

Research Article

Improved Glowworm Swarm Optimization Algorithm Based on a Sigmoid Function for the Absolute Value Equation

Feiran Wang ¹, Chen Chen ², Kai Chen ¹ and Liang Shen ²

¹College of Science, Guilin University of Aerospace Technology, Guilin, Guangxi 541004, China

²School of Management, Xuzhou Medical University, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Liang Shen; liang_shen@cumt.edu.cn

Received 4 May 2022; Accepted 31 July 2022; Published 20 September 2022

Academic Editor: Luca Reggiani

Copyright © 2022 Feiran Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Solving the absolute value equation (AVE) is a nondifferentiable NP-hard and continuous optimization problem with a wide range of applications. Because its solutions have different forms, it is challenging to design the most efficient algorithm that can solve different AVEs without using overcomplicated technical improvement and problem-dependent objectives. Hence, this paper proposed an improved glowworm swarm optimization (GSO) algorithm with an adaptive step size strategy based on the sigmoid function (SIGGSO) that solves the AVEs. Seven test AVEs, including multisolution and high-dimensional AVEs, are selected for testing and compared with seven metaheuristic algorithms. The experimental results show that the proposed SIGGSO algorithm has higher solution accuracy and stability when seeking multiple solution of AVEs compared to the basic GSO. Moreover, it obtains competitive advantages on multisolution and high-dimensional AVEs compared with other metaheuristic algorithms and provides an effective method for engineering and scientific calculations.

1. Introduction

The absolute value equation (AVE) is a nondifferentiable, NP-hard optimization problem in a continuous solution space. Many practical problems such as the site selection problem [1] and knapsack feasibility problem [2], which are nonlinear, nondifferentiable, multivariable, and multiparameter-complex optimization problems, are closely related to AVEs. AVEs are not only equivalent to standard and generalized linear complementarity problems but also to bilinear programming problems, which are typical mathematical programming problems with a wide range of applications in several disciplines. Compared to the other problems, the AVE has a simpler structure and easy to implement. Therefore, an in-depth study of AVEs is a good step towards solving the related problems as well.

Many algorithms have been proposed to solve the AVE. The results of traditional algorithms, such as the generalized Newton method [3], bilinear programming method [4], multivariate spectral gradient algorithm [5], and artificial neural networks (ANN) [6], have been reported. However,

the traditional algorithms struggle to deal with the objective functions which lack good analytical properties, e.g., continuity, differentiability, and smoothness.

With the vigorous development of metaheuristic algorithms, many scholars have attempted to utilize metaheuristic algorithms for practical problems. We refer to the following literatures for a few examples of applications. [7] improved the traditional particle swarm optimization (PSO) algorithm through introducing into the sharing information mechanism and the competition strategy called information sharing based PSO (IPSO). The novel algorithm IPSO has the similar rapid convergence speed and enhanced global search capability to the traditional PSO. The experimental results show that IPSO has better performance than the traditional PSO and the GA algorithm on benchmark functions, especially for difficult functions. [8] presented a chaotic monarch butterfly optimization (CMBO) algorithm to solve large-scale 0–1 knapsack problem. They introduce twelve well-known one-dimensional chaotic maps to tune the parameters of CMBO. Additionally, Gaussian mutation is used to perturb small part of solutions with worse fitness.

The proposed CMBO can outperform the standard MBO and other eight state-of-the-art canonical algorithms. [9] presents two binary variants of a Hunger Games Search Optimization (HGSO) algorithm based on V- and S-shaped transfer functions (BHGSO-V and BHGSO-S) within a wrapper FS model for choosing the best features from a large dataset. The experimental results demonstrate that the BHGSO-V algorithm can reduce dimensionality and choose the most helpful features for classification problems. For feature selection problem, [10] proposed the island algorithm (IA) with a Gaussian mutation strategy (IAGM) to find the optimal feature subset in the set of feature subsets. The new variant of IA can efficiently address the problem that as the number of iterations increasing, the island algorithm tends to local optimization. Colony predation algorithm (CPA) is a recently proposed algorithm which already is applied in some areas. For example, [11] proposed a framework within CPA called colony predation algorithm (CPA) with a kernel extreme learning machine (KELM), abbreviated as ECPA-KELM. The framework leads an efficient intelligence method for the diagnosis of COVID-19 from the perspective of biochemical indexes. The statistical analysis results show that ECPA-KELM can be used to discriminate and classify the severity of the COVID-19 as a possible computer-aided method and provide early warning for the treatment and diagnosis of COVID-19, effectively. [12] first proposed a novel metaheuristic algorithm called Harris hawks optimization (HHO) algorithm. The main inspiration of HHO is the cooperative behavior and chasing style of Harris' hawks in nature called surprise pounce. The founders do extensive experiments, and the statistical results and comparisons show that the HHO algorithm provides very promising and occasionally competitive results compared to well-established metaheuristic techniques. Since HHO came up to us, it has been widely used in many areas among which [13] proposed a novel satellite image segmentation technique based on dynamic HHO with a mutation mechanism (DHHO/M) and thresholding technique. Compared with the original Harris hawks optimization (HHO), the dynamic control parameter strategy and mutation operator used in DHHO/M can avoid falling into the local optimum and efficiently enhance the search capability. Experiments on various satellite images illustrate that the DHHO/M is superior to others in the following three aspects: fitness function evaluation, image segmentation effect, and statistical tests.

For solving AVEs, several metaheuristic algorithms including the genetic algorithm (GA), particle swarm optimization (PSO) algorithm, differential evolution (DE) algorithm, and harmony search (HS) algorithm are applied. The authors have studied the optimum correction of the absolute value equation by using GA [14] and obtained some computational results which prove the effectiveness of the algorithm. They did not give sufficient examples, instead only verifying the case of high-dimensional AVEs. PSO with exponentially decreasing inertia weight (EDIW) [15] has been deeply discussed. By employing dynamic changes of the inertia weight, PSO can easily escape from a local optimum and progress towards a global optimum. The authors gave a series of test AVEs with a unique solution or multiple

solutions to verify the effectiveness of EDIWPSO. However, a few test examples without high-dimensional AVEs are contained in the numerical experiments section. Despite this fact, the problem is that even for AVEs with multiple solutions, the EDIWPSO can only capture the multiple solutions of the AVE by running the algorithm several times. An improved adaptive differential evolution (IADE) algorithm [16] was proposed for solving AVEs. The algorithm combined global search ability and local search ability, using a quadratic adaptive mutation operation and crossover operation. Numerical results show that the improved algorithm can quickly find solutions of the test AVEs. However, it faces the same issue of insufficient test examples and comparisons. An improved harmony search algorithm with chaos (HSCH) [17] has been applied for solving AVEs, and this improved algorithm has a better optimization capability than the basic harmony search algorithm (HS) and the improved harmony search algorithm with differential mutation operator (HSDE). The authors verified the performance of HSCH by using three test AVEs, but their test still lacks any higher-dimensional AVEs, AVEs with multiple solutions, and comparisons with other metaheuristic algorithms.

Although many metaheuristic algorithms have been employed to solve optimization problems containing AVEs, existing metaheuristic algorithms still provide low precision solutions to AVEs and lack the ability to straightforwardly attain multiple solutions of AVEs. Therefore, the above algorithms constructed to solve AVEs lack the strong generalization ability required to work well. Hence, to address different forms of AVEs effectively, finding and optimizing an efficient algorithm possessing powerful universality is a critical issue.

Compared to the above algorithms for solving AVEs, the glowworm swarm optimization (GSO) algorithm has a natural advantage when solving multimodal optimization problems. Although the GSO algorithm shares certain characteristics with other metaheuristic algorithms compared in this study, there are several differences. GSO can simultaneously detect the multiple peaks of multimodal functions in parallel. This problem cannot be solved directly by the original version of the compared metaheuristic algorithms. Generally, the compared metaheuristic algorithms are used to find one global optima of the optimization problem. However, for solving AVEs, multiple solutions are generally obtained because of the existence of the absolute value vector $|x|$, and one important consideration of the algorithms solving AVEs is to locate as many solutions as possible. This capability is what separates GSO from the other metaheuristic intelligent algorithms. This is the motivation of improving the basic GSO and designing SIGGSO to solve AVE problems. Furthermore, GSO is not subject to conditions such as individual failures, the addition of noise, and the limitation of differentiable functions and multimodal functions, which may cause GSO to lose its searchability [18]. These advantages render GSO more adaptable to practical application problems, and practical optimization problems such as AVEs, which are multimodal functions under certain conditions, can be solved. However, because the basic GSO has troubles leaving local optima, we need to modify it to improve its ability to solve multisolution and high-dimensional AVEs. We illustrate this in Section 5.

Notably, the present scrutiny attempts to address a clear scientific gap with the following contributions:

- (i) Based on that sigmoid function can show a good balance between linear and nonlinear, an adaptive step size strategy derived from sigmoid function is designed and applied to GSO in this paper
- (ii) The introduction of this adaptive step size strategy can make GSO possess a strong ability to jump out local optima, compared with the basic GSO with fixed step size strategy
- (iii) The proposed improved GSO outperforms basic GSO in solving several test AVEs and obtains competitive advantages on multisolution and high-dimensional AVEs compared with other metaheuristic algorithms like PSO, GA, HHO, DE, and HS. This improved GSO can provide an effective method for engineering and scientific calculations

The rest of the paper is organized as follows: in Section 2, background information and the existence and uniqueness theory of AVE's solutions are provided. Section 3 describes the basic GSO. Section 4 describes the proposed SIGGSO in detail. The experimental design is presented in Section 5. In Section 6, conclusions and future research work are provided.

2. Absolute Value Equation

The general form of the AVE is $Ax - |x| = b$, where $A \in \mathbb{R}^{n \times n}$, $x, b \in \mathbb{R}^n$, and $|x|$ denote the absolute value of each component of x . It is an important subclass of the absolute value matrix equation $Ax + B|x| = b$ [19]. Subsequently, Mangasarian and Meyer [4] reported certain theoretical results on the existence and uniqueness of AVE solutions, which are listed below:

Lemma 1. *If $A \in \mathbb{R}^{n \times n}$ and all the singular values of A exceed one, then there exists a unique solution for the AVE for any $b \in \mathbb{R}^n$.*

Lemma 2. *For $A \in \mathbb{R}^{n \times n}$, if $\|A^{-1}\| \leq 1$, for any $b \in \mathbb{R}^n$, then there exists a unique solution for the AVE.*

Lemma 3. *If $b < 0$ and $\|A\|_\infty \leq \kappa/2$, where $\kappa = \min_i |b_i|/m a_{x_i} |b_i|$, then there exist 2^n nonidentical solutions for the AVE; these solutions have a different sign pattern and no zero component.*

For example, when $n = 2$, there are four solutions for the AVE with sign patterns of $\begin{pmatrix} + \\ + \end{pmatrix} \begin{pmatrix} + \\ - \end{pmatrix} \begin{pmatrix} - \\ + \end{pmatrix} \begin{pmatrix} - \\ - \end{pmatrix}$, respectively.

Since the GSO we consider is generally suitable for solving unconstrained optimization problems, we transform the AVE into an unconstrained optimization problem as follows:

Theorem 4. *The AVE can be equivalently transformed into the following unconstrained optimization problem:*

$$\min_{x \in \mathbb{R}^n} \Psi(x) = \frac{1}{2} \|f, (x)\|^2, \quad (1)$$

where $f(x) = Ax - |x| - b$, $\|x\|$ represents the Euclidean norm, $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}_+$, and if the optimal value of $\Psi(x)$ is zero, then x is the solution of AVE.

