elements in a system.



**Figure 1.** Three possible positions of elements in a system.

From Figure 1, we can observe that an element is most unstable in the position of point A, and we intuitively call it an unstable phase. On the contrary, in the position of point C, the element is most stable, and we name it a stable phase. The position of point B, between point A and point C, is the transition phase, and we term it a meta-stable phase. The definitions of the three phases are as follows.

**Definition 1.** *Unstable Phase (UP).* The element is in a phase of complete disorder, and moves freely towards an arbitrary direction. In the case of this phase, the element has a large range of motion, and has the ability of global divergence.

**Definition 2.** *Meta-stable Phase (MP).* The element is in a phase between disorder and order, and moves according to a certain law, such as towards the lowest point. In the case of this phase, the element has a moderate activity, and possesses the ability of local shrinkage.

**Definition 3.** *Stable Phase (SP).* The element is in a very regular phase of orderly motion. In the case of this phase, the element has a very small range of activity and has the ability of fine tuning.

According to the above definitions, we can give a more detailed description and examples about the characteristics of the three phases. These motion characteristics in three phases provide us with rich potential to develop the proposed PTBO algorithm. The following Table 1 summarizes the motional characteristics of the unstable phase, the meta-stable phase, and the stable phase.

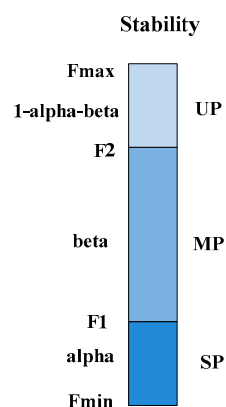
**Table 1.** The motional characteristics of elements in the three phases.

Phase	Main Property	Motion Tendency	Example
UP	disorder, moving towards an arbitrary direction	stochastic motion	gaseous molecule such as water, atom, or electron in normal temperature
MP	between disorder and order, moving according to a certain law	shrinkage motion	liquid molecule such as water or rapid freezing alloy crystal
SP	order, moving in a very regular mode	vibration motion	solid molecule such as water or an atom of paramagnetic phase in a nail

## 2.2. The Determination of Critical Interval about Three Phases

As mentioned before, we use stability, which is depicted by the fitness value of an objective function, to describe the degree of order or disorder of an element. In our proposed algorithm, the higher the fitness value of an element is, the worse the stability is. The question of how to divide the critical intervals of the unstable phase, the meta-stable phase, and the stable phase is a primary problem that must be addressed before the proposed algorithm is designed.

For simplicity, we use  $F_{\max}$  to denote a maximum fitness value, and we say this element is in the most unstable phase. On the contrary,  $F_{\min}$  denotes a minimum fitness value. Then, we set up that the stable phase accounts for  $\alpha$ , and the proportion of the meta-stable phase is  $\beta$ . So, the proportion of the unstable phase is  $1 - \alpha - \beta$ . The ratio of three critical intervals is shown in Figure 2.

**Figure 2.** The critical intervals of the three phases.

Although  $F_{\max}$  and  $F_{\min}$  are dynamically changed in each generation, which means the iteration process of phase transition, the basic relationship between the phase of the elements and the critical intervals is shown in Table 2.

**Table 2.** The relationship between the intervals and the three phases.

Phase	Interval
UP	$[F_{\max}, F2]$
MP	$(F2, F1)$
SP	$[F1, F_{\min}]$

### 3. Phase Transition-Based Optimization Algorithm

#### 3.1. Basic Idea of the PTBO Algorithm

In this work, the motion of elements from an unstable phase to another relative stable phase in PTBO is as natural selection in GA. Many of these iterations from an unstable phase to another relative stable phase can eventually make an element reach absolute stability. The diverse motional characteristics of elements in the three phases are the core of the PTBO algorithm to simulate this phase transition process of elements. In the PTBO algorithm, three corresponding operators are designed. An appropriate combination of the three operators makes an effective search for the global minimum in the solution space.

#### 3.2. The Correspondence of PTBO and the Phase Transition Process

Based on the basic law of elements transitioning from an unstable phase (disorder) to a stable phase (order), the correspondence of PTBO and phase transition can be summarized in Table 3.

**Table 3.** The correspondence of phase transition-based optimization (PTBO) algorithm and phase transition.

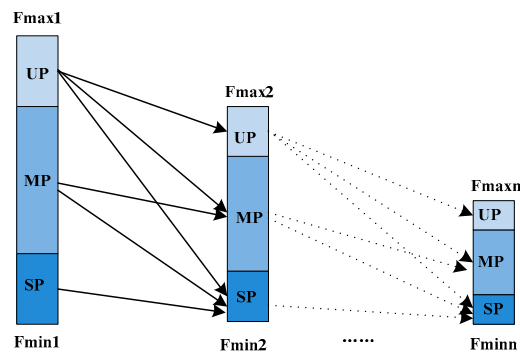
PTBO Algorithm	Phase Transition Process
Individual	Element
Population size	The number of elements
Fitness function	Stability degree of an element
Global optimal solution	The lowest stability degree of an element
Stochastic operator	The stochastic motion of an element in UP
Shrinkage operator	The shrinkage motion of an element in a MP
Vibration operator	The vibration and fine tuning of an element in SP

#### 3.3. The Overall Design of the PTBO Algorithm

The simplified cyclic diagram of the phase transition process in our PTBO algorithm is shown in Figure 3. It is a complete cyclic process of phase transition from an unstable phase to a stable phase. Firstly, in the first generation, we calculate the maximum and minimum fitness value for each element, respectively, and divide the critical intervals of the unstable phase, the meta-stable phase, and the stable phase according to the rules in Table 2. Secondly, the element will perform the relevant search according to its own phase. If the result of the new degree of stability is better than that of the original phase, the motion will be reserved. Otherwise, we will abandon this operation. That is to say, if the original phase of an element is UP, the movement direction may be towards UP, MP, and SP. However, if the original phase of an element is MP, the movement direction may be towards MP and SP. Of course, if the original phase of an element is SP, the movement direction is towards only SP. Finally, after much iteration, elements will eventually obtain an absolute stability.

Broadly speaking, we may think of PTBO as an algorithmic framework. We simply define the general operations of the whole algorithm about the motion of elements in the phase transition. In a word, PTBO is flexible for skilled users to customize it according to a specific scene of phase transition.

According to the above complete cyclic process of the phase transition, the whole operating process of PTBO can be summarized as three procedures: population initialization, iterations of three operators, and individual selection. The three operators in the iterations include the stochastic operator, the shrinkage operator, and the vibration operator.



**Figure 3.** A phase transition from an unstable phase to a stable phase.

### 3.3.1. Population Initialization

PTBO is also a population-based meta-heuristic algorithm. Like other evolutionary algorithms (EAs), PTBO starts with an initialization of a population, which contains a population size (the size is  $N$ ) of element individuals. The current generation evolves into the next generation through the three operators described as below (see Section 3.3.2). That is to say, the population continually evolves along with the proceeding generation until the termination condition is met. Here, we initialize the  $j$ -th dimensional component of the  $i$ -th individual as

$$X_{ij} = X_{j\min} + rand * (X_{j\max} - X_{j\min}) \quad (1)$$

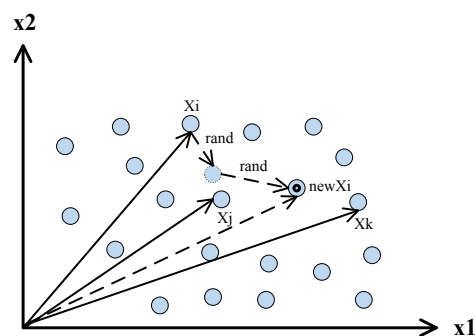
where  $rand$  is a uniformly distributed random number between 0 and 1,  $X_{j\max}$  and  $X_{j\min}$  are the upper boundary and lower boundary of  $j$ -th dimension of each individual, respectively.

### 3.3.2. Iterations of the Three Operators

Now we simply give some certain implementation details about the three operators in the three different phases.

#### (1) Stochastic operator

Stochastic diffusion is a common operation in which elements randomly move and pass one another in an unstable phase. Although the movement of elements is chaotic, it actually obeys a certain law from a statistical point of view. We can use the mean free path [43], which is a distance between an element and two other elements in two successive collisions, to represent the stochastic motion characteristic of elements in an unstable phase. Figure 4 simply shows the process of the free walking path of elements.



**Figure 4.** Two-dimensional example showing the process of the free walking path of elements.

The free walking path of elements is the distance traveled by an element and other individuals through two collisions. Therefore, the stochastic operator of elements may be implemented as follows:

$$newX_i = X_i + rand_1 * (X_j - X_i) + rand_2 * (X_k - X_i) \quad (2)$$

where  $newX_i$  is the new position of  $X_i$  after the stochastic motion,  $rand_1$  and  $rand_2$  are two random vectors, where each element is a random number in the range (0, 1), and the indices  $j$  and  $k$  are mutually exclusive integers randomly chosen from the range between 1 and  $N$  that is also different from the indices  $i$ .

## (2) Shrinkage operator

In a meta-stable phase, an element will be inclined to move closer to the optimal one. From a statistical standpoint, the geometric center is a very important digital characteristic and represents the shrinkage trend of elements in a certain degree. Figure 5 briefly gives the shrinkage trend of elements towards the optimal point.

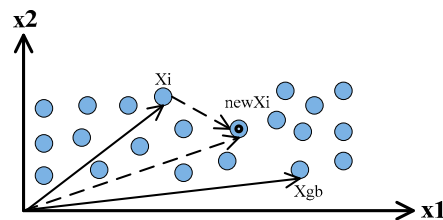


Figure 5. The shrinkage trend of elements towards the optimal point.

