Surprisingly Popular Algorithm-based Adaptive Euclidean Distance Topology Learning PSO

Xuan Wu†, Jizong Han†, Quanlong Cui, Liang Chen, Yanchun Liang,
Han Huang, Heow Pueh Lee, You Zhou⋆ and Chunguo Wu⋆

***Abstract*—The surprisingly popular algorithm (SPA) is a powerful crowd decision model proposed in social science, which can identify the knowledge possessed in of the minority. We have modelled the SPA to select the exemplars in PSO scenarios and proposed the Surprisingly Popular Algorithm-based Comprehensive Adaptive Topology Learning Particle Swarm Optimization. Due to the significant influence of the communication topology on exemplar selection, we propose an adaptive euclidean distance dynamic topology maintenance. And then we propose the Surprisingly Popular Algorithm-based Adaptive Euclidean Distance Topology Learning Particle Swarm Optimization (SpadePSO), which use SPA to guide the direction of the exploitation subpopulation. We analyze the influence of different topologies on the SPA. We evaluate the proposed SpadePSO on the full CEC2014 benchmark suite, the spread spectrum radar polyphase coding design and the ordinary differential equations models inference. The experimental results on the full CEC2014 benchmark suite show that the SpadePSO is competitive with PSO, OLPSO, HCLPSO, GL-PSO, TSLPSO and XPSO. The mean and standard deviation of SpadePSO are lower than the other PSO variants on the spread spectrum radar polyphase coding design. Finally, the ordinary differential equations models' inference results show that SpadePSO performs better than LatinPSO, specially designed for this problem. SpadePSO has lower requirements for population number than LatinPSO.**

***Index Terms*—particle swarm optimization, surprisingly popular, small world topology, comprehensive learning**

TABLE I
A LIST OF NOTATIONS USED IN THE PAPER.

Notation	Definition
$G(\boldsymbol{V}, \boldsymbol{E})$	Graph G with vertex set $\boldsymbol{V}$ and edge set $\boldsymbol{E}$
$G(\boldsymbol{V}, \boldsymbol{E}_{tmp})$	Graph G with vertex set $\boldsymbol{V}$ and edge set $\boldsymbol{E}_{tmp}$
$\boldsymbol{E}$	Edge set corresponding to the distance information
$\boldsymbol{E}_{tmp}$	Edge set randomly generated by assigned experts with high fitness values
$\boldsymbol{A}$	Adjacency matrix corresponding to the graph $G(\boldsymbol{V}, \boldsymbol{E})$
$\boldsymbol{B}$	Adjacency matrix corresponding to the graph $G(\boldsymbol{V}, \boldsymbol{E}_{tmp})$
$f(\cdot)$	Fitness function
r^{at}	Actual turnout
r^{et}	Expected turnout
θ	Surprisingly popularity degree
n_{exp}	Empirical number of experts
$\boldsymbol{T}(n_{exp})$	Set of the top n_{exp} particles according to fitness sorting
i_j	Descendingly ranking order of the particle $\boldsymbol{x}_j$
$prob(i_j)$	Connection probability in the $\boldsymbol{E}_{tmp}$
u_{lk}	Upper limit of the edges numbers in the $\boldsymbol{E}_{tmp}$
$v_{u_{lk}}$	Increasing velocity of u_{lk}
k	Out-degree of each particle in the knowledge transfer topology in the initial state

I. INTRODUCTION

MANY swarm intelligence algorithms and their variants have been proposed to solve various scientific research and practical problems, such as neural architecture search [1], ordinary differential equations optimization [2] and airfoil design [3]. Because of its simple principle, few parameters and fast convergence ability, particle swarm optimization (PSO) [4] is still an important tool for solving problems [5, 6].

There are a lot of reported studies on parameter setting, selection of neighborhood topology, improvement of learning strategy and hybridization of PSO with other algorithms [7] to balance exploration and exploitation. In most variants of PSO, the selection of learning exemplars depends on fitness directly, and individuals with high fitness might lead the population into a local trapping region when applied to complex multimodal problems. The selection method of the learning exemplars based on high fitness individuals implicitly adopts the idea of democratic voting, characterized by the majority predominance as well as the independence of personal judgment [8]. However, the democratic methods tend to highlight the most popular opinion, not necessarily the most correct. In fact, the democratic decision is biased for shallow and common information, at the expense of novel or specialized knowledge that is not widely known and shared [9, 10]. To overcome the limitations of democratic methods, Prelec et al. [11] proposed a sociological decision-making method called surprisingly popular algorithm (SPA) (or called surprisingly popular decision), which could identify the answer with the largest surprisingly popular degree from the swarm knowledge

This work is supported by the National Natural Science Foundation of China (61876069, 61876207 and 61972174), the Jilin Natural Science Foundation (20200201163JC), the Science and Technology Planning Project of Guangdong Province (2020A050510001), Guangdong Key-Project for Applied Fundamental Research (2018KZDXM076), and the STU Scientific Research Foundation for Talents (no. 35941918). *(†: The first two authors contributed equally to the work. Corresponding author: Chunguo Wu and You Zhou.)*

Xuan Wu, Jizong Han, You Zhou, Chunguo Wu are with the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin university, Changchun 130000, China(e-mail: wuuu20@mails.jlu.edu.cn; hanjz2116@mails.jlu.edu.cn; zyou@jlu.edu.cn; wucg@jlu.edu.cn).

Quanlong Cui is with the Commercial Quality And Efficiency Department, Baidu Inc, Beijing 100000,China.

Liang Chen is with the Department of Computer Science, Shantou University, Shantou 515000,China.

Yanchun Liang is with the Zhuhai Laboratory of Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, School of Computer, Zhuhai College of Jilin University, Zhuhai 519000, China.

Han Huang is with the School of Software Engineering, the South China University of Technology, Guangzhou 510000, China.

Heow Pueh Lee is with the Department of Mechanical Engineering, National University of Singapore, Singapore.

distribution, and hence, protect the valuable knowledge known by the minority.

Inspired by SPA, we try to guide the population not just with the particles with high fitness. Based on the preliminary work of the laboratory [12], we propose the adaptive euclidean distance topology and the Surprisingly Popular Algorithm-based Adaptive Euclidean Distance Topology Learning Particle Swarm Optimization (SpadePSO), and analyze the influence of different topology on SPA to give the topology selection under different dimensions.