Proof. Suppose that x^* is the optimal solution of $\Psi(x) = 1/2 \|f, (x)\|^2$; according to the positive definiteness of the Euclidean norm,

$$\Psi(x) = \frac{1}{2} \|f, (x^*)\|^2 = 0 \Leftrightarrow f(x^*) = 0 \Leftrightarrow Ax^* - |x^*| - b = 0. \quad (2)$$

□

Hence, the solution x^* of the AVE is equivalent to the optimal solution x^* of $\Psi(x) = 1/2 \|f, (x)\|^2$ when $\Psi(x)$ reach the optimal value zero.

3. Basic GSO

3.1. Basic Concepts of GSO. In 2005, Krishnanand and Ghose [20] proposed a new swarm intelligence optimization algorithm called the artificial glowworm swarm optimization (GSO) algorithm. After several years of development, the GSO has good prospects for application to optimization in the scheduling of certain tasks, vehicle routing problems, and building design [21–23].

Each iteration of the GSO execution process includes five phases: the glowworm deployment phase, luciferin update phase, movement probability calculation phase, location update phase, and neighborhood range update phase, which are described below.

3.2. Mathematical Model of GSO. In this section, we introduce the mathematical model of GSO proposed by Krishnanand and Ghose [20]. For the reader's convenience, we first provide a list of the notation we will use, shown in Table 1.

3.2.1. Glowworm Deployment Phase. n glowworms (solution vectors) are randomly placed in the feasible domain of the problem and labeled x_1, x_2, \dots, x_n , respectively, where each glowworm is considered to be an m -dimensional vector $x_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$, $i = 1, 2, \dots, n$. Each glowworm is initialized with luciferin level l_0 , local decision radius r_0 , step size s , threshold value of the number of glowworms contained in the local decision domain n_t , luciferin decay factor ρ , fitness enhancement factor γ , domain change rate β , sensor range of the glowworms r_s (threshold of the local decision domain), and iteration number M .

3.2.2. Luciferin Update Phase. The luciferin level of each glowworm is equal to that of the previous iteration plus a certain extracted proportion of the glowworm's current

TABLE 1: Symbolic description.

Symbol	Description
$x_i(t)$	The location of glowworm i at iteration t (solution vectors)
n	The number of glowworms
m	Dimension of solution
l_0	Initial luciferin level
$l_i(t)$	The luciferin level of glowworm i at iteration t
r_0	Local decision radius
s	Fixed step size
s_0	Initial step size of adaptive step size model
n_t	Threshold value of the number of glowworms
ρ	Luciferin decay factor
γ	Fitness enhancement factor
β	Domain change rate
r_s	Sensor range of the glowworms
$J(x_i, t)$	The fitness of glowworm i at iteration t
$N_i(t)$	The set of all neighboring glowworms of glowworm i at iteration t
$r_d^i(t)$	Local decision domain of glowworm i at iteration t
M	Iteration number

fitness, minus a certain proportion of the luciferin level that varies with time, as described by the following equation:

$$l_i(t,+,1) = (1,-,\rho)l_i(t) + \gamma J(x_i, (t,+,1)), \quad (3)$$

where $l_i(t)$ is the luciferin level at iteration t and $J(x_i, (t))$ indicates the fitness of glowworm i at iteration t , i.e., the corresponding value of the objective function.

3.2.3. Movement Probability-Calculating Phase. Each glowworm decides its movement direction according to the luciferin level of the glowworms in its local decision domain. For each glowworm i , the probability of moving toward a neighbor j is given by

$$P_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}, \quad (4)$$

where $j \in N_i(t)$, $N_i(t) = \{j, : , \|x_j, (t), -, x_i, (t)\|, <, r_d^i, (t)\}, ;, l_p, (t), <, l_p, (t)\}$ denotes the set of all neighboring glowworms of glowworm i at iteration t , $r_d^i(t)$ denotes the adaptive local decision domain of glowworm i at iteration t , and $0 \leq r_d^i(t) \leq r_s$.

3.2.4. Location Update Phase. Glowworm i select a glowworm $j \in N_i(t)$ with $P_{ij}(t)$ given by (4) and perform a location update. Glowworm i moves with a certain step size toward glowworm j with the maximum luciferin level in its

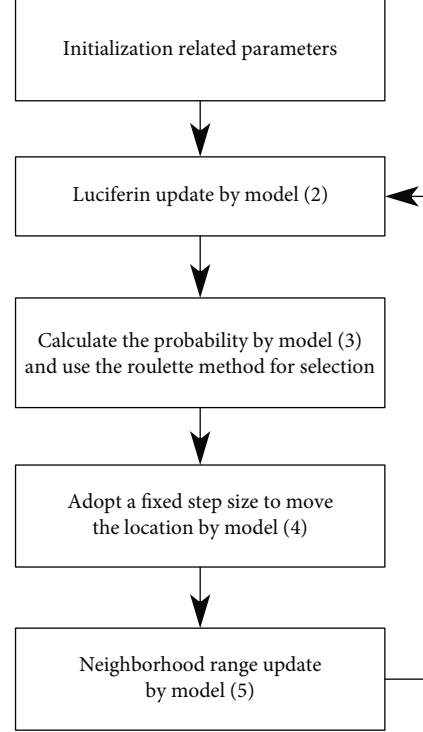


FIGURE 1: Flowchart of the GSO algorithm.

local decision domain; then, the movement process can be represented as

$$x_i(t,+,1) = x_i(t) + s \frac{x_j(t) - x_i(t)}{\|x_j, (t), -, x_i, (t)\|}. \quad (5)$$

3.2.5. Neighborhood Range Update Phase. Each glowworm uses an adaptive local decision radius, which changes at each iteration according to the number of neighboring glowworms (the local decision radius increases when the number of neighbors is smaller, and vice versa). At each iteration, the following rule is applied:

$$r_d^i(t,+,1) = \min \{r_s, , , m, a, x, \{0, , , r_d^i, (t), +, \beta, (n_t, -, |N_i, (t)|)\}\}. \quad (6)$$

A variant of neighborhood range update rule was first introduced in [18]. Its mathematical model is as follows:

$$r_d^i(t,+,1) = \frac{r_s}{1 + \beta D_i(t)}, \quad (7)$$

where $D_i(t) = N_i(t)/\pi r_s^2$ is the glowworm density in the local decision domain of glowworm i at iteration t and β is a constant representing the domain change rate.

The GSO flowchart is shown in Figure 1.

TABLE 2: Experimental results for AVE1 at various γ and ε setting.

	Metrics	$\lambda = 3$	$\lambda = 15$	$\lambda = 30$	$\lambda = 40$	$\lambda = 50$
$\varepsilon = 0.1$	Mean	2.09E-01	1.84E-10	7.96E-22	0	0
	Std	2.20E-02	2.21E-20	3.14E-43	0	0
$\varepsilon = 0.3$	Mean	5.16E-01	1.24E-07	1.39E-16	1.23E-22	4.54E-29
	Std	8.89E-02	4.74E-15	8.31E-33	8.18E-45	1.55E-56
$\varepsilon = 0.5$	Mean	1.54E+00	3.13E-05	1.97E-11	8.23E-16	6.50E-20
	Std	1.19E+00	5.95E-10	2.13E-22	2.56E-31	1.77E-39
$\varepsilon = 0.7$	Mean	3.27E+00	1.92E-02	3.31E-06	7.26E-09	3.28E-11
	Std	4.75E+00	1.38E-04	6.06E-12	2.19E-17	7.07E-22
$\varepsilon = 1$	Mean	1.08E+01	1.94E+01	2.34E+01	2.28E+01	2.54E+01
	Std	3.98E+01	1.25E+02	3.32E+02	3.88E+02	3.22E+02

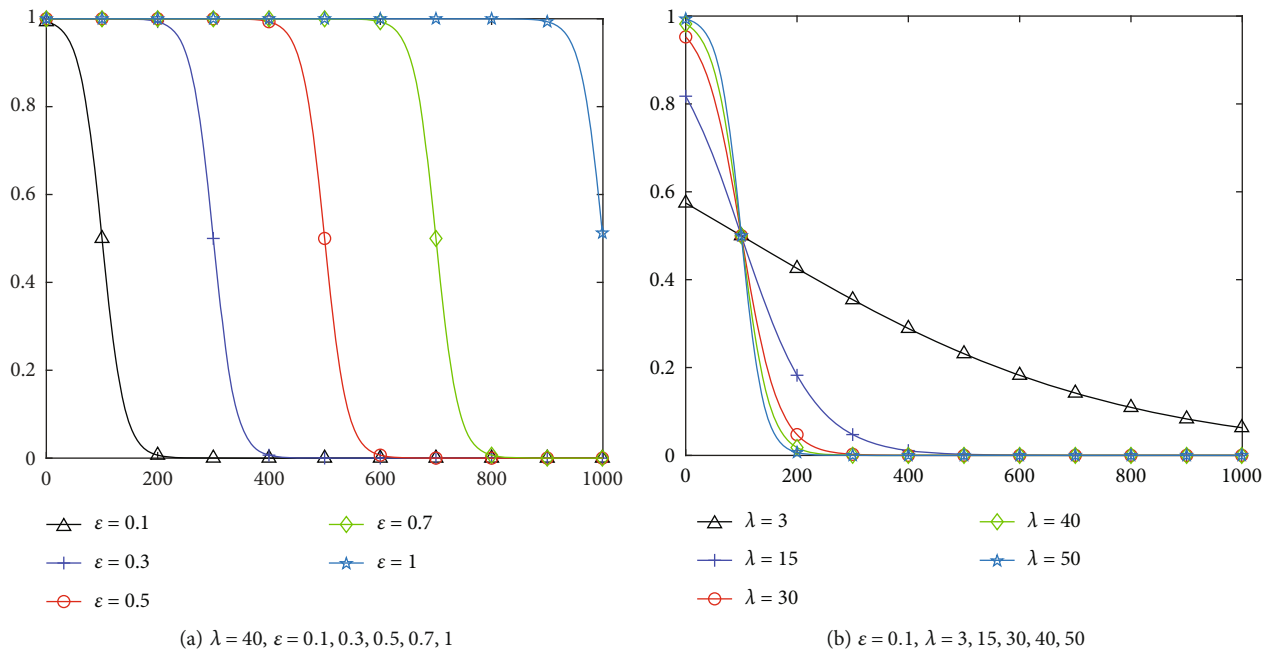


FIGURE 2: Curve of strategy (8).

4. Improved GSO Based on the Sigmoid Function

4.1. Basic Idea for GSO Improvement. In a neural network (NN), the sigmoid function is derived from the activation function used for limiting the output amplitude of a neuron. As it suppresses the output signal to a permissible range, it is also known as the suppression function. We refer to [24–26] for some application examples of sigmoid function. In general, the normal output range of a neuron can be expressed as a unit closed interval $[0, , 1]$. The value range of the sigmoid function is a continuous interval from 0-1 and is differentiable. It is for this reason that the sigmoid function has a wide range of applications in neural networks as a suppression function.

The sigmoid function is a strictly increasing function which shows better balance between linear and nonlinear

functions. An example of the sigmoid function is the logistic function:

$$\varphi(x) = \frac{1}{1 + e^{-ax}}, \quad (8)$$

where a is the tilt parameter, which can be modified to change the degree of tilt.