Hence, the gradual shrinkage to the central position is the best motion to elements in a meta-stable phase. So, the shrinkage operator of elements may be implemented as follows:

$$newX_i = newX_i + (X_{gb} - X_i) * N(0, 1) \quad (3)$$

where  $newX_i$  is the new position of  $X_i$  after the shrinkage operation,  $X_{gb}$  is the best individual in the population, and  $N(0, 1)$  is a normal random number with mean 0 and standard deviation 1. The Normal distribution is an important family of continuous probability distributions applied in many fields.

## (3) Vibration operator

Elements in a stable phase will be apt to only vibrate about their equilibrium positions. Figure 6 briefly shows the vibration of elements.

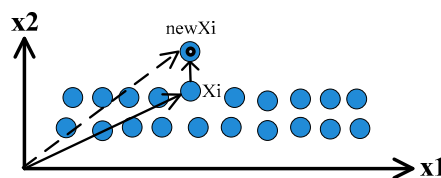


Figure 6. The vibration of elements in an equilibrium position.

Hence, the vibration operator of elements may be implemented as follows:

$$newX_i = newX_i + (2 * rand - 1) * stepSize \quad (4)$$

where  $newX_i$  is the new position of  $X_i$  after the vibration operation,  $rand$  is a uniformly distributed random number in the range (0, 1), and  $stepSize$  is the control parameter which regulates the amplitude

of jitter with a process of evolutionary generation. With the evolution of the phase transition, the amplitude of vibration will gradually become smaller. *stepSize* is described as follows:

$$stepSize = \exp \left( 1 - G / (G - g + 1) \right). \quad (5)$$

where the  $G$  and  $g$  denote the maximum number of iterations and current number of iteration respectively, and  $\exp()$  stands for the exponential function.

### 3.3.3. Individual Selection

In the PTBO algorithm, like other EAs, one-to-one greedy selection is employed by comparing a parent individual and its new generated corresponding offspring. In addition, this greedy selection strategy may raise diversity compared with other strategies, such as tournament selection and rank-based selection. The selection operation at the  $k$ -th generation is described as follows:

$$X_i^{k+1} = \begin{cases} X_i^k & \text{if } f(\text{new}X_i^k) > f(X_i^k) \\ \text{new}X_i^k & \text{if } f(\text{new}X_i^k) \leq f(X_i^k) \end{cases} \quad (6)$$

where  $f(X)$  is the objective function value of each individual.

### 3.4. Flowchart and Implementation Steps of PTBO

As described above, the main flowchart of the PTBO algorithm is given in Figure 7.

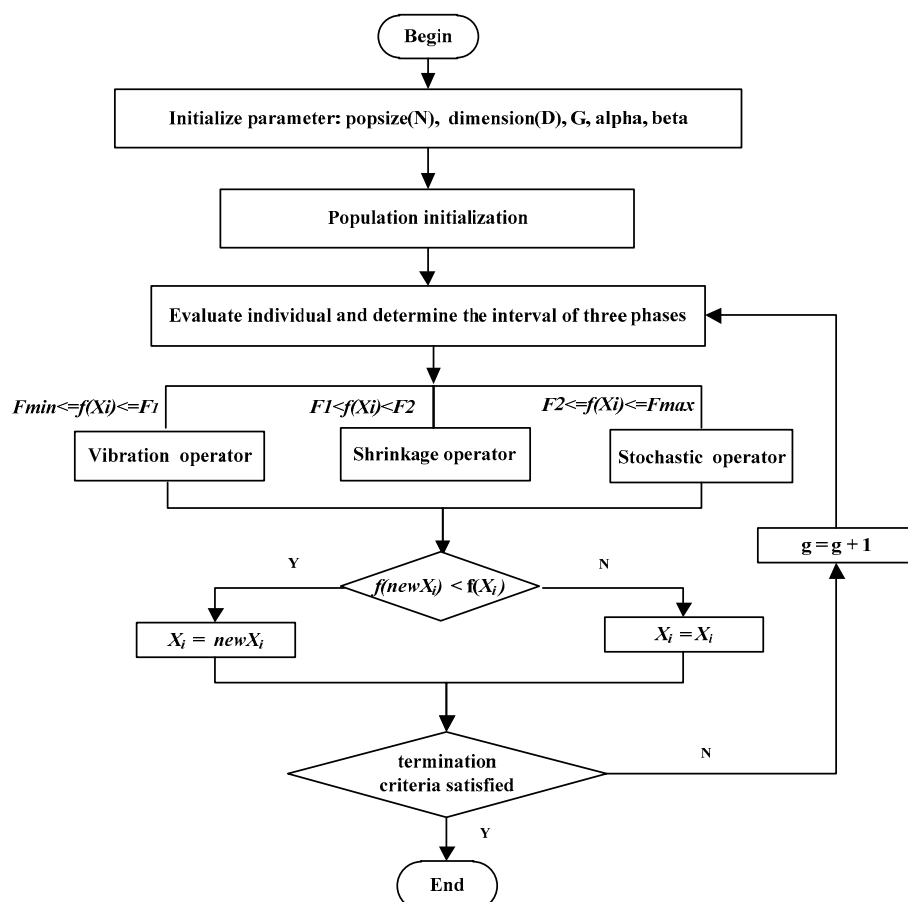


Figure 7. The main flowchart of PTBO.



The implementation steps of PTBO are summarized as follows:

**Step 1.** Initialization: set up algorithm parameters  $N$ ,  $D$ ,  $\alpha$  and  $\beta$ , randomly generate initial population of elements, and set  $g = 0$ ;

**Step 2.** Evaluation and partition interval: calculate the fitness values of all individuals and obtain the  $F_{\max}$  and  $F_{\min}$ , and divide the critical interval of UP, MP and SP according to Table 2;

**Step 3.** Stochastic operator: using Formula (2) to create  $newX_i$ ;

**Step 4.** Shrinkage operator: using Formula (3) to update  $newX_i$ ;

**Step 5.** Vibration operator: using Formula (4) and (5) to update  $newX_i$ ;

**Step 6.** Individual selection: accept  $newX_i$  if  $f(newX_i)$  is better than  $f(X_i)$ ;

**Step 7.** Termination judgment: if termination condition is satisfied, stop the algorithm; otherwise,  $g = g + 1$ , go to Step 3.

## 4. The Analysis of PTBO and a Comparative Study

### 4.1. The Analysis of Time Complexity

For PTBO, the main operations include the operation of population initialization and the stochastic operator, shrinkage operator, and vibration operator. The time complexity of each operation in a single iteration can be computed as follows:

- 1 Population initialization operation:  $O(N * D)$ .
- 2 Stochastic operator:  $O(N * D)$ .
- 3 Shrinkage operator:  $O(N * D)$ .
- 4 Vibration operator:  $O(N * D)$ .

Therefore, the total worst time complexity of PTBO in one iteration is  $3 * O(N * D) + O(N)$ . According to the operational rules of the symbol  $O$ , the worst time complexity of one iteration for PTBO can be simplified as  $O(N * D)$ . It is worth noting that PTBO has the similar time complexity to some popular meta-heuristic algorithms such as PSO ( $O(N * D)$ ).

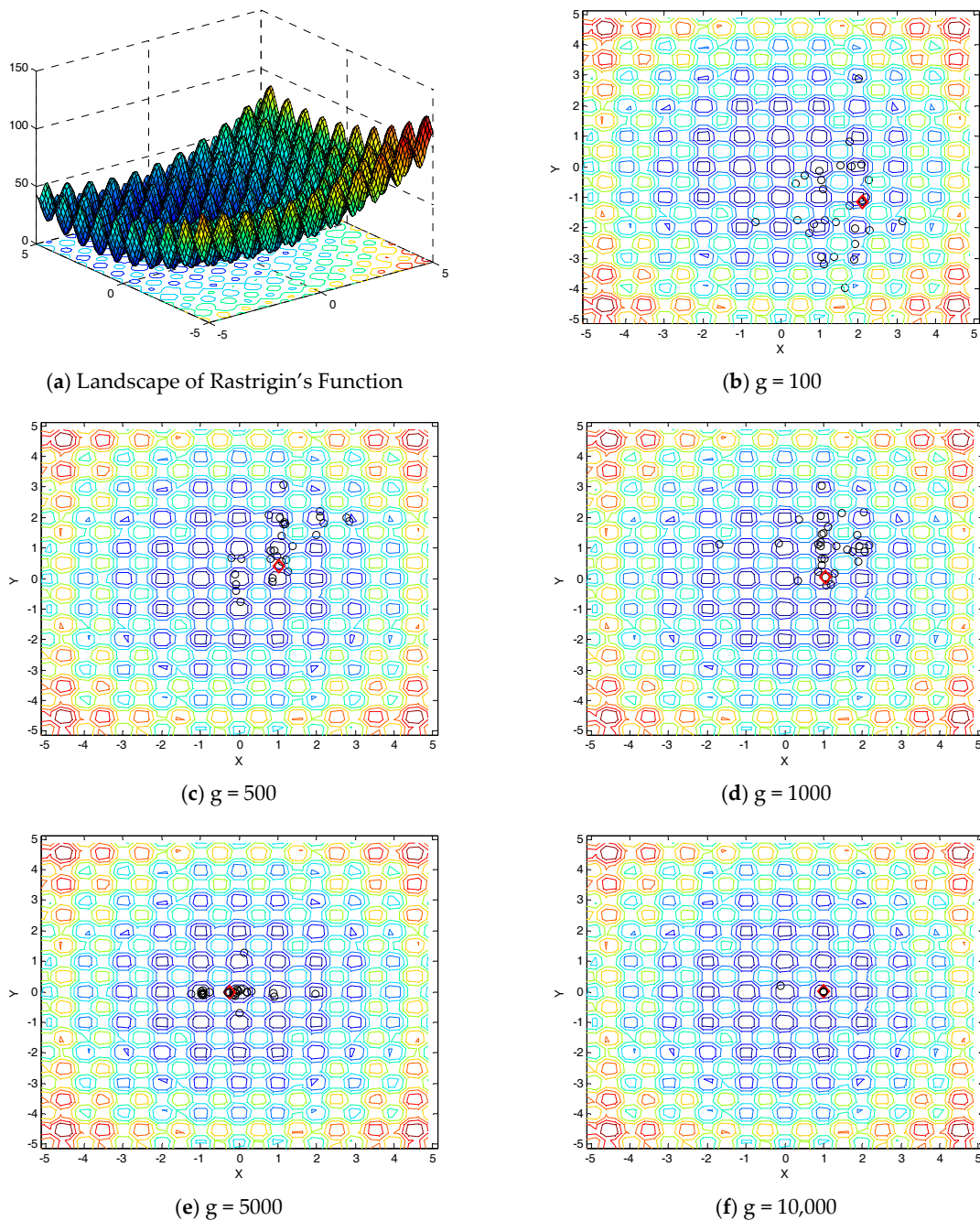
### 4.2. The Dynamic Implementation Analysis of PTBO