The remaining parts of this paper are organized as follows: Section II introduces the related work briefly; Section III proposes the adaptive euclidean distance topology and SpadePSO and analyzes the influence of different topology on SPA; Section IV presents the experimental validation; and finally, Section V presents the conclusion.

II. Related Work

We first introduce related PSO variants In Section II-A. And then, we introduce small-world network in Section II-B. Finally, we introduce the SPA.

A. PSO variants

Zhan et al. [13] proposed the orthogonal learning PSO (OLPSO), which combine the personal best experience and population best experience to construct a learning exemplar for each particle. Liang et al. [14] proposed the comprehensive learning PSO (CLPSO), which was a famous PSO variant with strong exploration. In CLPSO, each dimension of a particle learns from its personal best position with probability η and learns from the corresponding dimensions of the other particles' personal best positions selected by tournament selection with probability $(1-\eta)$ independently. The velocity update formula is as follows:

$$v_{i,j} = wv_{i,j} + c_1 r_{1,j}\left(x^{cl}_{i,j} - x_{i,j}\right), \tag{1}$$

where $v_{i,j}$ and $x_{i,j}$ denote the velocity and position of the j-th dimension of particle i, respectively. c_1 is the acceleration coefficients, and $r_{1,j}$ is uniformly distributed random numbers independently generated within [0, 1] for the j-th dimension. w is the inertia weight to control the velocity. x^{cl}_i is constructed by the comprehensive learning strategy (CLS). By making each dimension of particle learn from other particles, the CLS can help particles escape from the local optimal region with higher probability, so that the CLS has stronger exploration.

Nandar et al. [15] proposed the heterogeneous comprehensive learning PSO (HCLPSO). HCLPSO contains two heterogeneous sub-populations, named exploration sub-population and exploitation sub-population, both sub-populations use the CLS proposed in the literature [14]. The particles of exploration sub-population only learn from the personal best experiences of other particles in the same sub-population dimensionally according to Eq. (1); whereas the particles of exploitation sub-population learn from the personal best experience of particles in the whole population dimensionally and the population-best experience traditionally according to Eq. (2):

$$v_{i,j} = wv_{i,j} + c_1 r_{1,j}\left(x^{cl}_{i,j} - x_{i,j}\right) + c_2 r_{2,j}\left(x^{gbest}_{j} - x_{i,j}\right), \tag{2}$$

where x^{gbest}_j denote the j-th of the population historical best position.

Xu et al. [16] proposed the particle swarm optimization based on the dimensional learning strategy (TSLPSO) with two sub-populations, in which one sub-population guides its particles to search locally by constructing exemplars learning dimensionally from the particles of sub-population, and the other sub-population guides its particles to search globally by constructing exemplars learning comprehensively from the best particles of the whole population.

B. Small-world network

Watts and Strogatz [17] had shown that information transmission through social networks would be affected by three factors of network structure: the number of neighbors, the number of clusters and the average shortest path length from one node to another. He also proposed the small-world network(also known as the WS small-world network), which was a network between regular networks and random networks. In WS small-world network, all the edges are reconnected on the basis of ring and regular networks with probability μ. Newman [18] proposed NW small-world network, in which nodes would not break any connection between any two nearest neighbors, but instead, would add a connection between two nodes with probability μ. The NW small-world network is somewhat easier to analyze than the WS small-world network because it does not lead to the formation of isolated clusters, whereas this can indeed happen in the WS small-world network.

Compared with the regular networks and the random networks, the small world networks could suppress the impact of individual particles in the population and maintain the population diversity. Hence, the small world network was widely used as connecting topologies in PSO [19–21].

C. Surprisingly popular decision

To give more weight to correct knowledge that may not be widely known, Prelec et al. proposed what they termed as the "surprisingly popular" decision method [11], which hinges on asking people two things about a given question: What do they think the right answer is, and how popular do they think each answer will be? For example, is Philadelphia capital of Pennsylvania? As shown in Fig. 1, 80% of people do not know the geography of the United States, may only recall that Philadelphia is a large, famous, historically significant city in Pennsylvania, and conclude mistakenly that it is the capital. 20% of people vote no probably possess an additional piece of evidence, that the capital is Harrisburg. However, as shown in Fig. 1, people with different knowledge have different perceptions of the popularity of their answers. People who know that Harrisburg is the provincial capital know that the popularity of their answers will be very low, while people who only know Philadelphia will blindly believe that most people have the same answers as themselves.

If the majority voting is used to determine the answer, it is obvious that the wrong answer will be given to this question. Since the SPA could highlight the correct answer, if somebody

knows the right answer, but also realizes that answer for most people is wrong, both of these pieces of information can be expressed.

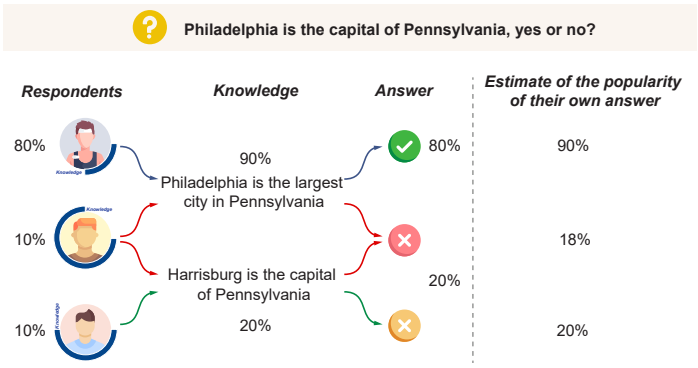


Fig. 1. Illustration of Surprisingly Popular Algorithm.

Based on inquiries of popularity, the SPA could highlight the correct answer. Lee et al. [22] used the SPA to predict the winners of National Football League (NFL) games and found that the SPA could predict better than many NFL media figures. Luo et al. [23] extend the SPA to classification problems and performed better than popular algorithms.