To utilize the sigmoid function to construct an adaptively decreasing step size, we need to modify the sigmoid function such that it is strictly monotonically decreasing, and the function value remains within the $[0, , 1]$ range. Therefore, we set an initial large step size s_0 during GSO execution and then multiply s_0 by the constructed function. Based on this analytical approach, we make the following changes to the sigmoid function:

Adaptive variable step-size GSO based on the Sigmoid function

Set the dimension of solution of AVE as m , number of glowworms as n , maximum iteration number as T_{\max} , $l_0, r_0, r_s, n_t, \rho, \gamma, \beta, s_0$,

$t = 1$

Let $x_i(t)$ be the location of glowworm i at time t ;

for $i = 1$ to n **do**

$l_i(0) = l_0$

$r_d^i(0) = r_0$;

while $(t \leq T_{\max})$ **do**: %Main loop

{

for $i = 1$ to n **do**: % Luciferin update phase

$l_i(t, +1) = \max \{0, (1 - \rho) l_i(t) + \gamma J_i(x_i(t, +1))\}$;

for $i = 1$ to n **do**: %Move phase

{

$N_i(t) = \{j, : \|x_j(t) - x_i(t)\| < r_d^i(t), l_j(t) < l_i(t)\}$;

for each glowworm $j \in N_i(t)$ **do**:

$P_{ij}(t) = l_j(t) - l_i(t) / \sum_{k \in N_i(t)} l_k(t) - l_i(t)$;

$j = \text{select_glowworm}(\vec{P})$; % Use the roulette method

$s(t) = s_0 (1 - 1 / (1 + \exp(-50(t/T_{\max} - 0.1))))$;

$x_i(t, +1) = x_i(t) + s(t)(x_j(t) - x_i(t) / \|x_j(t) - x_i(t)\|)$;

$r_d^i(t, +1) = r_s / 1 + \beta D_i(t)$;

}

$t \leftarrow t + 1$

}

PSEUDOCODE 1: The pseudocode of SIGGSO algorithm.

$$\varphi(t) = 1 - \frac{1}{1 + \exp(-\lambda, ((t/T_{\max}) - \varepsilon))}, \quad (9)$$

where λ and ε are undetermined parameters. Therefore, the fixed step size location update model (5) is changed into an adaptive variable step size model, as shown below:

$$x_i(t, +1) = x_i(t) + s_0 \cdot \varphi(t) \cdot \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|}. \quad (10)$$

As detailed in the numerical experiments (Section 5), we establish that performance of the GSO, based on the adaptively decreasing step size strategy (9), for solving AVEs is good. We call this improved GSO with adaptive variable step size model (10) SIGGSO, which comprises models (3), (4), (10), and (7).

To ensure that the improved adaptively decreasing step size model (10) can be better applied to the GSO for solving AVEs, the selection of the values of parameters λ and ε is critical.

Here, we consider AVE1 given in Section 5 for testing the set values of λ and ε . The mean value and standard deviation of the fitness when the optimal solution is obtained are listed in Table 2 for different values of λ and ε . The curve with different values of λ and ε of strategy (9) is displayed in Figure 2.

Table 2 shows that when ε is fixed, the results improve as λ is increased. When $\lambda = 50$, SIGGSO solves AVE1 with

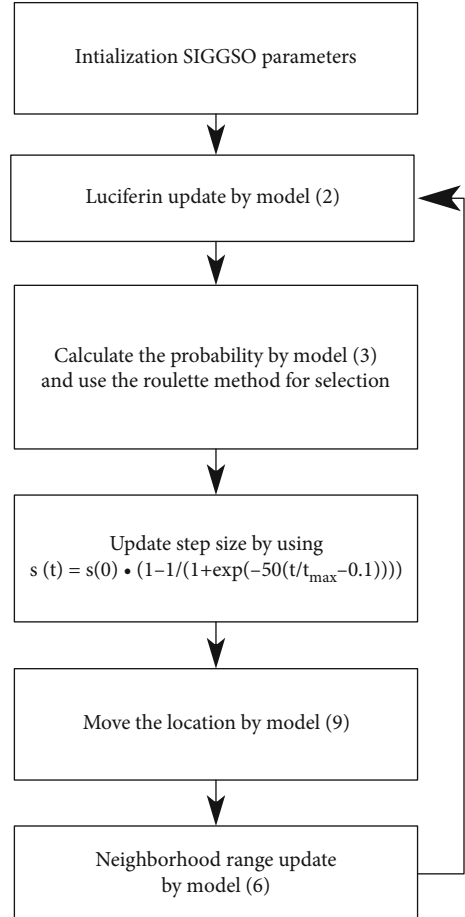


FIGURE 3: Flowchart of the SIGGSO algorithm.

TABLE 3: Time complexity analysis for population size n .

Phase	Time complexity	Max iteration number
Initialization	$O(n)$	n
Luciferin update phase	$O(T_{\max}, n)$	$T_{\max} \cdot n$
Move phase	$O(T_{\max}, n, l, e, n, g, t, h, (N_i, (t)))$	$T_{\max} \cdot n \cdot \text{length}(N_i, (t))$

TABLE 4: Time complexity analysis for m -dimensional AVEs.

Phase	Time complexity	Max iteration number
Initialize population locations	$O(n, m)$	$n \cdot m$
Luciferin update phase	$O(T_{\max}, n)$	$T_{\max} \cdot n$
Move phase	$O(T_{\max}, n, m)$	$T_{\max} \cdot n \cdot m$

TABLE 5: Compared metaheuristic algorithms.

Algorithm	Description
GSO	Basic glowworm swarm optimization algorithm [20].
SIGGSO	Improved glowworm swarm optimization algorithm proposed in this study for solving AVEs.
IADE	Improved adaptive differential evolution algorithm [16].
EDIWPSO	Particle swarm optimization algorithm with exponentially decreasing inertia weight [15].
HSCH	Improved harmony search algorithm with chaos [17].
GA	Genetic algorithm [14].
HHO	Harris hawks optimization algorithm [12].

very high accuracy, the minimum fitness is zero, and the algorithm is stable. When λ is fixed, the experimental results improve as ε is decreased. When $\varepsilon = 0.1$, SIGGSO solves AVE1 with very high accuracy, the minimum fitness is zero, and the algorithm is stable. From Figures 2(a) and 2(b), as λ increases, ε decreases, and the change trend of $\varphi(t)$ with t is first rapid decline and then gradual stabilization. This is also in line with our basic idea for adaptive step size improvement, namely, that during the early the iterations, the step size can exhibit a relatively large change, such that the glowworms can rapidly converge near the solutions, whereas in the later stages, the step size can exhibit a relatively small change, such that the glowworms can be fine-tuned near the solutions, thus rendering the solution more accurate. Based on this analysis, we fix $\lambda = 50$ and $\varepsilon = 1$ for the SIGGSO in the numerical experiments. We use the following model to update the locations of the glowworms:

$$\begin{cases} x_i(t, +1) = x_i(t) + s_0 \cdot \varphi(t) \cdot \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|}, \\ s(t) = 1 - \frac{1}{(1, +, e, x, p, (-, 50, (t/t_{\max}, -, 0, , 1)))}. \end{cases} \quad (11)$$

TABLE 6: Fixed parameter setting for the improved GSOs.

l_0	n_i	ρ	γ	β	s_0
5	5	0.4	0.6	0.08	0.2

TABLE 7: Variable parameter setting for the improved GSOs.

AVE	r_0	r_s
Low-dimensional AVEs	5	5
High-dimensional AVEs	20	20

The pseudocode of SIGGSO algorithm is depicted in Pseudocode 1, and the flowchart of SIGGSO is shown in Figure 3.

4.2. Time Complexity Analysis of SIGGSO. In this section, we will use big O notation for time complexity analysis.

4.2.1. Time Complexity Analysis for Population Size n . Pseudocode 1 shows that the SIGGSO algorithm has an outer loop and two inner loops, and the second inner loop has another inner loop.

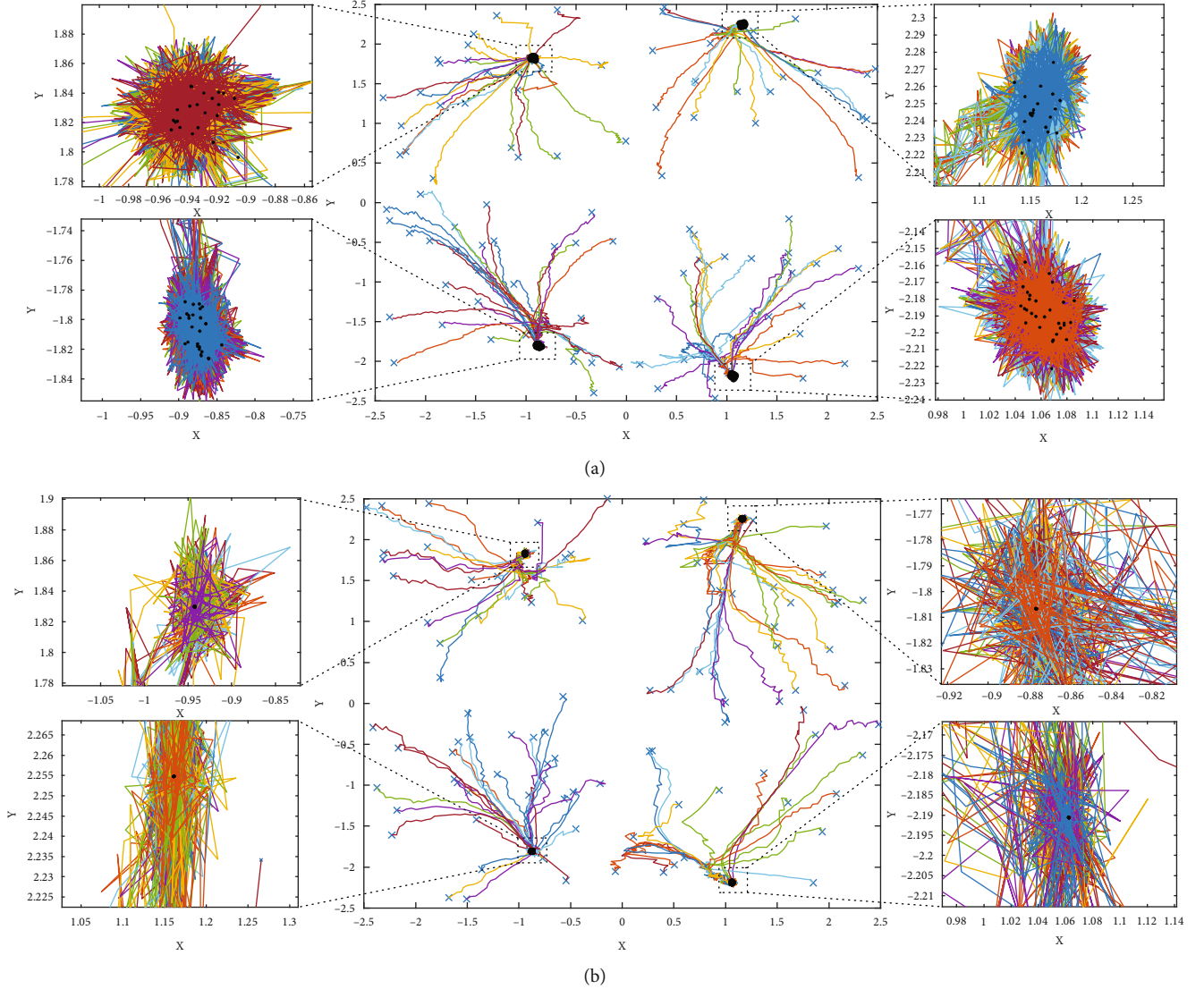


FIGURE 4: Process of capturing the multiple peaks of AVE2 through (a) GSO and (b) SIGGSO.

Remark 5. $a \cdot b$ represents the product of two scalar a and b .

Table 3 shows that the total time complexity of SIGGSO is

$$T(n) = O(n) + O(T_{\max}, \cdot, n) + O(T_{\max}, \cdot, n, \cdot, l, e, n, g, t, h, (N_i, (t))). \quad (12)$$

According to the addition rule, only the highest order of the time complexity is considered, and according to the multiplication rule, $O(c, n)$ is equivalent to $O(n)$ (c is constant). Therefore, the time complexity of SIGGSO is $O(n, \cdot, l, e, n, g, t, h, (N_i, (t)))$. As $\text{length}(N_i, (t))$ changes for each glowworm at each t , its range is constant $\leq \text{length}(N_i, (t)) \leq n$. Finally, $T(n)$ is between $O(n)$ and $O(n^2)$.

4.2.2. Time Complexity Analysis for m -Dimensional AVEs. We assume that both the maximum iteration number n

and T_{\max} are constants that are set when analyzing the time complexity for the AVE dimension.