In this section, the step-wise procedure for the implementation of PTBO for optimization is presented. For the demonstration of the process, Rastrigin's function [44] is herein considered as an example. Rastrigin's function is a classic test function in optimization theory in which the point of global minimum is surrounded by a large number of local minima. To converge to the global minimum without being stuck at one of these local minima, however, is extremely difficult. Some numerical solvers need to take a long time to converge to it. Three-dimensional contour plot for Rastrigin's function is shown in Figure 8a. Rastrigin's function is described as follows:

$$F(x) = \sum_{i=1}^D \left( x_i^2 - 10 \cos(2\pi x_i) + 10 \right) \quad (7)$$

In this experiment, we use 30 individuals to solve the above minimization problem, and the population distribution at various generations in an evolutionary process is shown in Figure 8b–f, with  $D = 2$ ,  $\alpha = 0.1$  and  $\beta = 0.8$ . In Figure 8b–f, the labels of red diamond represent the optimal point.

From Figure 8b–f, we can observe that the population distribution information can significantly vary at various generations during the run time. PTBO can effectively adapt to a time-varying search space or landscapes.



**Figure 8.** Population distribution at various generations in an evolutionary process of PTBO.

### 4.3. The Differences between PTBO and Other Algorithms

#### 4.3.1. The Differences between PTBO and PSO

Like PSO (Particle Swarm Optimization), PTBO is also introduced to deal with unconstrained global optimization problems with continuous variables. In the representational form of implementation, we can certainly think that PTBO is also based on a particle system in the same way as PSO. However, according to the overall design of the PTBO algorithm, there are some differences between PTBO and the classical PSO.

Firstly, in heuristic thought, PSO is inspired by biological behavior or habits for simulating animal swarm behavior, such as fish schooling and bird flocking, while PTBO is inspired by the phase transition phenomenon of elements in nature. Secondly, in PSO, the direction of a particle is calculated by two best positions, *pbest* and *gbest*. However, the motion direction of an element in PTBO is arbitrarily derived from the other two elements that are different from each other. It may enhance the diversity of the population and ensure the avoidance of premature convergence. Thirdly, in the design of the operators, each particle in PSO contains a velocity item. Nevertheless, in PTBO the concept of velocity does not exist. Besides, PSO uses the position information of *pbest* and *gbest* to record the updating of velocity or position. However, PTBO uses only the position information about *gbest* and the *pbest* position of elements is not considered.

#### 4.3.2. The Differences between PTBO and SMS

SMS (States of Matter Search) is a nature-inspired optimization algorithm based on the simulation of the states of matter phenomenon [35]. Specifically, SMS is devised by considering each state of matter at one different exploration–exploitation ratio by dividing the whole search process into three stages, i.e., gas, liquid, and solid.

Although the sources of inspiration for PTBO and SMS are similar, which are taken from a physical phenomenon about the states of matter, the evolution processes of PTBO and SMS are completely different. The evolution process of SMS is as follows. At first, all individuals in the population perform exploration in the mode of the gas state. Then, after 50% of the iterations, the search mode is changed into the liquid state for 40% of the iterations, i.e., the search between exploration and exploitation. Finally, the evolutionary process enters the stage of exploitation (liquid state) for 10% of the iterations. However, in PTBO, the three phases are coexistent in the entire search process. In other words, the three different phases of individuals execute dynamically different search tasks according to their phase in each generation. Hence, the implementation about the balance of exploration and exploitation between PTBO and SMS is completely different. Besides, the operators of PTBO and SMS are also completely different. In summary, it can be said that there are fundamental differences between PTBO and SMS.

## 5. Experimental Results

### 5.1. Benchmark Functions

In order to verify the effectiveness and robustness of the proposed PTBO algorithm, PTBO is applied in experimental simulation studies for finding the global minimum of the entire 28 test functions of the CEC 2013 special session [44]. The CEC 2013's test suite, which has been improved on the basis of CEC 2005 [38], covers various types of function optimization, and is summarized in Table 4.

The search range of all functions is between  $-100$  and  $100$  in every dimension. These problems are shifted or rotated to increase their complexity, and are treated as black-box problems. The explicit equations of the problems are not allowed to be used. The test suite of Table 4 consists of five uni-modal functions (F01 to F05), 15 multimodal functions (F06 to F20) and eight composition functions (F21 to F28).

**Table 4.** Benchmark functions tested by PTBO (CEC2013).

Type	Function ID	Functions Name
Uni-modal Functions	F01	Sphere Function
	F02	Rotated High Conditioned Elliptic Function
	F03	Rotated Bent Cigar Function
	F04	Rotated Discus Function
	F05	Different Powers Function
Multimodal Functions	F06	Rotated Rosenbrock's Function
	F07	Rotated Schaffers F7 Function
	F08	Rotated Ackley's Function
	F09	Rotated Weierstrass Function
	F10	Rotated Griewank's Function
	F11	Rastrigin's Function
	F12	Rotated Rastrigin's Function
	F13	Non-Continuous Rotated Rastrigin's Function
	F14	Schwefel's Function
	F15	Rotated Schwefel's Function
	F16	Rotated Katsuura Function
	F17	Lunacek Bi_Rastrigin Function
	F18	Rotated Lunacek Bi_Rastrigin Function
	F19	Expanded Griewank's plus Rosenbrock's Function
	F20	Expanded Scaffer's F6 Function
Composition Functions	F21	Composition Function 1 ( $n = 5$ , Rotated)
	F22	Composition Function 2 ( $n = 3$ , Unrotated)
	F23	Composition Function 3 ( $n = 3$ , Rotated)
	F24	Composition Function 4 ( $n = 3$ , Rotated)
	F25	Composition Function 5 ( $n = 3$ , Rotated)
	F26	Composition Function 6 ( $n = 5$ , Rotated)
	F27	Composition Function 7 ( $n = 5$ , Rotated)
	F28	Composition Function 8 ( $n = 5$ , Rotated)

### 5.2. Parameters Determination of the Interval Ratio of PTBO

In our PTBO algorithm, there are two parameters, the alpha and the beta, that need to be allocated to determine the critical intervals about the three phases. In a natural system, we can observe that the elements in the middle meta-stable phase account for the majority, and the elements in the unstable and stable phases occupy only a small proportion. This phenomenon is consistent with the two-eight law (or the 1/5th rule) [2]. For the case of simplicity, we give a value of 0.8 to beta, which is the proportion of the elements in the meta-stable phase. So, the elements in the unstable and stable phases account for 0.2 in total. The specific interval ratio settings of the three phases are shown in Table 5.

**Table 5.** Different interval ratio settings of the three phases.

Ratio Method	No.	Stable Phase	Meta-Stable Phase	Unstable Phase
Fixed ratio	Prop1	0.05	0.8	0.15
	Prop2	0.1	0.8	0.1
	Prop3	0.15	0.8	0.05
Random ratio	Prop4	0.2 * rand	0.8	0.2 * (1 – rand)

In order to determine which interval ratio is the most suitable for the PTBO algorithm, we conducted some compared experiments with 50 independent runs according to Table 5. The compared results of different interval ratios are listed in Table 6. In Table 6, the mean values are listed in the first line, and the standard deviations are displayed in the second line. We can intuitively observe that the ratio of prop4 has the best accuracy results compared with the other three ratios. Hence, in the subsequent experiments, we choose the value of beta to be 0.8, and the proportion of the unstable and stable phases is a random ratio of 0.2 in total.

Table 6. Compared results of different interval ratios.