III. Our Method

In Section III-A, we propose three definitions of the SPA calculation process and briefly reviewed how to model the SPA in PSO [12]. In Section III-B, we propose the adaptive euclidean distance topology. In Section III-C, We propose the Surprisingly Popular Algorithm-based Adaptive Euclidean Distance Topology Learning Particle Swarm Optimization (SpadePSO),and then analyze sub-population. In Section III-D, we analyze the influence of different topologies on SPA and give the topology selection under different dimensions.

A. Modeling the surprisingly popular decision in PSO

To better understand the SPA, and carry out the follow-up work, we give the following three definitions of the SPA calculation process.

Definition 1 (Knowledge Prevalence Degree): It is the proportion of the total number of people in the group who know a certain knowledge. For example, the knowledge prevalence degree of "Philadelphia is the largest city in Pennsylvania" is 90%, and the knowledge prevalence degree of "Harrisburg is the capital of Pennsylvania" is 20%, as shown in Fig. 1.

Definition 2 (Expected Turnout): It is the group's estimation for the popularity of a given answer, which is the sum of each individual's estimation. The individual's estimation is the approximate proportion of people who have the same knowledge with them, each respondent can estimate the popularity of their answers through the knowledge prevalence degree. Respondents with two pieces of knowledge will estimate that the popularity of their answers is the product of the knowledge prevalence degree of the two pieces of knowledge. Respondents with only one piece of knowledge will estimate that the popularity of their answers is equal with its knowledge prevalence degree.

As shown in the Fig. 1, the expected turnout of the answer "Yes":

$$80\% \times 90\% + 10\% \times (1 - 18\%) + 10\% \times (1 - 20\%)$$
$$= 72\% + 8.2\% + 8\% = 88.2\%.$$

The expected turnout of the answer "No":

$$80\% \times 10\% + 10\% \times 18\% + 10\% \times 20\%$$
$$= 8\% + 1.8\% + 2\% = 11.8\%.$$

Definition 3 (Surprisingly Popular Degree): It is the ratio of the actual turnout to the expected turnout. In the SPA, the actual turnout of the correct answer is much higher than its expected turnout, which has the highest surprisingly popular degree.

The surprisingly popular degree of the answer "Yes" is $80/88.2 = 0.907$, while the surprisingly popular degree of the answer "No" is $20/11.8 = 1.695$. Hence, the answer "No" is considered to be the right answer in the SPA.

Next, We will briefly review how to model the SPA in PSO [12]. Denote the PSO population with $\boldsymbol{n}$ particles as $G(\boldsymbol{V}, \boldsymbol{E})$. The directed edge $\boldsymbol{e_{ij}}$ means that the particle $\boldsymbol{x}_i$ has the privilege to assess the fitness of the particle $\boldsymbol{x}_j$ and learn from the particle $\boldsymbol{x}_j$. The asymmetry adjacency matrix $\boldsymbol{A} = (a_{ij})_{n*n}$ is defined as follows:

$$a_{ij} = \begin{cases} 1 & e_{ij} \in E \\ 0 & e_{ij} \notin E \end{cases} \quad i, j \in \{1, 2, \cdots, n\}. \tag{3}$$

For a PSO population with 5 particles, Fig. 2 illustrates our surprisingly decision model. For $\boldsymbol{A}$, the summarization of the i-th row is defined as the amount of knowledge of the particle $\boldsymbol{x}_i$. The percentage summarization of the j-th column is defined as the knowledge prevalence degree of the particle $\boldsymbol{x}_j$.

First, each particle $\boldsymbol{x}_i$ votes for its neighbors given by the knowledge transfer topology $G(\boldsymbol{V}, \boldsymbol{E})$ as follows:

$$j_i^* = \arg \max_{\boldsymbol{x}_j \in V, a_{ij}=1} \{f(\boldsymbol{x}_j)\}, (i \in \{1, 2, \cdots, n\}), \tag{4}$$

where $f(\cdot)$ is the fitness function. Define the voting vector as follows:

$$J^* = (j_1^*, j_2^*, \cdots, j_n^*).$$

Denote $\boldsymbol{C} = \{\boldsymbol{x}_k | k \in \{1, 2, \cdots, n\},$ and $\exists i \in \{1, 2, \cdots, n\}, s.t. k = j_i^*\}$ and $M = |\boldsymbol{C}|$. $\boldsymbol{C}$ is the set of candidate exemplars. Accordingly, there are K particles that are voted out as the promising particles (or say "answers").

Count the voting results, and then, get the actual turnout of the particle $\boldsymbol{x}_i$, computing as follows:

$$r_k^{at} = \begin{cases} \left(\sum_{i=1, j_i^*=k}^{n} a_{ik}\right)/n & k \in C \\ iu0 & \text{else} \end{cases}. \tag{5}$$

The knowledge prevalence degree of the particle $\boldsymbol{x}_j$, computing as follows

$$r_j^{kp} = \left(\sum_{i=1}^{n} a_{ij}\right)/n. \tag{6}$$

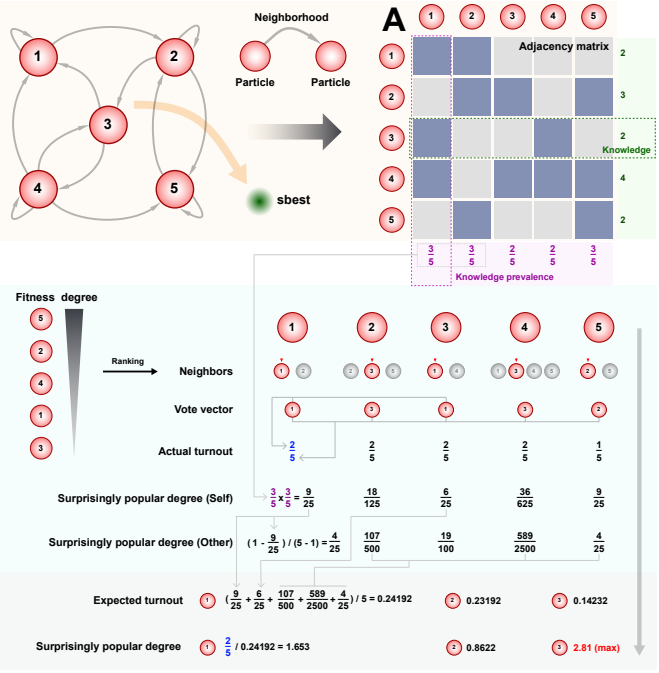


Fig. 2. Illustration of surprisingly popular degree calculation [12].