From Table 4, we obtain $T(n) = O(n, \cdot, m) + O(T_{\max}, \cdot, n) + O(T_{\max}, \cdot, n, \cdot, m)$, as per the addition and multiplication rules, and $T(n) = O(m)$.

5. Numerical Experiments

In this section, we first give seven test AVEs containing multi-resolution and high-dimensional ones and have analyzed the solution characteristics of the test AVEs. Parameter setting details is depicted in Section 5.3. The primary objective that shows the comparing the results of SIGGSO, basic GSO and other metaheuristic algorithms are described in Sections 5.4 and 5.5. On the compared results, we mainly consider the mean and standard deviation of fitness value as comparing metrics. Furthermore, the well-known Wilcoxon signed-rank test method performs the significance difference test on the comparing results.

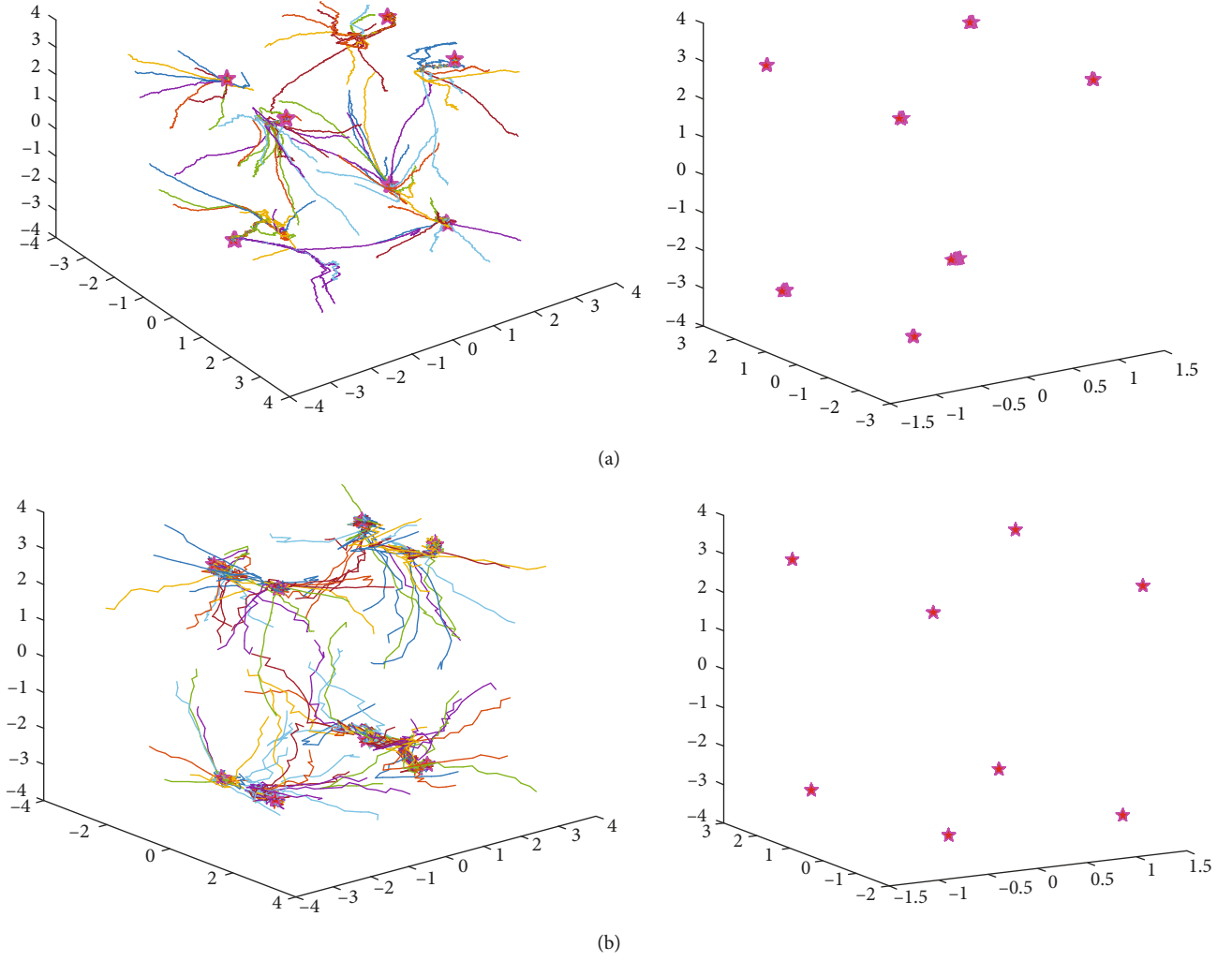


FIGURE 5: Process of capturing the multiple peaks of AVE3 through (a) GSO and (b) SIGGSO.

5.1. Test AVEs.

$$\text{AVE1} : A = \begin{pmatrix} 10 & 1 & 2 & 0 \\ 1 & 11 & 3 & 1 \\ 0 & 2 & 12 & 1 \\ 1 & 7 & 0 & 13 \end{pmatrix}, b = \begin{pmatrix} 12 \\ 15 \\ 14 \\ 20 \end{pmatrix},$$

$$\text{AVE2} : A = \begin{pmatrix} 0.1 & 0.02 \\ 0.2 & 0.01 \end{pmatrix}, b = \begin{pmatrix} -1 \\ -2 \end{pmatrix},$$

$$\text{AVE3} : A = \begin{pmatrix} 0.01 & 0.02 & 0.03 \\ 0.02 & 0.03 & 0.01 \\ 0.03 & 0.02 & 0.01 \end{pmatrix}, b = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}, \quad (13)$$

$$\text{AVE4} : A = [a_{ij}]_{m \times m}, \text{ when } i = j, a_{ij} = 500, \text{ when } i \neq j, a_{ij} = a_{ji} = 1 + \text{rand}, b = (A, -, I)e,$$

$$\text{AVE5} : A = R^T R + mI, b = (A, -, I)e, R = \text{rand}(m),$$

$$\text{AVE6} : A = R^T R + mI, b = (A, -, I)e, R = 10 \text{ rand}(m),$$

$$\text{AVE7} : A = [a_{ij}]_{m \times m}, a_{ii} = 4m, a_{i+1,i} = a_{i,i+1} = m (i \leq m-1), \text{ the rest } a_{ij} = 0.5, b = (A, -, I)e.$$

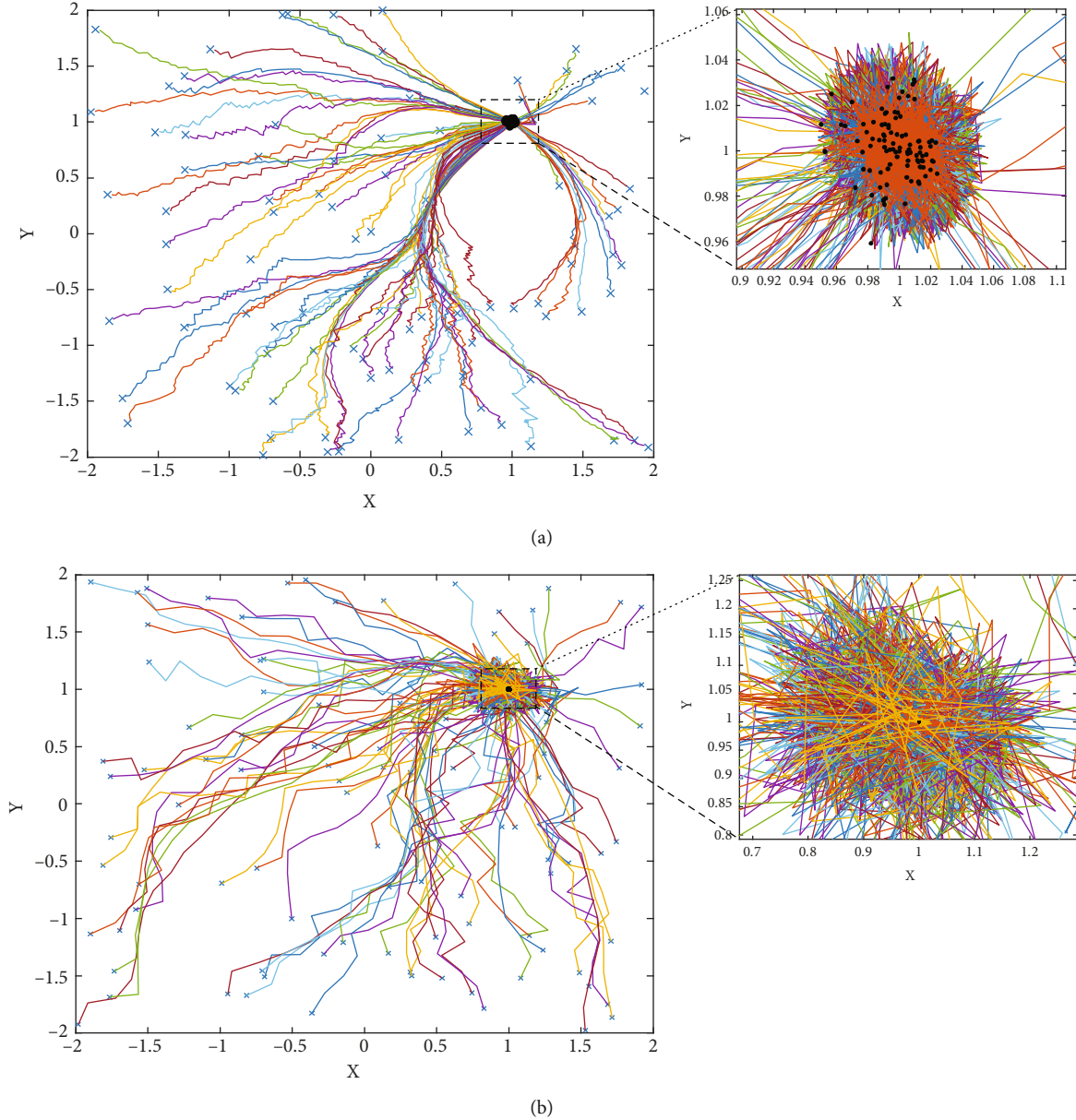


FIGURE 6: Process of capturing the single peak of AVE4 through (a) GSO and (b) SIGGSO.

In AVE4, AVE5, AVE6, and AVE7, m is the size of the problem, i.e., the dimension of the solution. rand is a random number that follows a uniform distribution in $[0,1]$, $\text{rand}(m)$ is an m -order square matrix whose elements are generated by rand , $(r, a, n, d)_{m \times 1}$ is m -dimensional column vector whose elements are generated by rand , I is an m -order identity matrix, and e is an m -dimensional column vector whose elements are all unity. R^T represents the transpose matrix of R .

The above AVEs were employed to test and compare the performances of SIGGSO with other metaheuristic algorithms which were programmed using MATLAB 2018a. All the experiments were performed on a PC configured with an Intel Core i5-7400 processor and 8-GB RAM. For a

quick impression, the compared algorithms are listed in Table 5.