Func.	Prop1	Prop2	Prop3	Prop4	Func.	Prop1	Prop2	Prop3	Prop4
F01	$6.16 \times 10^{-27}$ $2.08 \times 10^{-26}$	$6.13 \times 10^{-27}$ $3.70 \times 10^{-26}$	<b><math>3.82 \times 10^{-28}</math></b> <b><math>6.04 \times 10^{-28}</math></b>	$4.31 \times 10^{-28}$ $5.96 \times 10^{-28}$	F15	$4.45 \times 10^3$ <b><math>5.99 \times 10^2</math></b>	$4.57 \times 10^3$ $6.55 \times 10^2$	$4.27 \times 10^3$ $7.24 \times 10^2$	<b><math>4.23 \times 10^3</math></b> $7.64 \times 10^2$
F02	<b><math>7.90 \times 10^5</math></b> <b><math>2.89 \times 10^5</math></b>	$8.73 \times 10^5$ $3.25 \times 10^5$	$1.02 \times 10^6$ $3.66 \times 10^5$	$8.63 \times 10^5$ $6.60 \times 10^5$	F16	$4.10 \times 10^{-1}$ $2.01 \times 10^{-1}$	$4.71 \times 10^{-1}$ $2.59 \times 10^{-1}$	$4.11 \times 10^{-1}$ $2.07 \times 10^{-1}$	<b><math>3.94 \times 10^{-1}</math></b> <b><math>1.96 \times 10^{-1}</math></b>
F03	$1.95 \times 10^8$ $2.91 \times 10^8$	$1.12 \times 10^8$ $1.49 \times 10^8$	$8.01 \times 10^7$ $1.22 \times 10^8$	<b><math>3.84 \times 10^7</math></b> <b><math>5.22 \times 10^7</math></b>	F17	$7.58 \times 10^1$ $1.45 \times 10^1$	$7.13 \times 10^1$ $1.25 \times 10^1$	<b><math>6.68 \times 10^1</math></b> <b>9.94</b>	$6.72 \times 10^1$ $1.63 \times 10^1$
F04	<b><math>8.08 \times 10^3</math></b> $2.59 \times 10^3$	$9.03 \times 10^3$ $2.35 \times 10^3$	$1.30 \times 10^4$ <b><math>3.43 \times 10^3</math></b>	$8.26 \times 10^3$ $3.71 \times 10^3$	F18	$7.96 \times 10^1$ $1.47 \times 10^1$	$7.57 \times 10^1$ $1.15 \times 10^1$	$6.99 \times 10^1$ <b><math>1.14 \times 10^1</math></b>	<b><math>6.66 \times 10^1</math></b> $1.23 \times 10^1$
F05	$1.36 \times 10^{-14}$ $2.19 \times 10^{-14}$	$3.00 \times 10^{-15}$ $4.32 \times 10^{-15}$	<b><math>6.38 \times 10^{-22}</math></b> <b><math>1.18 \times 10^{-21}</math></b>	$2.64 \times 10^{-15}$ $4.44 \times 10^{-15}$	F19	$5.05 \times 10^2$ 1.61	$5.05 \times 10^2$ 1.45	<b><math>5.04 \times 10^2</math></b> <b>1.06</b>	<b><math>5.04 \times 10^2</math></b> $8.36 \times 10^{-1}$
F06	<b><math>2.32 \times 10^1</math></b> <b><math>2.57 \times 10^1</math></b>	$3.09 \times 10^1$ $2.88 \times 10^1$	$2.74 \times 10^1$ $2.58 \times 10^1$	$2.53 \times 10^1$ $2.67 \times 10^1$	F20	$1.21 \times 10^1$ <b>1.68</b>	$1.19 \times 10^1$ 2.01	$1.26 \times 10^1$ 2.07	<b><math>1.15 \times 10^1</math></b> 2.12
F07	$5.30 \times 10^1$ $2.31 \times 10^1$	$5.01 \times 10^1$ $1.90 \times 10^1$	$4.13 \times 10^1$ $1.93 \times 10^1$	<b><math>3.04 \times 10^1</math></b> <b><math>1.73 \times 10^1</math></b>	F21	$3.01 \times 10^2$ $7.76 \times 10^1$	$3.30 \times 10^2$ $8.86 \times 10^1$	$3.22 \times 10^2$ <b><math>7.19 \times 10^1</math></b>	<b><math>2.97 \times 10^2</math></b> $8.01 \times 10^1$
F08	<b><math>2.09 \times 10^1</math></b> <b><math>6.18 \times 10^{-2}</math></b>	<b><math>2.09 \times 10^1</math></b> $6.57 \times 10^{-2}$	<b><math>2.09 \times 10^1</math></b> $6.50 \times 10^{-2}$	<b><math>2.09 \times 10^1</math></b> $6.50 \times 10^{-2}$	F22	$1.01 \times 10^3$ $4.92 \times 10^2$	<b><math>8.03 \times 10^2</math></b> <b><math>2.79 \times 10^2</math></b>	$9.74 \times 10^2$ $4.68 \times 10^2$	$8.61 \times 10^2$ $3.11 \times 10^2$
F09	$2.49 \times 10^1$ 5.33	$2.39 \times 10^1$ <b>5.24</b>	$2.43 \times 10^1$ 5.54	<b><math>1.90 \times 10^1</math></b> 5.26	F23	$5.25 \times 10^3$ $8.49 \times 10^2$	$5.16 \times 10^3$ <b><math>6.32 \times 10^2</math></b>	$5.07 \times 10^3$ $7.01 \times 10^2$	<b><math>4.15 \times 10^3</math></b> $1.40 \times 10^3$
F10	$3.50 \times 10^{-1}$ $2.03 \times 10^{-1}$	$3.39 \times 10^{-1}$ $1.85 \times 10^{-1}$	$3.43 \times 10^{-1}$ <b><math>1.73 \times 10^{-1}</math></b>	$3.27 \times 10^{-1}$ $1.77 \times 10^{-1}$	F24	$2.42 \times 10^2$ <b>8.21</b>	$2.35 \times 10^2$ 8.66	<b><math>2.33 \times 10^2</math></b> 8.46	<b><math>2.33 \times 10^2</math></b> $1.04 \times 10^1$
F11	$4.85 \times 10^1$ $1.60 \times 10^1$	$4.05 \times 10^1$ $1.38 \times 10^1$	<b><math>3.89 \times 10^1</math></b> <b><math>1.12 \times 10^1</math></b>	$3.92 \times 10^1$ $1.21 \times 10^1$	F25	$2.81 \times 10^2$ $1.34 \times 10^1$	$2.79 \times 10^2$ $1.46 \times 10^1$	$2.78 \times 10^2$ $1.32 \times 10^1$	<b><math>2.74 \times 10^2</math></b> <b><math>1.02 \times 10^1</math></b>
F12	$1.40 \times 10^2$ $2.82 \times 10^1$	$1.28 \times 10^2$ $3.41 \times 10^1$	$1.21 \times 10^2$ $3.72 \times 10^1$	<b><math>6.32 \times 10^1</math></b> <b><math>2.51 \times 10^1</math></b>	F26	$2.53 \times 10^2$ $6.85 \times 10^1$	$2.77 \times 10^2$ $6.93 \times 10^1$	<b><math>2.48 \times 10^2</math></b> $6.71 \times 10^1$	$2.51 \times 10^2$ <b><math>6.57 \times 10^1</math></b>
F13	$1.61 \times 10^2$ $3.13 \times 10^1$	$1.53 \times 10^2$ $2.76 \times 10^1$	$1.47 \times 10^2$ <b><math>2.67 \times 10^1</math></b>	<b><math>1.18 \times 10^2</math></b> $3.07 \times 10^1$	F27	$7.50 \times 10^2$ <b><math>9.33 \times 10^1</math></b>	$7.41 \times 10^2$ $1.21 \times 10^2$	$7.23 \times 10^2$ $1.34 \times 10^2$	<b><math>6.38 \times 10^2</math></b> $1.16 \times 10^2$
F14	$9.52 \times 10^2$ $3.28 \times 10^2$	<b><math>8.63 \times 10^2</math></b> $3.31 \times 10^2$	$9.35 \times 10^2$ $3.49 \times 10^2$	$9.47 \times 10^2$ <b><math>2.81 \times 10^2</math></b>	F28	$3.93 \times 10^2$ $4.61 \times 10^2$	$3.46 \times 10^2$ $2.74 \times 10^2$	$3.51 \times 10^2$ $3.92 \times 10^2$	<b><math>2.96 \times 10^2</math></b> <b><math>2.83 \times 10^1</math></b>

Bold text is the best result obtained by different interval ratios.

### 5.3. Experimental Platform and Algorithms' Parameter Settings

For a fair comparison, all of the experiments are conducted on the same machine with an Intel 3.4 GHz central processing unit (CPU) and 4GB memory. The operating system is Windows 7 with MATLAB 8.0.0.783(R2012b).

On the test functions, we compare PTBO with classic PSO, DE, and six other recent popular meta-heuristic algorithms, which are BA, CS, BSO, WWO, WCA, and SMS. The first three algorithms belong to the second category, i.e., swarm intelligence, and the remaining three algorithms belong to the third category, i.e., intelligent algorithms simulating physical phenomena. The parameters adopted for the PTBO algorithm and the compared algorithms are given in Table 7.

**Table 7.** The parameter settings of the compared algorithms.

No.	Algorithm	Parameter Setting
1	PSO	$\omega \in [0.9, 0.4], c1 = c2 = 2, V_{\max} = 0.2 * range$
2	DE	$F = 0.5, CR = 0.9$
3	BA	$A = 0.25, r = 0.5$
4	CS	$pa = 0.25, stepsize = 0.05$
5	BSD	$m = 5, p_{replace} = 0.2, p_{one} = 0.8, p_{one\_center} = 0.4, p_{two\_center} = 0.5, k = 20$
6	WWO	$hMax = 12, alpha = 1.0026, betaMax = 0.25, betaMin = 0.001$
7	WCA	$Nsr = 4, dmax = 1e - 16$
8	SMS	Gas state: $\rho \in [0.8, 1], \beta = 0.8, \alpha = 0.8, H = 0.9$ Liquid state: $\rho \in [0.3, 0.6], \beta = 0.4, \alpha = 0.2, H = 0.2$ Solid state: $\rho \in [0, 0.1], \beta = 0.1, \alpha = 0, H = 0$
9	PTBO	alpha is random ratio, beta = 0.8

PSO: particle swarm optimization; DE: Differential Evolution; BA: Bat Algorithm; CS: Cuckoo Search; BSO: Brain Storm Optimization; WWO: Water Wave Optimization; WCA: Water Cycle Algorithm; SMS: States of Matter Search.

### 5.4. The Compared Experimental Results

Each of the experiments was repeated for 50 independent runs with different random seeds, and the average function values of the best solutions were recorded.

#### 5.4.1. Comparisons on Solution Accuracy

The accuracy results of the uni-modal, multimodal, and composition functions are given in Tables 8–10, respectively. The accuracy results are in terms of the mean optimum solution and the standard deviation of the solutions obtained by each algorithm over 300,000 function evaluation times (FES) or 10,000 maximum generations. In all experiments, the dimensions of all problems are 30. The best results among the algorithms are shown in bold. In each row of the three tables, the mean values are listed in the first line, and the standard deviations are displayed in the second line. A two-tailed t-test [45] is performed with a 0.05 significance level to evaluate whether the median fitness values of two sets of obtained results are statistically different from each other. In the below three tables, if PTBO significantly outperforms another algorithm, a ‘ $\Psi$ ’ is labeled in the back of the corresponding result obtained by this algorithm. Corresponding to ‘ $\xi$ ’, ‘ $\xi$ ’ denotes that PTBO is worse than other algorithms, and ‘ $\sim$ ’ denotes that there is no significance between PTBO and the compared algorithm. At the last row of each table, a summary of the total number of ‘ $\Psi$ ’, ‘ $\xi$ ’, and ‘ $\sim$ ’ is calculated.