Let the particle $\boldsymbol{x}_i$ vote $\boldsymbol{x}_{j_i^*}$ to be the most promising particle as the guide of searching direction, and then, the popularity of its answer is computed as follows:

$$\alpha_{i,j_i^*} = \prod\nolimits_{\{j|j\in N(n),a_{ij}=1\}} r_j^{kp}. \tag{7}$$

Since all opposite answers of the particle $\boldsymbol{x}_i$ share the popularities $1-\alpha_{i,j_i^*}$, each popularity of all opposite answers is defined as follows:

$$\alpha_{ij} = \left(1-\alpha_{i,j_i^*}\right)/(n-1), (j \in \boldsymbol{N}(n), \text{ and } j \neq j_i^*). \tag{8}$$

The average summarization of popularities of $\boldsymbol{x}_k \in \boldsymbol{C}$ from all particles is taken as the expected turnout, denoted as r_k^{et}, which is defined as follows:

$$r_k^{et} = \left(\sum\nolimits_{i=1}^{n} \alpha_{ik}\right)/n. \tag{9}$$

Hereby, the surprisingly popularity degree, $\theta_k (k \in \boldsymbol{N}(M))$, for each candidate exemplar $\boldsymbol{x}_k \in \boldsymbol{C}$, could be defined as follows:

$$\theta_k = r_k^{at}/r_k^{et}. \tag{10}$$

Finally, the exemplar $\boldsymbol{x}^{sbest}$, the particle with the maximal surprisingly popular degree, deriving from the SPA could be selected as follows:

$$k^* = \arg\max_{\boldsymbol{x}_k \in \boldsymbol{C}} \{\theta_k\} \tag{11}$$

$$\boldsymbol{x}^{sbest} = \boldsymbol{x}_{k^*}, (\boldsymbol{x}_{k^*} \in \boldsymbol{C}). \tag{12}$$

As an example shown in Fig. 2, the No. 3 particle is $\boldsymbol{x}^{sbest}$, with the maximal surprisingly popular degree 2.81.

B. The strategy of adaptive euclidean distance topology

One of the most important parts of SPA is the voting process, which selects the exemplar to guide the search direction. In this process, each particle is modelled as an agent possessing knowledge including fitness, position and velocity of its neighbors specified by the given knowledge transfer topology $G(\boldsymbol{V}, \boldsymbol{E})$. Therefore, the reasonable topology has a great influence on the SPA.

Cui et al. [12] defined a knowledge transfer topology $G(\boldsymbol{V}, \boldsymbol{E})$, which updating knowledge topology adaptively. But in the initial stage, the connection based on the serial number can not reflect the fitness and position information of particles. Therefore we propose a new knowledge transfer topology in this Section.

The primary aspect for particles to consider when determining and adjust their knowledge adaptively is the relative position information of particles in a multidimensional space, since particles with similar positions have a lot of overlapping search space. The topology composed of particles with similar positions has excellent clustering performance, which cannot only ensure the diversity of the population, but also speed up the convergence of the algorithm.

In the initial state, suppose that the knowledge transfer topology is $G(\boldsymbol{V}, \boldsymbol{E})$, which is a directed graph with vertex set $\boldsymbol{V}$ and edge set $\boldsymbol{E}$ and is constructed based on distance information with all particle out-degrees k. Suppose that $\boldsymbol{D} = (d_{ij})_{n*n}$ is the Euclidian distance matrix of all particles. Particle $\boldsymbol{x}_i$ will connect the first k particles with the shortest Euclidean distance. It is worth noting that the edge set $\boldsymbol{E}$ is used to represent the distance information among particles in this paper and the edge set $\boldsymbol{E}$ is used to represent the learning relationship among particles in [12].

To meet the constraints of small-world networks, the out-degree of the particle in $G(\boldsymbol{V}, \boldsymbol{E})$, denoted as u_{lk} (Upper limit of knowledge about the distance information), increases linearly with the iterations for the successive iterationand the updating formula is as follows:

$$u_{lk} = \left\lfloor k + \left(v_{u_{lk}} * \frac{t}{T}\right) \right\rfloor, \tag{13}$$

where t and T are the current and the maximal iteration numbers, respectively, $v_{u_{lk}}$ is the speed at which the upper limit of the knowledge of distance information.

It is worth noting that all particle has the same out-degree, but different in-degree during the iteration as shown in Fig. 3. In Fig. 3 the connection according to the Euclidian distance when u_{lk} is set to 4 (The graph allows self-loops, each node has a self-connected edge, which is not shown in Fig. 3), the exploration sub-population size is set to 3, the exploitation sub-population size is set to 5 and the dimension is set to 2 (Heterogeneous populations are described in Section III-C). It is clear that particle 1, in the center of the graph, has the most in-degree, so that has been learned by the most particles. Particle 2 and 4, at the bounds of the current search space, have the least in-degree, so that has been learned by the least particles. It is possible to find potential surprisingly popular exemplars in particles that are less learned or at the bounds of

the current search space, which can change the search direction and expand the search space reasonably.

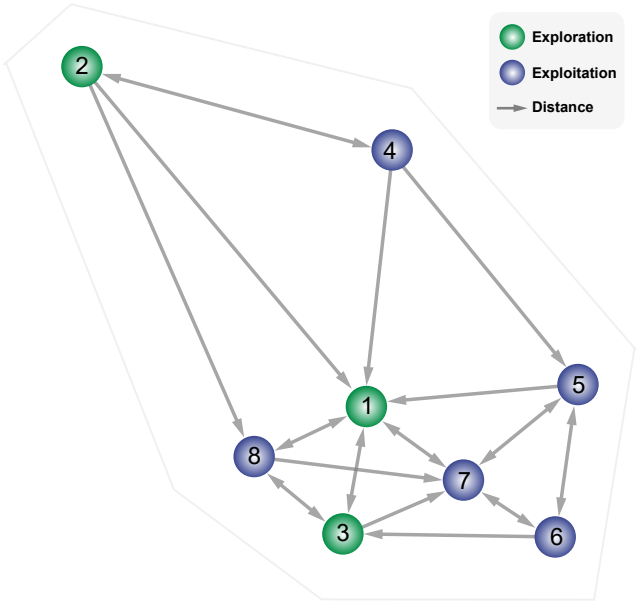


Fig. 3. Illustration of connection in the relative position.