5.2. Characteristics of the Test AVEs. For AVE1, AVE4, AVE5, and AVE6, we can easily obtain the singular values of the matrix A , which all exceed one. Hence, Lemma 1 tells us that both AVEs have unique solutions. The condition of Lemma 2, i.e., $\|A^{-1}\| \leq 1$, applies to A in AVE7; hence, a unique solution to AVE7 exists. A and b in AVE2 and AVE3 satisfy Lemma 3; hence, there exist 2^n solutions for these three AVEs. Furthermore, as AVE2 is a two-dimensional AVE, it has four solutions. The dimension of AVE3 is three; hence, it has eight solutions. Based on Lemma 3, the sign pattern of AVE2 is

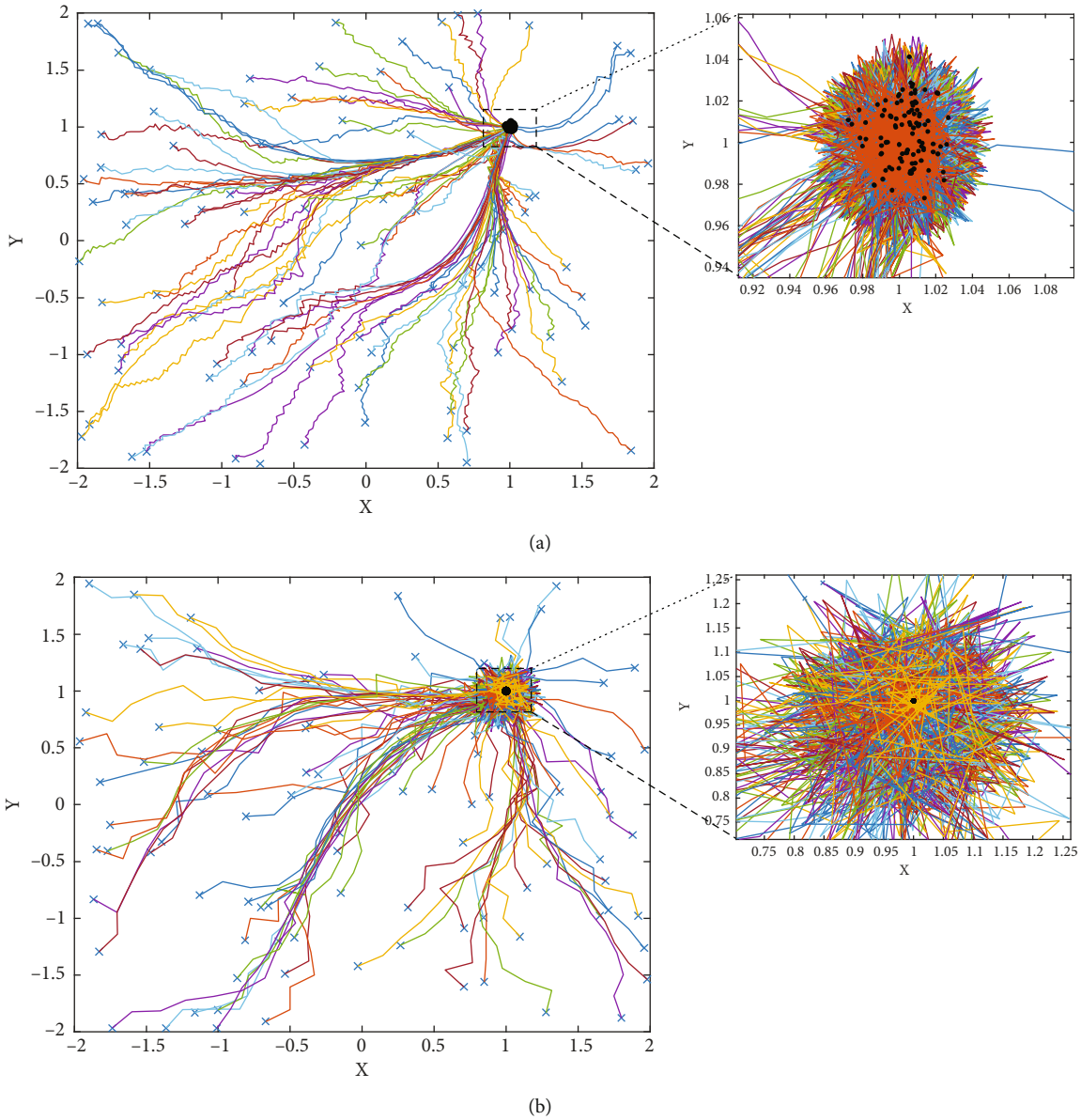


FIGURE 7: Process of capturing multiple peaks of AVE5 through (a) GSO and (b) SIGGSO.

$\begin{pmatrix} + \\ + \end{pmatrix} \begin{pmatrix} + \\ - \end{pmatrix} \begin{pmatrix} - \\ + \end{pmatrix} \begin{pmatrix} - \\ - \end{pmatrix}$. Analogously, the sign pattern of AVE3 is

$$\begin{pmatrix} - \\ - \\ + \end{pmatrix} \begin{pmatrix} - \\ + \\ + \end{pmatrix} \begin{pmatrix} + \\ - \\ + \end{pmatrix} \begin{pmatrix} + \\ - \\ - \end{pmatrix} \begin{pmatrix} - \\ - \\ - \end{pmatrix} \begin{pmatrix} - \\ + \\ - \end{pmatrix} \begin{pmatrix} + \\ + \\ - \end{pmatrix} \begin{pmatrix} + \\ + \\ + \end{pmatrix}. \quad (14)$$

5.3. *Parameter Setting.* Here are the parameter setting details of the compared algorithms:

- (a) For SIGGSO, the fixed parameter settings which are not specifically tuned for every AVEs are the same as those in Krishnanand and Ghose [27, 28]. We list them in Table 6. A full factorial analysis is carried out in Krishnanand and Ghose [18] to show that the choice of r_s has some influence on the performance of the algorithm, in terms of the total number of peaks captured, and they suggested that r_s is equal to r_0 . Based on extensive numerical experiments on test AVEs, we determined the appropriate values of r_s and r_0 , as shown in Table 7
- (b) For IADE, EDIWPSO, HSCH, HHO, and GA, all the parameter settings were the same as those in their referred papers (discussed in Section 1)

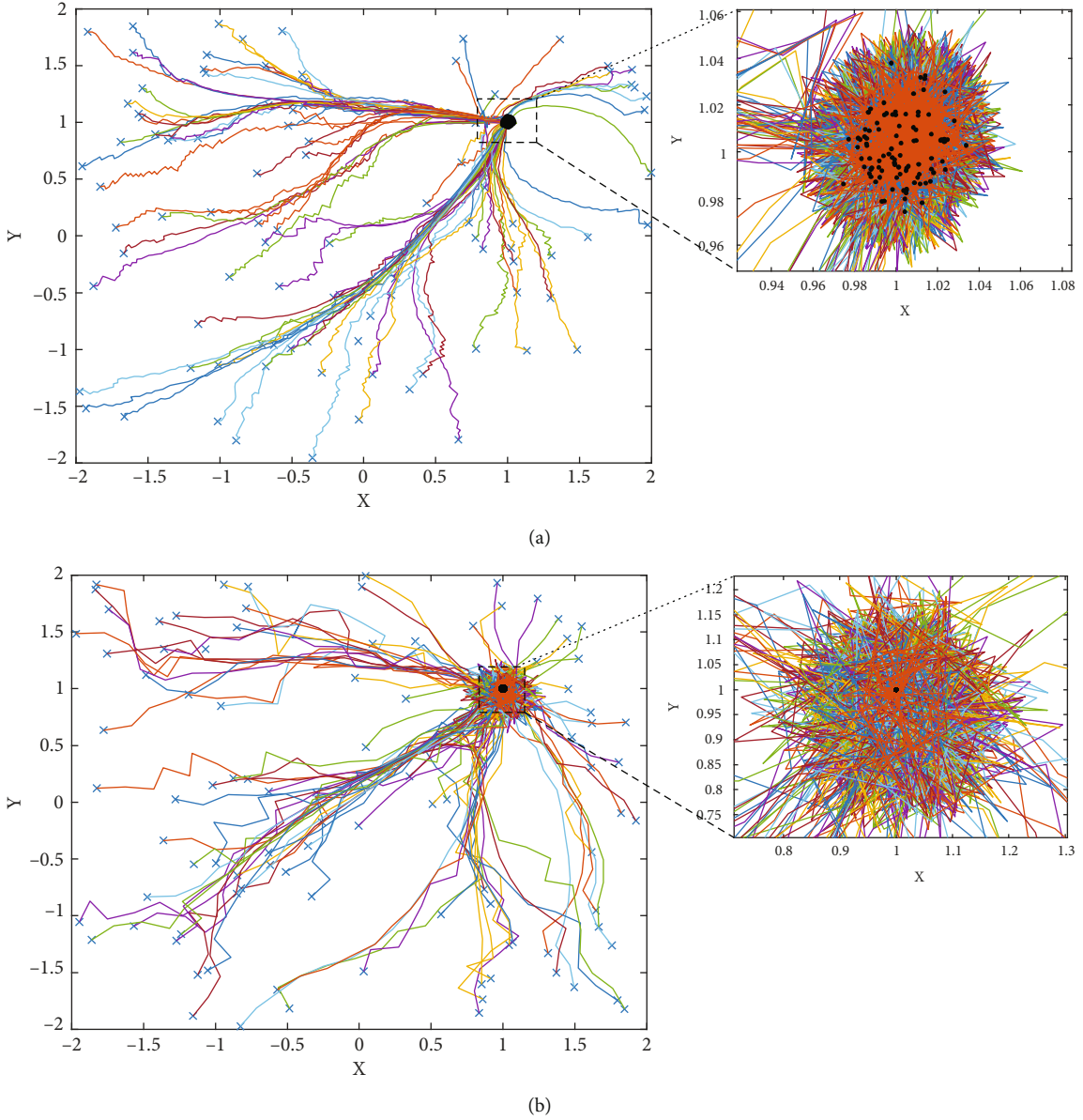


FIGURE 8: Process of capturing single peak of AVE6 through (a) GSO and (b) SIGGSO.

Remark 6. The low-dimensional AVEs are AVE1, AVE2, and AVE3. The high-dimensional AVEs are AVE4, AVE5, AVE6, and AVE7 whose dimension equal or exceed 100.

Remark 7. For all the algorithms compared in this study, the initial population was generated randomly in the solution space, the maximum number of iterations was 1000, and the population size was 50.

Remark 8. To give distinct expressions, AVE4, AVE5, AVE6, and AVE7 are denoted as “AVE name_dimension of solution.”

Remark 9. We consider $\Psi(x) = 1/2\|f(x)\|^2$ in Theorem 1 to be the fitness function of all test AVEs. The theoretical optimal values of the unconstrained optimization problem $\min \Psi(x) = 1/2\|f(x)\|^2$ are all zero.