#### (1) The accuracy results of uni-modal functions

The results of the uni-modal functions are shown in Table 8 in terms of the mean optimum solution and the standard deviation of the solutions.

In Table 8, among the five functions, PTBO has yielded the best results on two of them (F03 and F05). Although PTBO is worse than WCA, which obtains the best results on F01, F02, and F04, we observe from the statistical results that the performance of PTBO in uni-modal functions is significantly better than PSO, DE, BA, CS, BSO, WWO, and SMS.

**Table 8.** The results of solution accuracy for the uni-modal functions.

Func.	PTBO	PSO	DE	BA	CS	BSO	WWO	WCA	SMS
F01	$4.31 \times 10^{-28}$ $5.96 \times 10^{-28}$	$3.56 \times 10^3 \dagger$ $2.65 \times 10^3$	$2.65 \times 10^3 \dagger$ $8.98 \times 10^{-1}$	$1.05 \times 10^{-3} \dagger$ $1.29 \times 10^{-4}$	$4.46 \times 10^3 \dagger$ $3.10 \times 10^2$	$3.71 \times 10^{-3} \dagger$ $5.71 \times 10^{-3}$	$2.62 \times 10^{-28} \xi$ <b><math>1.30 \times 10^{-28}</math></b>	<b><math>1.58 \times 10^{-28} \xi</math></b> $1.45 \times 10^{-28}$	$4.63 \times 10^4 \dagger$ $4.24 \times 10^3$
F02	$8.63 \times 10^5$ $6.60 \times 10^5$	$1.12 \times 10^7 \dagger$ $1.56 \times 10^7$	$1.56 \times 10^7 \xi$ <b><math>2.05 \times 10^5</math></b>	$4.82 \times 10^4 \xi$ $1.81 \times 10^4$	$4.21 \times 10^7 \dagger$ $4.04 \times 10^6$	$1.17 \times 10^6 \dagger$ $3.35 \times 10^5$	$1.20 \times 10^6 \dagger$ $7.19 \times 10^5$	<b><math>7.53 \times 10^5 \xi</math></b> $7.54 \times 10^5$	$4.91 \times 10^8 \dagger$ $8.49 \times 10^7$
F03	<b><math>3.84 \times 10^7</math></b> $5.22 \times 10^7$	$4.98 \times 10^{10} \dagger$ $4.72 \times 10^{10}$	$4.72 \times 10^{10} \dagger$ $3.57 \times 10^7$	$8.96 \times 10^8 \dagger$ $3.48 \times 10^8$	$1.72 \times 10^{10} \dagger$ $2.11 \times 10^9$	$1.66 \times 10^8 \dagger$ $2.22 \times 10^8$	$3.98 \times 10^8 \dagger$ $6.71 \times 10^8$	$2.94 \times 10^9 \dagger$ $4.30 \times 10^9$	$1.00 \times 10^{10} \dagger$ <b>0.00</b>
F04	$8.26 \times 10^3$ $3.71 \times 10^3$	$5.64 \times 10^3 \xi$ $8.41 \times 10^3$	$8.41 \times 10^3 \dagger$ $1.27 \times 10^3$	$2.42 \times 10^4 \dagger$ $3.91 \times 10^2$	$6.05 \times 10^4 \dagger$ $6.61 \times 10^2$	$5.44 \times 10^3 \xi$ $2.22 \times 10^3$	$5.57 \times 10^4 \dagger$ $1.05 \times 10^4$	<b><math>6.55 \times 10^1 \xi</math></b> <b><math>2.07 \times 10^2</math></b>	$8.14 \times 10^4 \dagger$ $9.79 \times 10^3$
F05	<b><math>2.64 \times 10^{-15}</math></b> <b><math>4.44 \times 10^{-15}</math></b>	$1.58 \times 10^3 \dagger$ $1.25 \times 10^3$	$1.25 \times 10^3 \dagger$ $1.84 \times 10^1$	$1.37 \times 10^{-2} \dagger$ $2.08 \times 10^{-3}$	$1.09 \times 10^3 \dagger$ $6.70 \times 10^1$	$4.12 \times 10^{-2} \dagger$ $1.74 \times 10^{-2}$	$5.53 \times 10^{-2} \dagger$ $2.90 \times 10^{-1}$	$2.95 \times 10^{-12} \dagger$ $7.89 \times 10^{-12}$	$1.38 \times 10^4 \dagger$ $2.66 \times 10^3$
	$\dagger/\xi/\sim$	4/1/0	4/1/0	4/1/0	5/0/0	4/1/0	4/1/0	2/3/0	5/0/0

" $\dagger$ ", " $\xi$ ", and " $\sim$ " denote that the performance of PTBO is better than, worse than, and similar to that of the corresponding algorithm, respectively. Bold text is the best result obtained by the compared algorithms.