Except $G(\boldsymbol{V},\boldsymbol{E})$, we use the temporary directed graph $G(\boldsymbol{V},\boldsymbol{E}_{tmp})$ proposed in [12] to speed up the convergence of the algorithm. In every iteration, $G(\boldsymbol{V},\boldsymbol{E}_{tmp})$ is generated by the n_{exp} largest fitness values particles with a certain probability, named experts. Define $\boldsymbol{T}(n_{exp})$ is the set of the experts. The generating rule of the adjacency matrix $\boldsymbol{B}$ of $G(\boldsymbol{V},\boldsymbol{E}_{tmp})$ is as follows:

$$b_{ij}=\begin{cases}1 & \boldsymbol{x}_j\in T(n_{exp}) \text{ and } rand<prob(i_j)\\ 0 & \text{else}\end{cases}, \quad (14)$$

where n_{exp} is an empirical number of experts, and $prob(i_j)$ is the re-connection probability defined as follows:

$$prob(i_j)=\begin{pmatrix} n-i_j \\ n_{exp^{-1}}\end{pmatrix}/\begin{pmatrix} n \\ n_{exp}\end{pmatrix}, i_j\in \boldsymbol{T}(n_{exp}). \quad (15)$$

Suppose the relative position of the population is shown in Fig. 4. A SPA is made in the t-th iteration according to $G_t(\boldsymbol{V},\boldsymbol{E})$ and $G_t(\boldsymbol{V},\boldsymbol{E}_{tmp})$. Supposed u_{lk} in $G_t(\boldsymbol{V},\boldsymbol{E})$ is set to 3. Each particle $\boldsymbol{x}_i$ connects 2 particles with the shortest Euclidean distance and connects itself. Specifically, the joint-directed graph $G_t(\boldsymbol{V},\boldsymbol{E}\cup\boldsymbol{E}_{tmp})$ could be generated and used to vote by using the proposed SPA in SpadePSO. Supposed u_{lk} in $G_{t+1}(\boldsymbol{V},\boldsymbol{E})$ increases by 1 according to Eq. (14), Each particle $\boldsymbol{x}_i$ connects 3 particles with the shortest Euclidean distance and connects itself.

C. SpadePSO

To balance exploration and exploitation, we propose the Surprisingly Popular Algorithm-based Adaptive Euclidean Distance Topology Learning Particle Swarm Optimization (SpadePSO), which contains two heterogeneous subpopulations. The velocity of the particle in exploration subpopulation is updated according to Eq. (1), which endows the

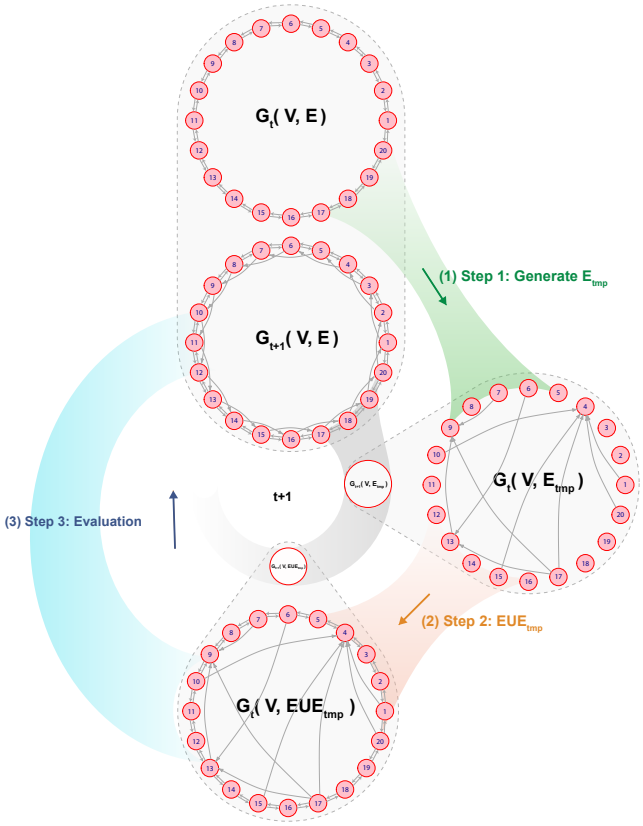


Fig. 4. Illustration of Topology Updating Process.

subpopulation strong global exploration ability. The velocity of the particle in exploitation sub-population is updated according to the Eq. (16), which endows the subpopulation strong local exploitation ability:

$$v_{i,j}=wv_{i,j}+c_1r_{1,j}\left(x_{i,j}^{cl}-x_{i,j}\right)+c_2r_{2,j}\left(x_j^{sbest}-x_{i,j}\right), \quad (16)$$

where x_j^{sbest} denote the j-th of the particle with the maximal surprisingly popular degree, used to guide the exploitation subpopulation.

In the exploration sub-population, particles only use the CLS to learn only from the personal best experience of the other particles selected randomly from this same subpopulation. Hence, there is no accumulation of learning experience about search direction in the exploitation sub- population, which has high population diversity and strong exploration ability. In the exploitation sub-population, particles learn not only from the exemplar constructed by the DLS, but also from the best experience of the entire population. As depicted in fig. 3, the knowledge transfer topology $G(\boldsymbol{V},\boldsymbol{E})$ consists of the entire population, and $\boldsymbol{x}^{sbest}$ is constructed by the entire population. Therefore, the exploitation sub-population has stronger exploitation ability. The flowchart is shown in Fig. 5.