5.4. Advantage of SIGGSO in Solving Multisolution AVEs. Through the analysis of the solution characteristics of AVE2 and AVE3 in Section 5.2, we obtained four solutions of AVE2 and eight solutions of AVE3 by executing SIGGSO. The four solutions of AVE2 are

$$\begin{pmatrix} 1.1612 \\ 2.2548 \end{pmatrix} \begin{pmatrix} 1.0624 \\ -2.1906 \end{pmatrix} \begin{pmatrix} -0.8762 \\ -1.8067 \end{pmatrix} \begin{pmatrix} -0.9424 \\ 1.8298 \end{pmatrix}. \quad (15)$$

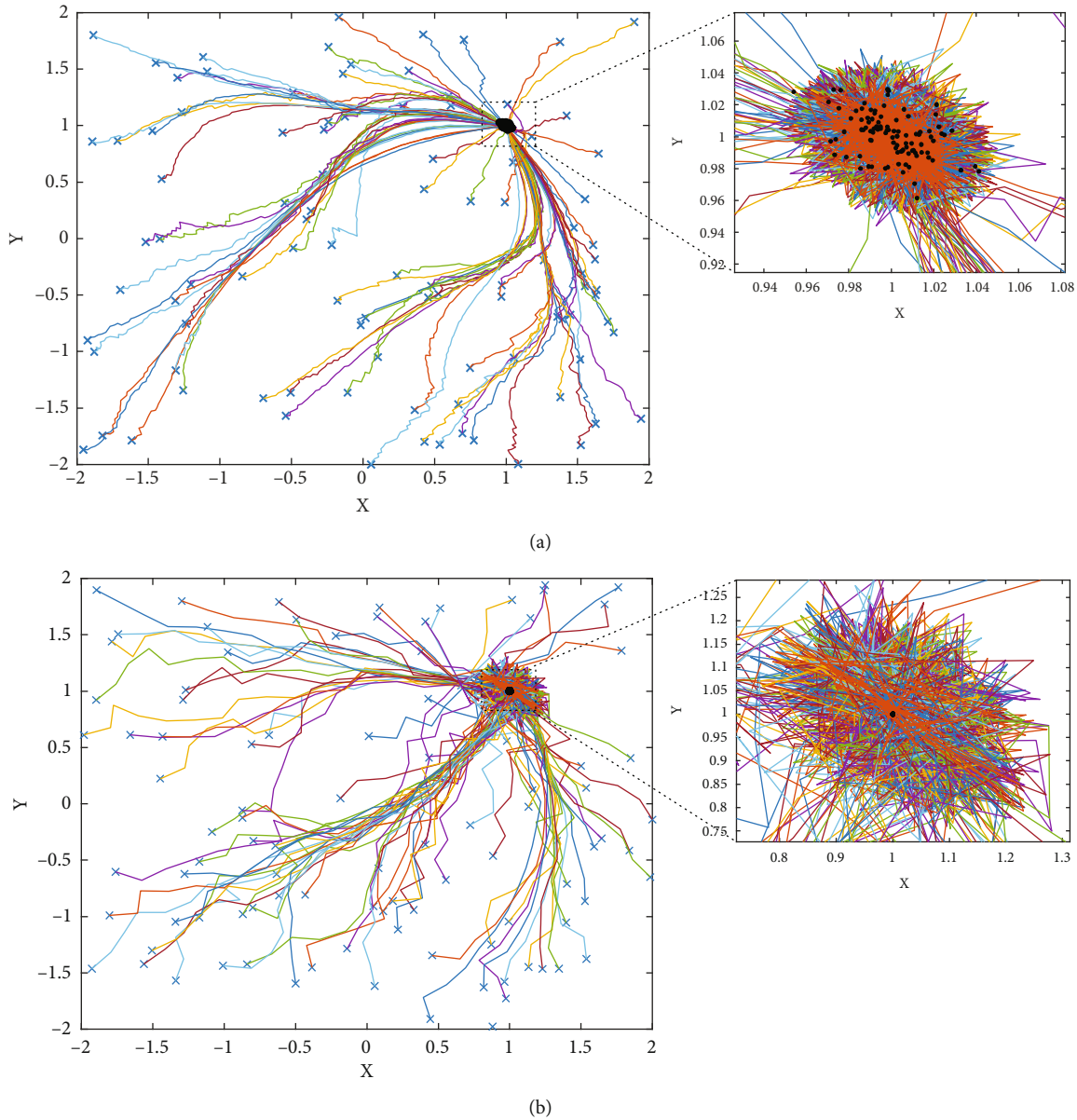


FIGURE 9: Process of capturing single peak of AVE7 through (a) GSO and (b) SIGGSO.

The eight solutions of AVE3 are

$$\begin{pmatrix} -1.0394 \\ -1.9503 \\ 2.9594 \end{pmatrix} \begin{pmatrix} -1.1213 \\ 2.0701 \\ 3.0381 \end{pmatrix} \begin{pmatrix} 1.0683 \\ -1.9961 \\ 3.0461 \end{pmatrix} \begin{pmatrix} 0.8815 \\ -1.9301 \\ -2.9583 \end{pmatrix} \begin{pmatrix} -0.8663 \\ -1.8967 \\ -2.9070 \end{pmatrix} \begin{pmatrix} -0.9412 \\ 2.0117 \\ -2.9822 \end{pmatrix} \begin{pmatrix} 0.9594 \\ 2.0507 \\ -3.0211 \end{pmatrix} \begin{pmatrix} 1.1470 \\ 2.0911 \\ 3.1078 \end{pmatrix}. \quad (16)$$

The following figures show the process of capturing the AVE solutions directly obtained by GSO and SIGGSO. To clearly demonstrate the location changing process of the glowworms, we adjusted certain parameters in this subsection. For all the AVEs used in the simulation, we set $n =$

100, $T_{\max} = 1000$. For AVE4, AVE5, AVE6, and AVE7, we set $m = 2$.

Figures 4–9 indicate that under the same number of iterations, for solving AVE2–AVE7, SIGGSO has a strong ability to leave the local optima and then converge to the global

TABLE 8: Mean and standard deviation of the fitness value obtained by the seven algorithms on the test AVEs.

AVE	Metrics	GSO	SIGSO	IADE	EDIWPSO	HSCH	HHO	GA
AVE1	Mean	3.82E-01	0	0	0	5.32E-04	3.77E-02	2.91E-03
	Std	2.05E-01	0	0	0	1.79E-04	2.45E-02	2.20E-03
AVE2	Mean	1.13E-03	1.94E-17	2.78E-17	2.78E-17	5.68E-06	1.00E-06	2.20E-06
	Std	6.66E-04	3.72E-17	2.85E-17	2.85E-17	3.33E-06	1.74E-06	4.02E-06
AVE3	Mean	1.14E-02	0	0	0	3.33E-05	1.63E-04	1.72E-05
	Std	1.87E-02	0	0	0	1.53E-05	2.68E-04	3.31E-05
AVE4_100	Mean	3.02E+02	2.04E+02	6.47E-01	6.05E+02	1.11E+03	2.68E+01	5.16E+02
	Std	1.83E+01	1.30E+01	4.65E-01	9.80E+01	1.93E+01	1.47E+01	6.23E+01
AVE4_200	Mean	4.18E+02	2.86E+02	5.98E+01	1.00E+03	1.77E+03	4.77E+01	1.26E+03
	Std	2.47E+01	1.30E+01	1.42E+01	2.01E+02	1.99E+01	3.27E+01	5.56E+01
AVE4_500	Mean	6.00E+02	4.62E+02	5.40E+02	1.73E+03	3.06E+03	1.51E+02	2.64E+03
	Std	2.46E+01	1.37E+01	3.58E+01	3.91E+02	2.92E+01	8.29E+01	4.19E+01
AVE5_100	Mean	7.90E+01	5.89E+01	1.24E+01	1.19E+02	2.62E+02	1.49E+02	1.89E+02
	Std	4.44E+00	5.40E+00	3.38E+00	2.38E+01	8.97E+00	3.88E+01	1.15E+01
AVE5_200	Mean	2.33E+02	1.75E+02	1.78E+02	3.38E+02	8.19E+02	5.61E+02	7.23E+02
	Std	1.80E+01	1.00E+01	1.19E+01	9.02E+01	1.40E+01	8.81E+01	2.35E+01
AVE5_500	Mean	9.58E+02	7.66E+02	1.50E+03	1.43E+03	3.48E+03	2.79E+03	3.27E+03
	Std	5.28E+01	3.78E+01	8.62E+01	2.56E+02	4.20E+01	2.85E+02	4.65E+01
AVE6_100	Mean	1.10E+03	6.46E+02	9.77E+02	1.27E+03	2.81E+03	3.12E+03	2.74E+03
	Std	1.61E+02	1.18E+02	7.18E+01	2.84E+02	1.15E+02	4.38E+02	2.17E+02
AVE6_200	Mean	2.92E+03	1.82E+03	3.82E+03	4.32E+03	8.85E+03	9.44E+03	8.79E+03
	Std	3.21E+02	2.15E+02	3.32E+02	9.29E+02	3.69E+02	9.45E+02	6.29E+02
AVE6_500	Mean	1.13E+04	9.03E+03	1.80E+04	1.88E+04	3.85E+04	3.79E+04	3.78E+04
	Std	8.74E+02	1.09E+03	1.02E+03	3.78E+03	1.00E+03	4.51E+03	1.69E+03
AVE7_100	Mean	2.38E+02	1.56E+02	1.42E+01	4.24E+02	9.22E+02	1.59E+01	4.18E+02
	Std	1.78E+01	1.22E+01	3.13E+00	7.49E+01	2.34E+01	1.05E+01	4.10E+01
AVE7_200	Mean	6.70E+02	4.68E+02	2.41E+02	1.53E+03	2.93E+03	7.38E+01	1.94E+03
	Std	3.30E+01	3.06E+01	3.03E+01	2.97E+02	3.99E+01	4.50E+01	8.43E+01
AVE7_500	Mean	2.51E+03	1.91E+03	2.54E+03	7.08E+03	1.27E+04	2.77E+02	1.04E+04
	Std	8.45E+01	5.94E+01	9.54E+01	1.31E+03	1.24E+02	1.89E+02	2.24E+02

optima, compared to GSO. Furthermore, Figures 4 and 5 reflect that SIGGSO possesses the ability in simultaneously locating as many solutions of AVE as possible which cannot be achieved directly by the original version of the compared metaheuristic algorithms.

5.5. Comparison of the Performances of GSO, SIGGSO, IADE, EDIWPSO, HSCH, HHO, and GA in Solving AVEs. We executed each of these algorithm 20 times independently and then compared the mean value and standard deviation of the fitness value of different test AVEs, as shown in Table 8. Table 9 shows the statistical results for the data of Table 8 using left-sided Wilcoxon signed-rank test. Table 10 shows the mean value and standard deviation of the execution time generated by the seven compared algorithms.

Table 8 indicates the following: (a) when solving low-dimensional, multisolution AVEs, SIGGSO, IADE, and EDIWPSO exhibit excellent solution accuracy and stability, whereas the performance of HSCH and GA is unsatisfactory the performance of GSO and HHO is the worst. As a matter

TABLE 9: Statistical results for the data of Table 6 using left-sided Wilcoxon signed-rank test ($\alpha = 0.05$).

Comparing order	p value	H	Signed-rank
SIGGSO vs. GSO	2.1282e-05	1	33
SIGGSO vs. IADE	0.2307	0	146
IADE vs. SIGGSO	0.7770	0	205
SIGGSO vs. EDIWPSO	5.5635e-06	1	2
SIGGSO vs. HSCH	9.4199e-06	1	24
SIGGSO vs. HHO	0.0150	1	126.5
SIGGSO vs. GA	9.1269e-07	1	0

of fact, HSCH and GA did not perform well on the whole test AVEs. Moreover, SIGGSO outperforms all the other algorithms in the comparison on AVE2. (b) When solving high-dimensional AVEs, EDIWPSO, HSCH, and GA are far inferior to SIGGSO and IADE in all aspects of our comparison. The performance of GSO is slightly inferior to SIGGSO on all high-dimensional AVEs. Furthermore, IADE outperforms

TABLE 10: Mean and standard deviation of the execution time (s) generated by the seven algorithms on the test AVEs.