**Table 9.** The results of solution accuracy for the multimodal functions.

Function	PTBO	PSO	DE	BA	CS	BSO	WWO	WCA	SMS
F06	$2.53 \times 10^1$ $2.67 \times 10^1$	$2.51 \times 10^2 \ddagger$ $2.11 \times 10^2$	$3.20 \times 10^1 \ddagger$ $2.51 \times 10^1$	<b>1.36</b> $\xi$ <b>6.61</b>	$4.99 \times 10^2 \ddagger$ $5.70 \times 10^1$	$4.30 \times 10^1 \ddagger$ $2.84 \times 10^1$	$5.51 \times 10^1 \ddagger$ $2.76 \times 10^1$	$3.27 \times 10^1 \ddagger$ $2.67 \times 10^1$	$6.18 \times 10^3 \ddagger$ $9.16 \times 10^2$
F07	$3.04 \times 10^1$ $1.73 \times 10^1$	$1.19 \times 10^2 \ddagger$ $4.67 \times 10^1$	<b>2.44</b> $\times 10^1 \xi$ $1.43 \times 10^1$	$2.59 \times 10^8 \ddagger$ $4.83 \times 10^7$	$2.51 \times 10^8 \ddagger$ $2.44 \times 10^7$	$1.59 \times 10^2 \ddagger$ $1.11 \times 10^2$	$8.91 \times 10^1 \ddagger$ $3.02 \times 10^1$	$1.90 \times 10^2 \ddagger$ $4.00 \times 10^1$	$5.50 \times 10^3 \ddagger$ $3.85 \times 10^3$
F08	<b>2.09</b> $\times 10^1$ $6.50 \times 10^{-2}$	<b>2.09</b> $\times 10^1 \sim$ $5.71 \times 10^{-2}$	<b>2.09</b> $\times 10^1 \sim$ <b>4.33</b> $\times 10^{-2}$	$2.10 \times 10^1 \sim$ $6.52 \times 10^{-2}$	$2.11 \times 10^1 \sim$ $6.15 \times 10^{-2}$	<b>2.09</b> $\times 10^1 \sim$ $7.40 \times 10^{-2}$	<b>2.09</b> $\times 10^1 \sim$ $5.56 \times 10^{-2}$	<b>2.09</b> $\times 10^1 \sim$ $5.56 \times 10^{-2}$	<b>2.09</b> $\times 10^1 \sim$ $5.31 \times 10^{-2}$
F09	$1.90 \times 10^1$ 5.26	$2.18 \times 10^1 \sim$ 2.73	<b>1.86</b> $\times 10^1 \xi$ 4.40	$5.77 \times 10^1 \ddagger$ 3.29	$5.73 \times 10^1 \ddagger$ <b>1.16</b>	$3.42 \times 10^1 \ddagger$ 3.72	$2.76 \times 10^1 \ddagger$ 4.29	$3.40 \times 10^1 \ddagger$ 3.01	$4.00 \times 10^1 \ddagger$ 1.48
F10	$3.27 \times 10^{-1}$ $1.77 \times 10^{-1}$	$7.01 \times 10^2 \ddagger$ $4.13 \times 10^2$	$6.65 \times 10^{-1} \sim$ 1.21	1.14 $\ddagger$ $6.95 \times 10^{-1}$	$9.05 \times 10^2 \ddagger$ $4.90 \times 10^1$	$9.62 \times 10^{-1} \sim$ $2.16 \times 10^{-1}$	$1.41 \times 10^{-1} \xi$ <b>7.49</b> $\times 10^{-2}$	<b>3.17</b> $\times 10^{-1} \sim$ $2.00 \times 10^{-1}$	$5.60 \times 10^3 \ddagger$ $6.76 \times 10^2$
F11	<b>3.92</b> $\times 10^1$ <b>1.21</b> $\times 10^1$	$1.05 \times 10^2 \ddagger$ $5.68 \times 10^1$	$2.27 \times 10^1 \xi$ 7.87	$9.90 \times 10^2 \ddagger$ $2.71 \times 10^1$	$9.90 \times 10^2 \ddagger$ $2.24 \times 10^1$	$6.03 \times 10^2 \ddagger$ $1.03 \times 10^2$	$1.22 \times 10^2 \ddagger$ $3.80 \times 10^1$	$1.05 \times 10^2 \ddagger$ $4.84 \times 10^1$	$7.65 \times 10^2 \ddagger$ $6.72 \times 10^1$
F12	$6.32 \times 10^1$ $2.51 \times 10^1$	$1.07 \times 10^2 \ddagger$ $3.62 \times 10^1$	<b>4.14</b> $\times 10^1 \xi$ <b>1.21</b> $\times 10^1$	$9.50 \times 10^2 \ddagger$ $1.28 \times 10^1$	$9.52 \times 10^2 \ddagger$ $1.34 \times 10^1$	$5.97 \times 10^2 \ddagger$ $1.04 \times 10^2$	$1.36 \times 10^2 \ddagger$ $3.51 \times 10^1$	$3.38 \times 10^2 \ddagger$ $9.25 \times 10^1$	$7.63 \times 10^2 \ddagger$ $5.78 \times 10^1$
F13	<b>1.18</b> $\times 10^2$ <b>3.07</b> $\times 10^1$	$2.80 \times 10^2 \ddagger$ $3.13 \times 10^1$	$2.09 \times 10^2 \ddagger$ $3.63 \times 10^1$	$1.12 \times 10^3 \ddagger$ $4.65 \times 10^1$	$1.12 \times 10^3 \ddagger$ $3.75 \times 10^1$	$6.51 \times 10^2 \ddagger$ $7.86 \times 10^1$	$2.07 \times 10^2 \ddagger$ $5.72 \times 10^1$	$3.46 \times 10^2 \ddagger$ $6.77 \times 10^1$	$7.68 \times 10^2 \ddagger$ $5.35 \times 10^1$
F14	<b>9.47</b> $\times 10^2$ <b>2.81</b> $\times 10^2$	$2.00 \times 10^3 \ddagger$ $6.45 \times 10^2$	$1.46 \times 10^3 \ddagger$ $4.71 \times 10^2$	$5.72 \times 10^3 \ddagger$ $4.38 \times 10^2$	$5.90 \times 10^3 \ddagger$ $4.38 \times 10^2$	$4.42 \times 10^3 \ddagger$ $7.77 \times 10^2$	$3.23 \times 10^3 \ddagger$ $6.47 \times 10^2$	$2.37 \times 10^3 \ddagger$ $7.83 \times 10^2$	$7.72 \times 10^3 \ddagger$ $3.88 \times 10^2$
F15	$4.23 \times 10^3$ $7.64 \times 10^2$	<b>4.02</b> $\times 10^3 \xi$ $6.52 \times 10^2$	$7.21 \times 10^3 \ddagger$ $3.53 \times 10^2$	$5.02 \times 10^3 \ddagger$ $2.58 \times 10^2$	$4.99 \times 10^3 \ddagger$ <b>2.57</b> $\times 10^2$	$4.33 \times 10^3 \sim$ $6.67 \times 10^2$	$5.16 \times 10^3 \ddagger$ $1.72 \times 10^3$	$4.72 \times 10^3 \ddagger$ $6.74 \times 10^2$	$7.59 \times 10^3 \ddagger$ $2.60 \times 10^2$
F16	$3.94 \times 10^{-1}$ $1.96 \times 10^{-1}$	1.94 $\ddagger \sim$ $4.89 \times 10^{-1}$	2.49 $\ddagger$ $3.29 \times 10^{-1}$	1.98 $\ddagger$ $9.87 \times 10^{-1}$	1.89 $\ddagger$ $8.45 \times 10^{-1}$	<b>3.13</b> $\times 10^{-1} \xi$ <b>1.16</b> $\times 10^{-1}$	2.00 $\ddagger$ $8.11 \times 10^{-1}$	1.63 $\ddagger$ $4.95 \times 10^{-1}$	2.59 $\ddagger$ $3.08 \times 10^{-1}$
F17	<b>6.72</b> $\times 10^1$ $1.63 \times 10^1$	$8.45 \times 10^1 \ddagger$ $3.92 \times 10^1$	$6.39 \times 10^1 \xi$ 9.90	$8.29 \times 10^2 \ddagger$ <b>1.12</b> $\times 10^1$	$8.42 \times 10^2 \ddagger$ $1.41 \times 10^1$	$6.10 \times 10^2 \ddagger$ $1.03 \times 10^2$	$1.43 \times 10^2 \ddagger$ $3.67 \times 10^1$	$2.39 \times 10^2 \ddagger$ $1.09 \times 10^2$	$1.51 \times 10^3 \ddagger$ $1.02 \times 10^2$
F18	<b>6.66</b> $\times 10^1$ $1.23 \times 10^1$	$1.26 \times 10^2 \ddagger$ $4.99 \times 10^1$	$2.92 \times 10^2 \ddagger$ $1.86 \times 10^1$	$8.26 \times 10^2 \ddagger$ $1.17 \times 10^1$	$8.33 \times 10^2 \ddagger$ <b>1.02</b> $\times 10^1$	$5.13 \times 10^2 \ddagger$ $9.43 \times 10^1$	$1.42 \times 10^2 \ddagger$ $3.31 \times 10^1$	$4.66 \times 10^2 \ddagger$ $1.43 \times 10^2$	$1.53 \times 10^3 \ddagger$ $8.20 \times 10^1$
F19	<b>5.04</b> $\times 10^2$ <b>8.36</b> $\times 10^{-1}$	$2.16 \times 10^3 \ddagger$ $3.38 \times 10^3$	$5.65 \times 10^2 \ddagger$ 2.84	$3.37 \times 10^3 \ddagger$ $1.85 \times 10^2$	$1.55 \times 10^3 \ddagger$ $7.16 \times 10^1$	$5.12 \times 10^2 \sim$ 2.41	$5.16 \times 10^2 \sim$ 1.71	$5.14 \times 10^2 \sim$ 5.60	$5.90 \times 10^5 \ddagger$ $2.78 \times 10^5$
F20	<b>1.15</b> $\times 10^1$ 2.12	$1.24 \times 10^1 \ddagger$ $8.98 \times 10^{-1}$	$1.28 \times 10^1 \ddagger$ $5.20 \times 10^{-1}$	$1.50 \times 10^1 \ddagger$ <b>0.00</b>	$1.50 \times 10^1 \ddagger$ <b>0.00</b>	$1.45 \times 10^1 \ddagger$ $9.18 \times 10^{-2}$	$1.29 \times 10^1 \ddagger$ 1.00	$1.50 \times 10^1 \ddagger$ <b>0.00</b>	$1.50 \times 10^1 \ddagger$ $5.71 \times 10^{-2}$
$\ddagger/\xi/\sim$		12/1/2	8/5/2	13/1/1	14/0/1	10/1/4	12/1/2	10/3/2	14/0/1

“ $\ddagger$ ”, “ $\xi$ ”, and “ $\sim$ ” denote that the performance of PTBO is better than, worse than, and similar to that of the corresponding algorithm, respectively. Bold text is the best result obtained by the compared algorithms.



### (2) The accuracy results of multimodal functions

From Table 9, it can be observed that the mean value and the standard deviation value of PSO displays the best results for the function F15. DE obtains the best results on F07, F09, and F12, and BA has the best result for the function F06. BSO has the best result for the function F16. WCA obtains the best results on F8 and F10. CS, WWO, and SMS do not obtain any best result except for the function F8. With regard to the function F8, there is no big difference about all eight algorithms. The PTBO algorithm performs well for the functions F11, F13, F14, F17, F18, and F20, and according to the data of the last row in Table 9, it can be concluded that the PTBO algorithm has good performance in solution accuracy for multimodal benchmark functions.

### (3) The accuracy results of composition functions

It can be seen from Table 10 that DE acquires the best results on F25, and WWO obtains the best results on F26. However, PTBO obtains the best results on F21, F22, F23, F24, F27, and F28. In general, PTBO shows the best overall performance from the statistical results according to the data of the last row in Table 10.