The population diversity can be used to measure the exploration and exploitation [24]. To more intuitively measure the exploration and exploitation abilities of the two subpopulations of the SpadePSO, we compare the diversities of the exploration sub-population, the exploitation sub-population

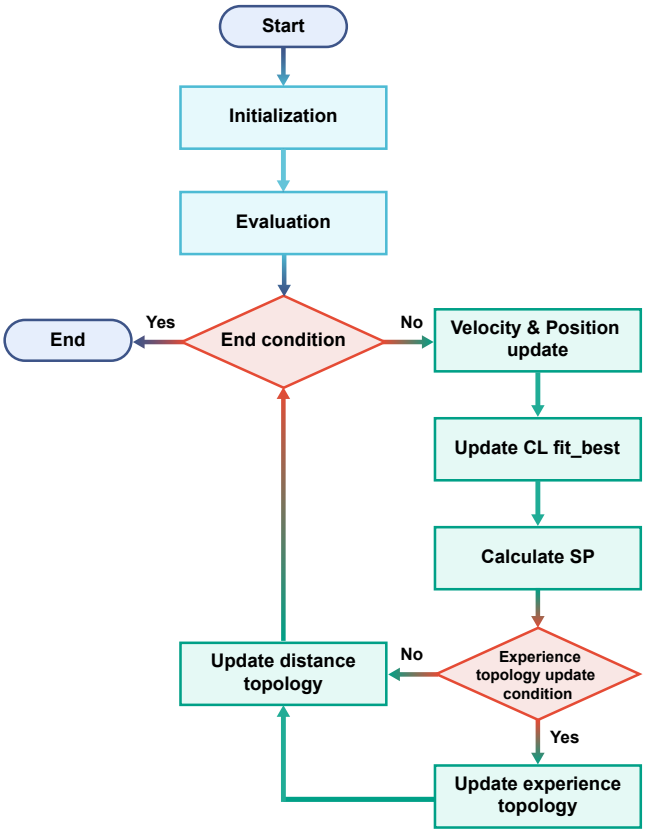


Fig. 5. Flowchart of the proposed algorithm

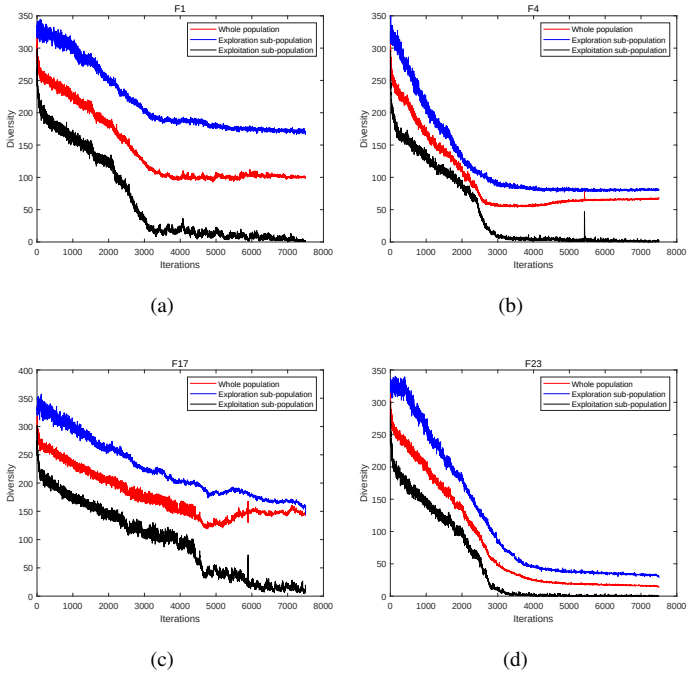


Fig. 6. Diversity comparisons between the exploration sub-population, the exploitation sub-population, and the whole population.

and the whole population of SpadePSO. The population diversity is calculated according to the following formulas.

$$Diversity = (\sum_{i=1}^{n} \sqrt{\sum_{j=1}^{D} (x_{i,j} - \overline{x_j})})/n, \tag{17}$$

$$\overline{x_j} = (\sum_{i=1}^{n} x_{i,j})/n. \tag{18}$$

where n is the population size, D is the dimension of the search space, $x_{i,j}$ denotes the j-th dimension of the i-th particle, x_j denotes the j-th dimension of the center position $\overline{x}$ of the population.

In Fig. 6, we present the diversity curves of some CEC2014 benchmark suite functions (F1, F4, F17 and F23) [25], where the dimension of the search space, the population size and the maximum number of function evaluations are, 30, 40 and 7500, respectively. The results show that the exploration sub-population maintains the highest diversity, which can ensure that SpadePSO jump out of a local trapping region. The exploitation sub-population keeps the smallest diversity, consequently, converges rapidly. To sum up, the heterogeneous population design has reached our original intention.

D. Influence of different topology on SPA

Reasonable topology can improve the performance of the algorithm. To prove the advantages of our topology, we compare the performance of the SPA with three different topologies on the CEC2014 benchmark suite [25]. The CEC2014 benchmark suite consists of 30 test functions and the different dimensions, including D = 10, 30, 50 and 100, are tested. The rest of the experimental details are presented in Section IV.

The first topology is proposed in Section III-B. In the initial stage, the connections between particles are determined by the Euclidean distance. During the iteration, except for the Euclidean distance information, selected particles with high fitness values are used to speed up the convergence of the algorithm. The second is the topology of SPA-CatlePSO proposed in [12]. In the initial stage, each particle x_i is unidirectionally connected to particles numbered from $i+1$ to $i+k$. During the iteration, particles are guided according to the learning experience in the search direction. And selected particles with high fitness values are used. If particle $\boldsymbol{x}_i$ votes to $\boldsymbol{x}^{sbest}$ and its fitness value is improved in the current iteration, the particle $\boldsymbol{x}_i$ will connect to $\boldsymbol{x}^{sbest}$ in the next iteration. In the third topology, we combine the first and second topologies. In the initial stage, the connections between particles are determined by the Euclidean distance. During the iteration, particles are guided according to the learning experience in the search direction, Euclidean distance information and particles with high fitness values. If particle $\boldsymbol{x}_i$ votes to $\boldsymbol{x}^{sbest}$ and its fitness value is improved in the current iteration, the particle $\boldsymbol{x}_i$ will connect to $\boldsymbol{x}^{sbest}$ in the next iteration.