AVE	Metrics	GSO	SIGSO	IADE	EDIWPSO	HSCH	HHO	GA
AVE4_100	Mean	1.29E+00	1.64E+00	1.43E+00	2.27E-01	4.38E+00	1.69E+00	7.04E-01
	Std	1.865E+01	7.10E-02	1.09E-01	2.69E-02	1.89E-01	7.17E-02	6.96E-02
AVE4_200	Mean	1.65E+00	2.69E+00	1.75E+00	6.33E-01	7.69E+00	2.36E+00	9.11E-01
	Std	6.51E-03	9.14E-02	3.50E-02	2.82E-02	5.29E-02	7.03E-03	1.09E-01
AVE4_500	Mean	4.52E+00	5.42E+00	3.00E+00	1.45E+00	1.90E+01	1.15E+01	1.69E+00
	Std	9.50E-02	1.05E-01	7.79E-02	6.85E-02	7.86E-02	3.46E+01	1.13E-01
AVE5_100	Mean	1.74E+00	1.92E+00	1.29E+00	2.49E-01	4.13E+00	2.84E+00	6.10E-01
	Std	4.23E-02	5.40E-02	2.53E-02	5.26E-02	8.14E-02	2.36E-02	4.37E-02
AVE5_200	Mean	3.10E+00	2.74E+00	1.69E+00	5.82E-01	7.72E+00	4.85E+00	9.00E-01
	Std	1.53E-01	9.21E-02	4.56E-02	2.62E-02	6.84E-02	3.33E-01	3.83E-02
AVE5_500	Mean	8.40E+00	5.45E+00	2.97E+00	1.60E+00	1.89E+01	1.61E+01	1.67E+00
	Std	1.29E+00	5.41E-02	1.04E-01	1.84E-01	1.19E-01	7.02E-01	6.83E-02
AVE6_100	Mean	1.91E+00	1.77E+00	1.26E+00	2.29E-01	4.11E+00	2.32E+00	6.03E-01
	Std	6.79E-01	1.13E-01	2.31E-02	3.60E-02	6.68E-02	7.84E-03	3.79E-02
AVE6_200	Mean	2.40E+00	3.42E+00	1.82E+00	6.96E-01	8.13E+00	3.65E+00	1.08E+00
	Std	3.88E-02	7.18E-01	8.94E-02	5.42E-02	1.61E-01	8.05E-02	1.01E-01
AVE6_500	Mean	4.13E+00	6.20E+00	3.27E+00	2.16E+00	1.95E+01	5.41E+00	1.70E+00
	Std	2.15E-02	3.26E-01	2.26E-01	1.11E+00	4.12E-01	4.34E-02	1.09E-01
AVE7_100	Mean	1.14E+00	1.88E+00	1.36E+00	2.43E-01	4.12E+00	1.65E+00	7.12E-01
	Std	1.77E-02	1.16E-01	7.86E-02	4.50E-02	5.84E-02	2.67E-02	1.46E-01
AVE7_200	Mean	1.67E+00	2.80E+00	1.75E+00	6.76E-01	7.79E+00	2.33E+00	8.76E-01
	Std	8.45E-03	1.53E-01	4.43E-02	5.69E-02	1.27E-01	3.53E-03	8.53E-02
AVE7_500	Mean	8.37E+00	5.74E+00	4.63E+00	1.77E+00	2.14E+01	1.59E+01	2.13E+00
	Std	1.64E+01	2.42E-01	1.17E+00	1.37E-01	1.56E-01	1.42E+01	9.64E-02

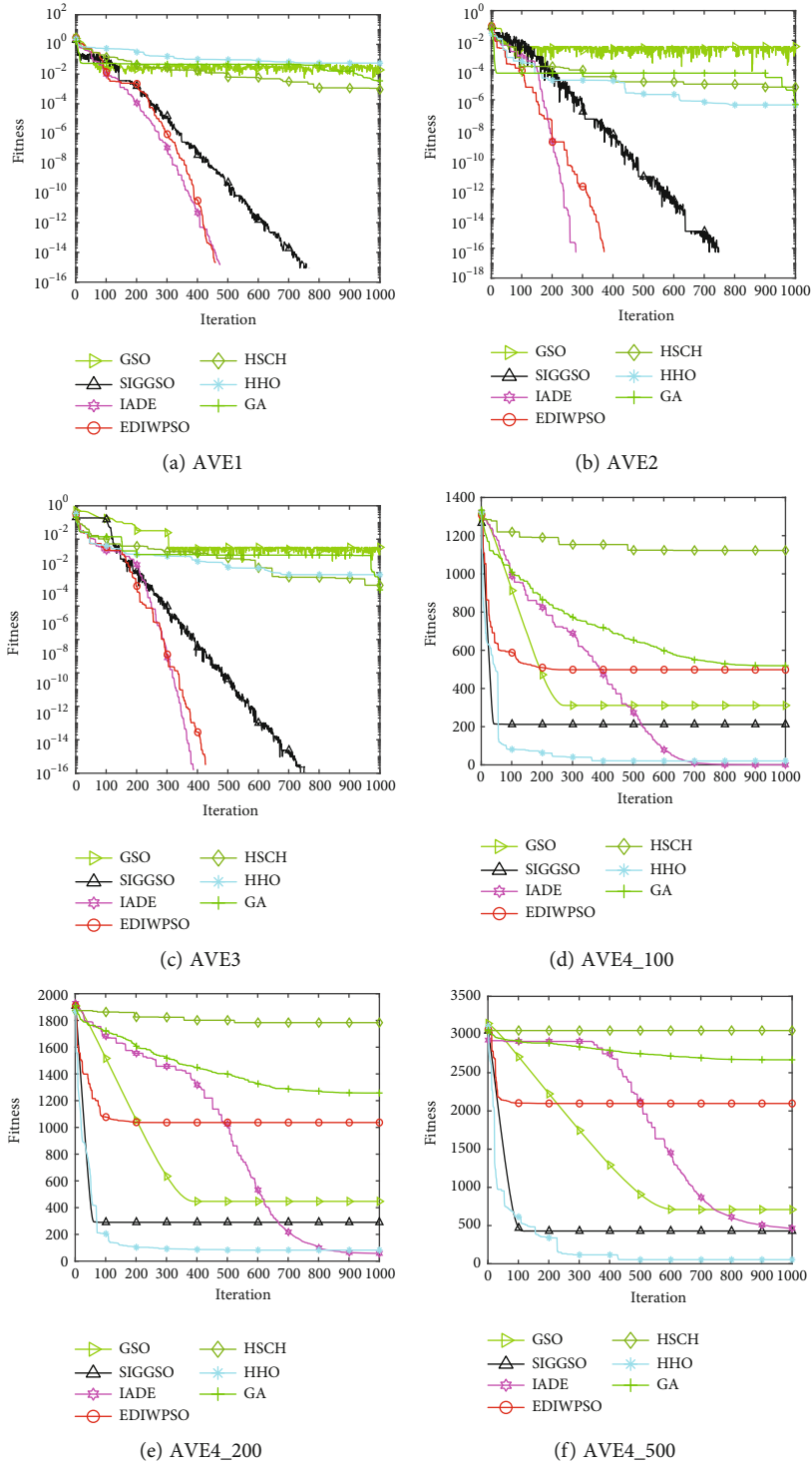


FIGURE 10: Continued.

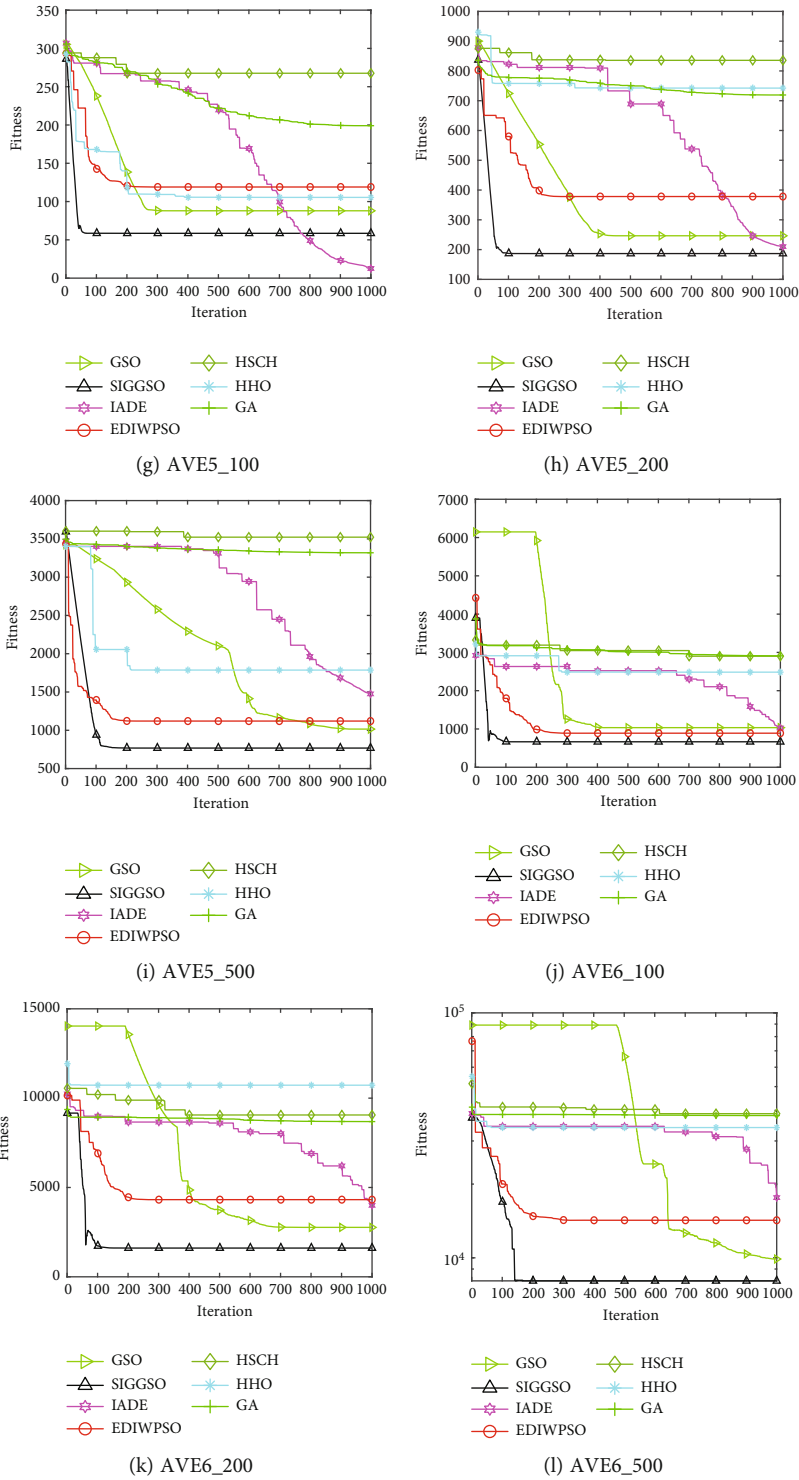


FIGURE 10: Continued.

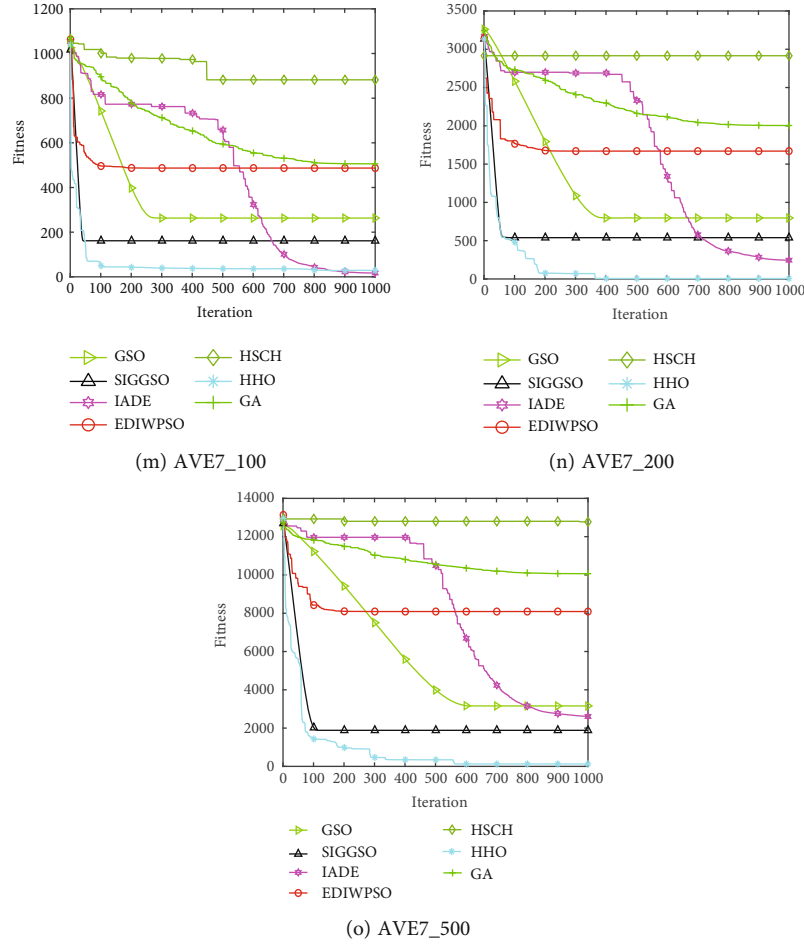


FIGURE 10: Fitness plots of the iterative process of the compared AVEs.