**Table 10.** The results of solution accuracy for the composition functions.

Function	PTBO	PSO	DE	BA	CS	BSO	WWO	WCA	SMS
F21	$2.97 \times 10^2$ $8.01 \times 10^1$	$4.91 \times 10^2 \dagger$ $2.21 \times 10^2$	$4.20 \times 10^2 \dagger$ $8.52 \times 10^1$	$3.01 \times 10^2 \sim$ $2.68 \times 10^{-2}$	$1.50 \times 10^3 \dagger$ $8.81 \times 10^1$	$3.48 \times 10^2 \dagger$ $8.87 \times 10^1$	$3.25 \times 10^2 \sim$ $8.01 \times 10^1$	$3.51 \times 10^2 \sim$ $8.51 \times 10^1$	$3.64 \times 10^3 \dagger$ $2.17 \times 10^2$
F22	$8.61 \times 10^2$ $3.11 \times 10^2$	$2.17 \times 10^3 \dagger$ $6.53 \times 10^2$	$1.33 \times 10^3 \dagger$ $3.94 \times 10^2$	$8.52 \times 10^3 \dagger$ <b><math>1.76 \times 10^2</math></b>	$8.52 \times 10^3 \dagger$ $1.93 \times 10^2$	$5.62 \times 10^3 \dagger$ $8.61 \times 10^2$	$4.09 \times 10^3 \dagger$ $7.54 \times 10^2$	$4.01 \times 10^3 \dagger$ $1.23 \times 10^3$	$8.39 \times 10^3 \dagger$ $3.46 \times 10^2$
F23	$4.15 \times 10^3$ $1.40 \times 10^3$	$4.16 \times 10^3 \sim$ $8.98 \times 10^2$	$6.61 \times 10^3 \dagger$ $9.13 \times 10^2$	$7.55 \times 10^3 \dagger$ $5.35 \times 10^2$	$7.74 \times 10^3 \dagger$ $3.99 \times 10^2$	$5.67 \times 10^3 \dagger$ $6.67 \times 10^2$	$5.40 \times 10^3 \sim$ $1.33 \times 10^3$	$6.64 \times 10^3 \dagger$ $6.71 \times 10^2$	$8.25 \times 10^3 \dagger$ <b><math>2.72 \times 10^2</math></b>
F24	$2.33 \times 10^2$ <b><math>1.04 \times 10^1</math></b>	$2.84 \times 10^2 \dagger$ $1.23 \times 10^1$	$2.95 \times 10^2 \dagger$ $1.05 \times 10^1$	$7.03 \times 10^2 \dagger$ $9.88 \times 10^1$	$7.28 \times 10^2 \dagger$ $1.43 \times 10^1$	$3.34 \times 10^2 \dagger$ $1.89 \times 10^1$	$2.63 \times 10^2 \dagger$ $1.44 \times 10^1$	$3.08 \times 10^2 \dagger$ $3.54 \times 10^1$	$3.58 \times 10^2 \dagger$ $1.05 \times 10^1$
F25	$2.74 \times 10^2$ $1.02 \times 10^1$	$3.03 \times 10^2 \sim$ 9.32	<b><math>2.66 \times 10^2 \xi</math></b> <b>7.55</b>	$4.48 \times 10^2 \dagger$ $2.23 \times 10^1$	$4.45 \times 10^2 \dagger$ 4.24	$3.56 \times 10^2 \dagger$ $1.76 \times 10^1$	$3.04 \times 10^2 \sim$ $1.47 \times 10^1$	$3.17 \times 10^2 \sim$ $1.11 \times 10^1$	$3.73 \times 10^2 \dagger$ 8.47
F26	$2.51 \times 10^2$ $6.57 \times 10^1$	$3.23 \times 10^2 \dagger$ $6.95 \times 10^1$	$2.70 \times 10^2 \dagger$ $7.11 \times 10^1$	$5.52 \times 10^2 \dagger$ $3.00 \times 10^1$	$5.15 \times 10^2 \dagger$ $3.47 \times 10^1$	$2.38 \times 10^2 \xi$ $7.46 \times 10^1$	<b><math>2.00 \times 10^2 \xi</math></b> <b><math>3.74 \times 10^{-2}</math></b>	$3.51 \times 10^2 \sim$ $7.68 \times 10^1$	$2.62 \times 10^2 \dagger$ $1.61 \times 10^1$
F27	<b><math>6.38 \times 10^2</math></b> $1.16 \times 10^2$	$9.62 \times 10^2 \dagger$ $7.94 \times 10^1$	$7.14 \times 10^2 \sim$ $7.45 \times 10^1$	$3.30 \times 10^3 \dagger$ $2.89 \times 10^2$	$2.83 \times 10^3 \dagger$ $8.94 \times 10^1$	$1.36 \times 10^3 \dagger$ $9.72 \times 10^1$	$9.90 \times 10^2 \dagger$ $1.20 \times 10^2$	$1.22 \times 10^3 \dagger$ $8.93 \times 10^1$	$1.50 \times 10^3 \dagger$ <b><math>4.49 \times 10^1</math></b>
F28	<b><math>2.96 \times 10^2</math></b> $2.83 \times 10^1$	$2.11 \times 10^3 \dagger$ $5.50 \times 10^2$	$3.00 \times 10^2 \sim$ <b><math>1.09 \times 10^{-1}</math></b>	$7.53 \times 10^3 \dagger$ $4.46 \times 10^2$	$8.82 \times 10^3 \dagger$ $2.61 \times 10^2$	$4.95 \times 10^3 \dagger$ $7.15 \times 10^2$	$5.36 \times 10^2 \dagger$ $5.90 \times 10^2$	$2.03 \times 10^3 \dagger$ $1.69 \times 10^3$	$5.34 \times 10^3 \dagger$ $4.17 \times 10^2$
$\dagger/\xi/\sim$		6/0/2	5/1/2	7/0/1	8/0/0	7/0/1	4/1/3	5/0/3	8/0/0

" $\dagger$ ", " $\xi$ ", and " $\sim$ " denote that the performance of PTBO is better than, worse than, and similar to that of the corresponding algorithm, respectively. Bold text is the best result obtained by the compared algorithms.

### (4) The total results of solution accuracy of the 28 functions

A summary of the total number of ' $\dagger$ ', ' $\xi$ ', and ' $\sim$ ' about the solution accuracy of 28 functions is given in Table 11. From Table 11, it can be observed that PTBO has the best performance of the nine algorithms.

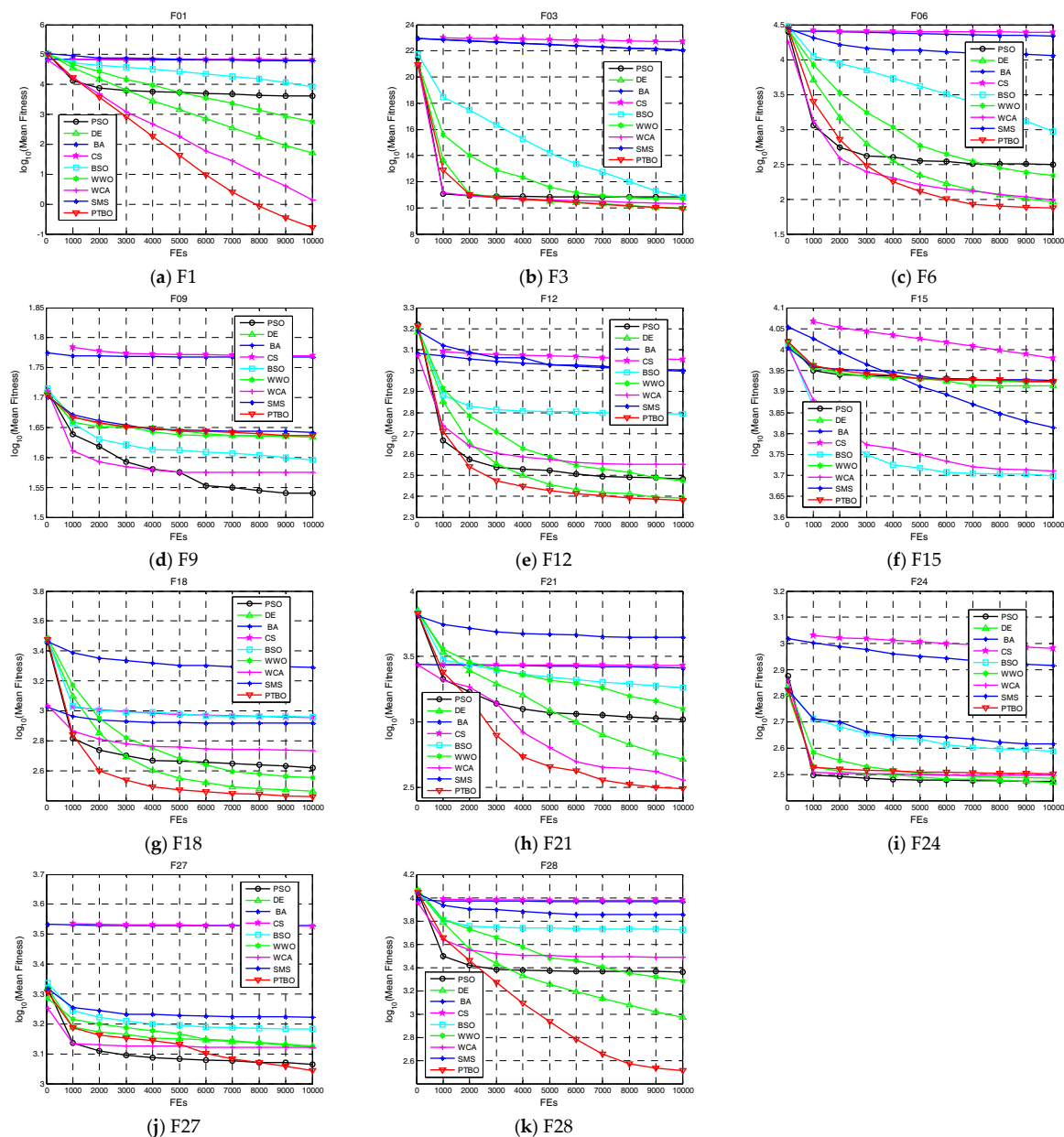
**Table 11.** The total results of solution accuracy for the 28 functions.

PTBO vs.	PSO	DE	BA	CS	BSO	WWO	WCA	SMS
F01–05	4/1/0	4/1/0	4/1/0	5/0/0	4/1/0	4/1/0	2/3/0	5/0/0
F06–20	12/1/2	8/5/2	13/1/1	14/0/1	10/1/4	12/1/2	10/3/2	14/0/1
F21–28	6/0/2	5/1/2	7/0/1	8/0/0	7/0/1	4/1/3	5/0/3	8/0/0
$\dagger/\xi/\sim$	22/2/4	17/7/4	24/4/2	27/0/1	21/2/5	20/3/5	17/6/5	27/0/1

#### 5.4.2. The Comparison Results of Convergence Speed

Due to page limitation, Figure 9 presents the convergence graphs of parts of the 28 test functions in terms of the mean fitness values achieved by each of the nine algorithms for 50 runs. From Figure 9,

it can be observed that PTBO converges towards the optimal values faster than the other algorithms in most cases, i.e., F1, F6, F12, F18, F21, F27, and F28.



**Figure 9.** Convergence performance of the compared algorithms on parts of functions.

### 5.4.3. The Comparison Results of Wilcoxon Signed-Rank Test

To further statistically compare PTBO with the other eight algorithms, a Wilcoxon signed-rank test [45] has been carried out to show the differences between PTBO and the other algorithms. The  $p$ -values on every function by a two-tailed test with a significance level of 0.05 between PTBO and other algorithms are given in Table 12.

**Table 12.** Wilcoxon signed-rank test of 28 functions.

PTBO vs.	R <sup>+</sup>	R <sup>−</sup>	p-Value
PSO	305	73	<b><math>2.69 \times 10^{-4}</math></b>
DE	215	163	$3.16 \times 10^{-1}$
BA	325	53	<b><math>1.19 \times 10^{-4}</math></b>
CS	368	10	<b><math>3.79 \times 10^{-6}</math></b>
BSO	340	38	<b><math>1.57 \times 10^{-4}</math></b>
WWO	359	19	<b><math>2.52 \times 10^{-5}</math></b>
WCA	320	58	<b><math>1.20 \times 10^{-3}</math></b>
SMS	360	18	<b><math>2.96 \times 10^{-5}</math></b>

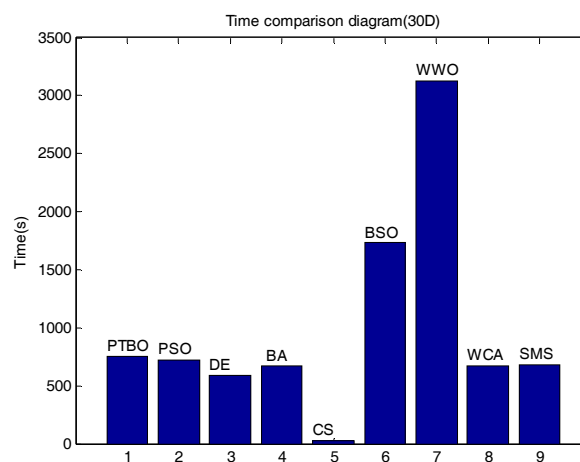
Bold text is the best result obtained by the compared algorithms.