To make the experimental results more convincing, we use the well-known nonparametric statistics analysis Wilcoxon signed ranks test [26] to check whether the algorithm performance priority is significant. The comparing results are based on the mean values of errors. TABLE II presents the Wilcoxon signed ranks test of the SPA with three different topologies on 30 functions with 10, 30, 50, 100 dimensions, respectively. The symbols +, -, $\approx$ indicate that the first

topology performs significantly better (+), significantly worse (-), or not significantly different (≈) compared to the given algorithm. The number in brackets in the table is the p-value between the two algorithms. An algorithm can be considered significantly better than another at different significant levels, if its p-value is less than 0.05.

As shown in TABLE II, the first topology performs best in 30, 50 and 100 dimensions but worst in 10 dimensions. The first topology is significantly better than the second topology in 30 dimensions and better than the third topology in 100 dimensions. The reason for this is that the Euclidean distance information has stronger exploration than the learning experience in the search direction, Therefore, if the problem is multidimensional, the most suitable topology for SPA is the first topology proposed in this paper.

TABLE II
THE WILCOXON SIGNED RANKS TEST OF THREE DIFFERENT TOPOLOGIES OF SPA.

the first topology VS.	Dimension / sign	10D	30D	50D	100D
the second topology	+	13	19 (0.05)	18	18
	-	16	10	12	11
	≈	1 (0.61)	1	1(0.33)	0 (0.21)
the third topology	+	13	15	16	17(0.08)
	-	15	14	13	11
	≈	2 (0.82)	1 (0.74)	1(0.84)	2

IV. EVALUATING SPADEPSO

There are 5 parts to this Section. Section IV-A includes parameters selection of SpadePSO. Section IV-B describes the complexity of the proposed SpadePSO. Section IV-C evaluates the performance of SpadePSO on the full CEC2014 benchmark suite [25]. Sections IV-D and IV-E are designed for practical problems.

In Section IV-A, IV-B and IV-C, The test problem is the CEC2014 benchmark suite[25]. The CEC2014 benchmark suite consists of 30 test functions. Functions F1 $\sim$ F3 are unimodal. F4 $\sim$ F16 are simple multimodal functions. F17 $\sim$ F22 are hybrid functions. Finally, F23 $\sim$ F30 are composite functions combing multiple test problems into a complex landscape. The 30-dimension is tested in Section IV-A. The different dimensions, including D = 10, 30, 50 and 100, are tested in Sections IV-B and IV-C. For all functions, the search range is $[-100, 100]^D$. In Section IV-D, We evaluate the performance of SpadePSO on a widely used practical optimization problem known as the spread spectrum radar polyphase code design, whose dimension D is 20. Modeling of dynamic systems in the field of physical, biology and chemistry are commonly achieved by Ordinary Differential Equations. In Section IV-E, We use SpadePSO to infer the parameters and structure of the HIV model whose dimension D is 15.

The population size is set uniformly as 40. The size of exploitation sub-population and exploration sub-population in SpadePSO is 5:3. The maximum value of velocity v_{max} is set to 10% of the search range [27] and the maximum number of objective function calls per run is $D\times$10000 [28]. The number of runs per function is 30.

A. Algorithm for Parameter selection

For the proposed SpadePSO algorithm, there are 3 parameters to be adjusted to fully maintain the application environment of SPA and ensure the convergence speed. These parameters include the out-degree of each particle in the small-world topology in the initial state (k), the upper limit of degree changing speed ($v_{u_{lk}}$) in Eq. (13) and the number of expert particles (n_{exp}) in the temporary matrix in Eq. (14). Traditional parameters, including the size of sub-population and the acceleration coefficients (c, c_1, c_2), are used as the same as those in HCLPSO [15].

In this Section, the parameters are tuned using the first 16 functions of the CEC2014 benchmark suite. The results are ranked based on the mean values of errors in TABLE III. According to this final rank, we chose k = 2 and v_{ulk} = 6 to test the expert number n_{exp}. Finally, n_{exp} = 5 is the optimal expert number. After the parameter trials, the optimal parameters are listed in TABLE IV.

TABLE III
THE FRIEDMAN TEST OF THE KEY PARAMETER TUNE.

Friedman test of $k + v_{ulk}$ without the expert temporary graph					
k	2	4	6	8	10
v_{ulk}	2	4	6	8	10
Ave. rank	3.19	2.66	2.84	3.16	3.16
Final rank	5	1	2	3	4

Friedman test of k and v_{ulk} without the expert temporary graph								
k	1	2	3	4	5	6	7	8
v_{ulk}	7	6	5	4	3	2	1	0
Ave. rank	4.13	3.88	4.84	4.49	4.59	5.19	5.22	3.88
Final rank	3	1	6	4	5	7	8	2

Friedman test of n_{exp} with k = 2, v_{ulk} = 6					
n_{exp}	2	3	4	5	6
Ave. rank	3.28	2.97	3.06	2.75	2.94
Final rank	5	3	4	1	2

TABLE IV
PARAMETERS SETTING OF THE COMPARED ALGORITHMS.

	Algorithm	Parameters settings	Ref.
1.	PSO (1995)	w : 0.9∼0.4, c_1 = c_2 = 2	[4]
2.	CLPSO (2006)	w : 0.9∼0.4, c_1 = c_2 = 1, c = 0.5	[14]
3.	OLPSO (2011)	w : 0.9∼0.4, c_1 = c_2 = 1, c = 0.5	[13]
4.	L-SHADE (2014)	$r^{N^{init}}$ = 18, r^{arc} = 2.6, p = 0.11, H = 6	[29]
5.	HCLPSO (2015)	w : 0.99∼0.2, c_1 : 2.5∼0.5, c_2 : 0.5∼2.5, c : 3∼1.5	[15]
6.	GL-PSO (2016)	w = 0.7298, c_1 = c_2 = 1.49618, C_R = 0.5, I = 4, δ = 0.2, $F_i \in [-1, -0.4] \cup [0.4, 0.1]$	[30]
7.	TSLPSO (2019)	w : 0.9∼0.4, c_1 = c_2 = 1.5, c_3 : 0.5∼2.5	[16]
8.	XPSO (2020)	η = 0.5, $Stag_{max}$ = 5, p=0.2	[31]
9.	LatinPSO (2019)	w = 0.7, c_1 = 1.5, c_2 = 1.5	[32]
10.	SpadePSO	w = 0.99∼0.2, c_1 = 2.5∼0.5, c_2 = 0.5∼2.5 c : 3∼1.5, k = 2, v_{ulk} = 6, n_{exp} = 5	