SIGGSO on AVE4_100, AVE4_200, AVE7_100, and AVE7_200. However, we noticed that IADE has a poor performance on all 500-dimensional AVEs compared with SIGGSO, which means that IADE failed to perform well on higher-dimensional AVEs. HHO outperforms SIGGSO on 50% test AVEs including AVE4_100, AVE4_200, AVE4_500, AVE7_100, AVE7_200, and AVE7_500. On high-dimensional AVE5 and AVE6, HHO did not perform well than SIGGSO.

For more convincing statistical analysis, Wilcoxon signed-rank test (WSRT) is adopted to perform pairwise comparisons between SIGGSO and the rest of algorithms. Here is the hypothesis for SIGGSO proposed in this study:

H_0 . The mean and standard deviation of the fitness value obtained by the SIGGSO is greater than that of another algorithm.

H_1 . The mean and standard deviation of the fitness value obtained by the SIGGSO is less than or equal to that of another algorithms.

We use the left-sided WSRT function $\text{signrank}(x, y, 'tail', 'left')$ in MATLAB for the above hypothesis test. Table 9 gives the statistical results on all test AVEs.

According to the p value, Table 9 shows that SIGGSO shows a significant improvement over GSO, EDIWPSO, HSCH, HHO, and GA with a level of significance $\alpha = 0.05$.

For the comparison between SIGGSO and IADE, the p value is 0.2307 (>0.05), which means that the difference between these two algorithms cannot be deemed significant. Meanwhile, for the comparison between IADE and SIGGSO, the p value is 0.7770 (>0.2307), which means that we prefer to accept the H_1 hypothesis over SIGGSO vs. IADE; i.e., SIGGSO possesses competitive advantages compared with IADE.

Table 10 shows that SIGGSO always requires more execution time than IADE, EDIWPSO, and GA on all the compared AVEs. Although EDIWPSO and GA required far less time than SIGGSO, the three algorithms fail to solve AVEs well, as they have lower solution accuracy and weaker stability than SIGGSO. Due to its' superiority in execution time compared to SIGGSO, IADE is preferable for solving AVEs, until the performance of IADE on higher-dimensional and multipeak AVEs is improved. Compared to SIGGSO, the execution time of HSCH is unacceptable in solving higher-dimensional AVEs. For example, HSCH required almost 19, 18.9, 19.5, and 21.4 seconds for solving AVE4_500, AVE5_500, AVE6_500, and AVE7_500, respectively. This is nearly 3.5 times than the time required by SIGGSO. Compared to SIGGSO, HHO not only required less time but also did not perform well on most test AVEs. Based on the

previous analysis, there is a trade-off in SIGGSO between the execution time and the solution accuracy and the capacity for solving high-dimensional AVEs.

Figure 10 displays the fitness plots of one iterative process of the compared AVEs obtained by the compared algorithms.

As shown in Figures 10(a)–10(c), SIGGSO requires more iterations to reach the global optima than the other metaheuristic algorithms. However, when solving high-dimensional AVEs containing AVE5_200, AVE_500, AVE6_100, AVE6_200, and AVE6_500, SIGGSO converges significantly faster, i.e., with fewer iterations than the other algorithms. Moreover, the ability of SIGGSO to leave the local optima is stronger than that of the other four algorithms. For instance, Figures 10(i)–10(l) show that the ability of IADE to leave the local optima is weaker.

All figures show that when solving either multisolution AVEs or high-dimensional AVEs, GA and HSCH fluctuate considerably during their early iterations and cannot leave the local optima, and their solution accuracy is the worst out of the five algorithms.

One point that is worth to consider is that SIGPSO requires slightly less iterations than EDIWPSO on most test AVEs. For example, from Figure 10(k), we can see that SIGPSO requires about 120 iterations to converge to a global optimum, whereas EDIWPSO needs more than 200 iterations. This shows that the adaptive model (9) designed in this paper is a thinkable adaptive step size strategy.

Considering that we need a robust algorithm to solve AVEs, SIGGSO is absolutely an effective technique to this end, especially for multisolution AVEs or high-dimensional AVEs. This is because SIGGSO uses effective adaptive step size techniques with GSO, and it indicates that the strategy selecting method is useful for solving AVEs.

6. Conclusion

The AVE is a NP-hard problem, and its solution has several forms. There are both low-dimensional and high-dimensional AVEs. There are also single- or multisolution AVEs. This study verified the advantages of GSO in solving multisolution AVEs compared to the other metaheuristic algorithms. As the basic GSO has relatively poor solution accuracy and struggles to solve high-dimensional AVEs, an improved GSO based on the sigmoid function, called SIGGSO, was proposed in this study. The sigmoid function was used to reconstruct the adaptive step size model in GSO. This improvement enhances the convergence rate of the GSO during the early iterations and improves its ability to capture the global optima during the later stages. Through numerical experiments, it was established that SIGGSO had higher solution accuracy and better stability compared to the other algorithms. For solving high-dimensional AVEs, it could converge to the global optimum with fewer iterations. The proposed SIGGSO algorithm can be applied to linear complementarity, bilinear programming, concave minimization, and other problems. Moreover, it can be applied to other continuous optimization problems.

In the end, we suggest several potential directions of future research. We intend to focus on improving the speed

of the algorithm for higher-dimensional AVEs, which is a limitation of SIGGSO, and we hope to find other activation functions of NN that may further improve the performance of GSO.

Data Availability

The data that support the findings of this study are available from the first author upon reasonable request.

Conflicts of Interest

There are no conflicts of interest to declare.

Acknowledgments

This research was supported by the 71st General Grant of China Postdoctoral Science Foundation (Project No. 2022M712680), General Research Project of Basic Science (Natural Science) in Jiangsu Province (Project No. 22KJB110027), General Project of Philosophy and Social Science Research in Jiangsu Universities (Project No. 2021SJA1079), and Research Initiation Foundation of Xuzhou Medical University (Project No. D2019046).

References

- [1] S. Yamanaka and M. Fukushima, *A Branch-and-Bound Method for Absolute Value Programs and Its Application to Facility Location Problems*, Kyoto University, Kyoto, 2011.
- [2] O. L. Mangasarian, “Knapsack feasibility as an absolute value equation solvable by successive linear programming,” *Optimization Letters*, vol. 3, no. 2, pp. 161–170, 2009.
- [3] B. Huang and C. Ma, “Convergent conditions of the generalized Newton method for absolute value equation over second order cones,” *Applied Mathematics Letters*, vol. 92, pp. 151–157, 2019.
- [4] O. L. Mangasarian and R. R. Meyer, “Absolute value equations,” *Linear Algebra and its Applications*, vol. 419, no. 2–3, pp. 359–367, 2006.
- [5] Z. Yu, L. Li, and Y. Yuan, “A modified multivariate spectral gradient algorithm for solving absolute value equations,” *Applied Mathematics Letters*, vol. 121, article 107461, 2021.
- [6] F. Wang, Z. Yu, and C. Gao, “A smoothing neural network algorithm for absolute value equations,” *Engineering*, vol. 7, no. 9, pp. 567–576, 2015.
- [7] X. Rao and X. Yan, “Particle swarm optimization algorithm based on information sharing in Industry 4.0,” *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 4328185, 2022.
- [8] Y. Feng, J. Yang, C. Wu, M. Lu, and X. J. Zhao, “Solving 0–1 knapsack problems by chaotic monarch butterfly optimization algorithm with Gaussian mutation,” *Memetic Computing*, vol. 10, no. 2, pp. 135–150, 2018.
- [9] R. M. Devi, M. Premkumar, P. Jangir, B. S. Kumar, D. Alrowaili, and K. S. Nisar, “BHGSO: binary hunger games search optimization algorithm for feature selection problem,” *CMC-Computers Materials & Continua*, vol. 70, no. 1, pp. 557–579, 2022.
- [10] L. Han, H. Xu, J. Ma, and Z. Jia, “A feature selection method of the island algorithm based on Gaussian mutation,” *Wireless*

- Communications and Mobile Computing*, vol. 2022, Article ID 1438999, 10 pages, 2022.
- [11] B. Shi, H. Ye, L. Zheng et al., "Evolutionary warning system for COVID-19 severity: colony predation algorithm enhanced extreme learning machine," *Computers in Biology and Medicine*, vol. 136, article 104698, 2021.
- [12] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [13] H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, "Dynamic Harris hawks optimization with mutation mechanism for satellite image segmentation," *Remote Sensing*, vol. 11, no. 12, p. 1421, 2019.
- [14] S. Ketabchi, H. Moosaei, and S. Fallahi, "Optimal error correction of the absolute value equation using a genetic algorithm," *Mathematical and Computer Modelling*, vol. 57, no. 9-10, pp. 2339–2342, 2013.
- [15] J. M. Feng and S. Y. Liu, "Particle swarm optimization algorithm based on inertia weight exponentially decreasing for solving absolute value equations," *Journal of Jilin University (Science Edition)*, vol. 54, no. 6, pp. 1265–1269, 2016.
- [16] G. Ning and Y. Zhou, "An improved differential evolution algorithm for solving absolute value equations," *In International Conference on High Performance Computing and Applications*, vol. 9576, pp. 38–47, 2015.
- [17] L. Yong, S. Liu, S. Tuo, and K. Gao, "Improved harmony search algorithm with chaos for absolute value equation," *Telkomnika*, vol. 11, no. 4, p. 835, 2013.
- [18] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions," *Swarm Intelligence*, vol. 3, no. 2, pp. 87–124, 2009.
- [19] J. Rohn, "A theorem of the alternatives for the equation $Ax + B|x| = b$," *Linear and Multilinear Algebra*, vol. 52, no. 6, pp. 421–426, 2004.
- [20] K. N. Krishnanand and D. Ghose, "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics," in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005*, pp. 84–91, Pasadena, CA, USA, 2005.
- [21] M. Marinaki and Y. Marinakis, "A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands," *Expert Systems with Applications*, vol. 46, pp. 145–163, 2016.
- [22] Y. Sun, R. Ma, J. Chen, and T. Xu, "Heuristic optimization for grid-interactive net-zero energy building design through the glowworm swarm algorithm," *Energy and Buildings*, vol. 208, article 109644, 2020.
- [23] J. Zhou and S. Dong, "Hybrid glowworm swarm optimization for task scheduling in the cloud environment," *Engineering Optimization*, vol. 50, no. 6, pp. 949–964, 2018.
- [24] F. Wang, C. Wang, K. Wang, Q. Jiang, B. Wang, and W. He, "Multiobjective differential evolution with discrete elite guide in internet of vehicles roadside unit deployment," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 4207130, 13 pages, 2021.
- [25] S. Ezadi, T. Allahviranloo, and S. Mohammadi, "Two new methods for ranking of Z-numbers based on sigmoid function and sign method," *International Journal of Intelligent Systems*, vol. 33, no. 7, pp. 1476–1487, 2018.
- [26] S. R. Chiluveru, M. Tripathy, and B. Mohapatra, "Accuracy controlled iterative method for efficient sigmoid function approximation," *Electronics Letters*, vol. 56, no. 18, pp. 914–916, 2020.
- [27] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization: a new method for optimising multi-modal functions," *International Journal of Computational Intelligence Studies*, vol. 1, no. 1, pp. 93–119, 2009.
- [28] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization for searching higher dimensional spaces," in *Innovations in Swarm Intelligence*, pp. 61–75, Springer, Berlin, Heidelberg, 2009.