From the results of the signed-rank test in Table 12, we can observe that our PTBO algorithm has a significant advantage over seven other algorithms in p-value, which are PSO, BA, CS, BSO, WWO, WCA, and SMS. Although PTBO has no significant advantage over DE, PTBO significantly outperforms DE in R<sup>+</sup> value. R<sup>+</sup> is the sum of ranks for the functions on which the first algorithm outperformed the second [46], and the differences are ranked according to their absolute values. So, according to the statistical results, it can be concluded that PTBO generally offered better performance than other algorithms for all 28 functions.

The above comparisons between PTBO and other nature-based meta-heuristic algorithms may offer a possible explanation for why PTBO could obtain better results on some optimization problems and that it is possible for PTBO to deal with more complex problems better.

#### 5.4.4. The Comparison Results of Time Complexity

The total comparisons of mean time complexity of the 28 functions about the nine algorithms are given in Figure 10. From Figure 10, we can observe that PTBO is ranked sixth, and is only better than BSO and WWO. However, we can find that the mean CPU time of PTBO is slightly worse than PSO and DE, and it also confirms the analysis in Section 4.1.



**Figure 10.** The mean central processing unit (CPU) time of the nine compared algorithms. 30D: 30 dimensions.

## 6. Conclusions

In this work, a new meta-heuristic optimization algorithm simulating the phase transition of elements in a natural system, named PTBO, has been described. Although the proposed PTBO algorithm is an algorithm with some similarities to the SMS and PSO algorithms, the main concepts

are slightly different. It is very simple and very flexible when compared to the existing nature-inspired algorithms. It is also very robust, at least for the test problems considered in this work. From the numerical simulation results and comparisons, it is concluded that PTBO can be used for solving uni-modal and multimodal numerical optimization problems, and is similarly effective and efficient compared with other optimization algorithms. It is worth noting that PTBO performs slightly worse than PSO and DE in time complexity. In the future, further research contents include (1) developing a more effective division method for the three phases, (2) combining the PTBO algorithm with other evolution algorithms, and (3) applying PTBO to real-world optimization problems, such as the reliability–redundancy allocation problem and structural engineering design optimization problems.

**Acknowledgments:** This research was partially supported by Industrial Science and technology project of Shaanxi Science and Technology Department (Grant No. 2016GY-088) and the Project of Department of Education Science Research of Shaanxi Province (Grant No. 17JK0371). Besides those, this research was partially supported by the fund of National Laboratory of Network and Detection Control (Grant No. GSYSJ2016007). The author declares that the above funding does not lead to any conflict of interests regarding the publication of this manuscript.

**Author Contributions:** Lei Wang conceived and designed the framework; Zijian Cao performed the experiments and wrote the paper.

**Conflicts of Interest:** The author declares that there is no conflict of interests regarding the publication of this paper.

## References

1. Yang, X.S. *Nature-Inspired Optimization Algorithms*; Elsevier: Amsterdam, The Netherlands, 2014.
2. Beyer, H.G.; Schwefel, H.P. Evolution strategies—A comprehensive introduction. *Nat. Comput.* **2002**, *1*, 3–52. [[CrossRef](#)]
3. Fogel, L.J.; Owens, A.J.; Walsh, M.J. *Artificial Intelligence through Simulated Evolution*; Wiley: New York, NY, USA, 1966.
4. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press, The MIT Press: London, UK, 1975.
5. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [[CrossRef](#)]
6. Garg, H. A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data. In *Handbook of Research on Artificial Intelligence Techniques and Algorithms*; IGI Global: Hershey, PA, USA, 2015; pp. 620–654.
7. Holland, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: London, UK, 1992.
8. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic strategy for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
9. Civicioglu, P. Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*, 8121–8144. [[CrossRef](#)]
10. Simon, D. Biogeography-based optimization. *IEEE Trans. Evolut. Comput.* **2008**, *12*, 702–713. [[CrossRef](#)]
11. Garg, H. An efficient biogeography based optimization algorithm for solving reliability optimization problems. *Swarm Evolut. Comput.* **2015**, *24*, 1–10. [[CrossRef](#)]
12. Civicioglu, P. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Comput. Geosci.* **2012**, *46*, 229–247. [[CrossRef](#)]
13. Rozenberg, G.; Bäck, T.; Kok, J.N. *Handbook of Natural Computing*; Springer: Berlin/Heidelberg, Germany, 2011.
14. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milano, Italy, 1992. (In Italian)
15. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks (ICNN), Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
16. Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst.* **2002**, *22*, 52–67. [[CrossRef](#)]

17. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [[CrossRef](#)]
18. Garg, H.; Rani, M.; Sharma, S.P. An efficient two phase approach for solving reliability–redundancy allocation problem using artificial bee colony technique. *Comput. Oper. Res.* **2013**, *40*, 2961–2969. [[CrossRef](#)]
19. Garg, H. Solving structural engineering design optimization problems using an artificial bee colony algorithm. *J. Ind. Manag. Optim.* **2014**, *10*, 777–794. [[CrossRef](#)]
20. He, S.; Wu, Q.H.; Saunders, J.R. Group search optimizer: An optimization algorithm inspired by animal searching behavior. *IEEE Trans. Evolut. Comput.* **2009**, *13*, 973–990. [[CrossRef](#)]
21. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, India, 9–11 December 2009; pp. 210–214.
22. Garg, H. An approach for solving constrained reliability-redundancy allocation problems using cuckoo search algorithm. *Beni-Suef Univ. J. Basic Appl. Sci.* **2015**, *4*, 14–25. [[CrossRef](#)]
23. Dai, C.H.; Chen, W.R.; Song, Y.H.; Zhu, Y.H. Seeker optimization algorithm: A novel stochastic search algorithm for global numerical optimization. *J. Syst. Eng. Electron.* **2011**, *21*, 300–311. [[CrossRef](#)]
24. Yang, X.S.; Gandomi, A.H. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* **2012**, *29*, 464–483. [[CrossRef](#)]
25. Gandomi, A. Bird mating optimizer: An optimization algorithm inspired by bird mating strategies. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 1213–1228.
26. Shi, Y.H. Brain storm optimization algorithm. In Proceedings of the Second International Conference of Swarm Intelligence, Chongqing, China, 12–15 June 2011; pp. 303–309.
27. Shi, Y.H. An optimization algorithm based on brainstorming process. *Int. J. Swarm Intell. Res.* **2011**, *2*, 35–62. [[CrossRef](#)]
28. Mirjalili, S. How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Appl. Intell.* **2015**, *43*, 150–161. [[CrossRef](#)]
29. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
30. Lee, K.S.; Geem, Z.W. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 3902–3933. [[CrossRef](#)]
31. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel meta-heuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110*, 151–166. [[CrossRef](#)]
32. Shah-Hosseini, H. The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. *Int. J. Bio-Inspired Comput.* **2009**, *1*, 71–79. [[CrossRef](#)]
33. Zheng, Y.J. Water wave optimization: A new nature-inspired meta-heuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [[CrossRef](#)]
34. Cuevas, E.; Echavarría, A.; Ramírez-Ortegón, M.A. An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Appl. Intell.* **2014**, *40*, 256–272. [[CrossRef](#)]
35. Kirkpatrick, S.; Gelatt, J.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)] [[PubMed](#)]
36. Granville, V.; Krivanek, M.; Rasson, J.P. Simulated annealing: A proof of convergence. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 652–656. [[CrossRef](#)]
37. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evolut. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
38. Suganthan, P.N.; Hansen, N.; Liang, J.J.; Deb, K.; Chen, Y.P.; Auger, A.; Tiwari, S. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL Rep.* **2005**, *2005*, 2005005.
39. Martin, O.C.; Monasson, R.; Zecchina, R. Statistical mechanics methods and phase transitions in optimization problems. *Theor. Comput. Sci.* **2001**, *265*, 3–67. [[CrossRef](#)]
40. Barbosa, V.C.; Ferreira, R.G. On the phase transitions of graph coloring and independent sets. *Phys. A Stat. Mech. Appl.* **2004**, *343*, 401–423. [[CrossRef](#)]
41. Mitchell, M.; Newman, M. Complex systems theory and evolution. In *Encyclopedia of Evolution*; Oxford University Press: New York, NY, USA, 2002; pp. 1–5.

42. Metastability. Available online: <https://en.wikipedia.org/wiki/Metastability> (accessed on 10 January 2017).
43. Sondheimer, E.H. The mean free path of electrons in metals. *Adv. Phys.* **2001**, *50*, 499–537. [[CrossRef](#)]
44. Liang, J.J.; Qu, B.; Suganthan, P.N.; Hernández-Díaz, A.G. *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization*; Technical Report, 201212; Computational Intelligence Laboratory, Zhengzhou University: Zhengzhou, China; Nanyang Technological University: Singapore, 2012.
45. Gibbons, J.D.; Chakraborti, S. *Nonparametric Statistical Inference*; Springer: Berlin/Heidelberg, Germany, 2011.
46. Garcí'a, S.; Molina, D.; Lozano, M.; Herrera, F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *J. Heuristics* **2009**, *15*, 617–644. [[CrossRef](#)]



© 2017 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 (<http://creativecommons.org/licenses/by/4.0/>).