B. Algorithm Complexity

This Section describes the complexity of the proposed SpadePSO algorithms. All experiments are executed on the system described as follows:

- OS: Windows 10
- CPU: core i7 (2.90GHz)

- RAM: 16GB
- LANGUAGE: MATLAB 2020a

C. The CEC2014 benchmark suite

The experiment is performed with the CEC2014 benchmark suite to verify the strength of the proposed algorithms: SpadePSO. The algorithms in comparison include PSO [4], CLPSO [14], OLPSO [13], L-SHADE [29], HCLPSO [15], GL-PSO [30], TSLPSO [18], XPSO [31] and SpadePSO. XPSO, HCLPSO and TSLPSO are algorithms that improved the neighborhood topology of the population. CLPSO proposed the comprehensive learning strategy which has stronger exploration. OLPSO used the orthogonal learning strategy to construct the learning exemplar. L-SHADE, one of the state-of-the-art Differential Evolution (DE) algorithms, was the champion algorithm of the CEC2014 benchmark suite. HCLPSO and TSLPSO were heterogeneous particle swarm optimization, with two sub-populations responsible for exploitation and exploration, respectively. GL-PSO integrated the advantages of the genetic algorithm into PSO by introducing crossover, mutation and selection operations to construct the learning exemplars. In XPSO, particles learned from both locally and globally best particles, and dynamically update the topology. The parameter settings of each algorithm are listed in TABLE IV.

TABLE V presents the Wilcoxon signed ranks test of the proposed SpadePSO on 30 functions with 10, 30, 50, 100 dimensions, respectively. The comparing results are based on the mean values of errors.

As shown in TABLE V, SpadePSO is significantly better than PSO, OLPSO, GL-PSO and XPSO in 10, 30, 50 and 100 dimensions. The performance of SpadePSO and CLPSO is very similar on 30, 50 and 100 dimensions. However compared with CLPSO, SpadePSO has stronger exploitation, which is the reason that SpadePSO performs better on 10-dimension functions. With the increase of dimensions, SpadePSO performs better than TSLPSO. It is very interesting that SpadePSO behaves significantly better than HCLPSO on 10 and 50 dimensions but similarly to HCLPSO on 30 and 100 dimensions. It is a strange situation. They are all heterogeneous algorithms and the mechanism of heterogeneous algorithms is very complicated and worth further investigation. L-SHADE is significantly better than PSO and all variants. Paper [33], [34] show the performance of other variants of DE algorithm on the CEC2014 benchmark suite. It can be clearly observed that other variants of DE algorithm are better than PSO algorithms. Hence, the essential difference between DE and PSO is worth deliberate investigation.

TABLE V
COMPARISON OF SPADEPSO WITH STATE-OF-THE-ART OPTIMIZATION ALGORITHMS ON THE CEC2014 BENCHMARK SUITE.

SpadePSO VS.	Dimension / sign	10D	30D	50D	100D
PSO	+	29 (0.00)	23 (0.01)	28 (0.00)	28 (0.00)
	-	1	7	2	2
	≈	0	0	0	0
CLPSO	+	19 (0.08)	15	15	17
	-	11	15	14	13
	≈	0	0 (0.88)	1 (0.58)	0 (0.88)
OLPSO	+	25 (0.00)	19 (0.01)	22 (0.00)	20 (0.02)
	-	5	11	7	10
	≈	0	0	1	0
L-SHADE	+	4	1	2	4
	-	25 (0.00)	28 (0.00)	24 (0.00)	26 (0.00)
	≈	1	1	4	0
HCLPSO	+	25 (0.01)	20	21 (0.05)	19
	-	5	10	8	10
	≈	0	0 (0.32)	1	1 (0.27)
GL-PSO	+	29 (0.00)	24 (0.00)	26 (0.00)	24 (0.00)
	-	1	6	4	6
	≈	0	0	0	0
TSLPSO	+	17	19	15	18 (0.09)
	-	13	11	14	12
	≈	0 (0.73)	0 (0.19)	1 (0.19)	0
XPSO	+	25(0.00)	26(0.00)	27(0.00)	25 (0.00)
	-	4	4	3	5
	≈	1	0	1	0

D. Spread spectrum radar polyphase coding design

In this Section, we take the optimization problem of spread spectrum radar polyphase coding design (SSRP) as a practical problem to verify the performance of SpadePSO. Since its NP-hard property and the fact that its fitness function is piecewise smooth, SSRP is widely used as the optimization object of swarm intelligence algorithms [35, 36]. We adopt the min-max nonlinear optimization problem model [37] as the fitness function to solve the problem of SSRP, the formula is as follows:

$$\begin{aligned} &\min_{x\in X} f(x) = \max\{\phi_1(x), ...\phi_{2m}(x)\}, \\ &X = \{(x_1, ..., x_n) \in R^n | 0 \le x_j \ge 2\pi, j = 1, ..., n\} \end{aligned} \tag{19}$$

where $m = 2n - 1$, and

$$\begin{aligned} &\phi_{2i-1}(x) = \sum_{j=i}^{n} \cos(\sum_{k=|2i-j-1|+1}^{j} x_k), i = 1, ...n, \\ &\phi_{2i}(x) = 0.5 + \sum_{j=i+1}^{n} \cos(\sum_{k=|2i-j|+1}^{j} x_k), i = 1, ...n-1, \\ &\phi_{m+i}(x) = -\phi_i(x), i = 1, ...m, \end{aligned} \tag{20}$$

As shown in TABLE VI, the mean and standard deviation of SpadePSO are both lower than the others. The best objective value of XPSO is 1.03, which is the optimal fitness, and its mean is 1.55. The mean of PSO is similar to SpadePSO, but its standard deviation is larger.

TABLE VI
COMPARISON OF SPADEPSO WITH STATE-OF-THE-ART OPTIMIZATION ALGORITHMS ON THE SSRP.

Algorithm / Stat.	PSO	CLPSO	OLPSO	HCLPSO	GL-PSO	TSLPSO	XPSO	SpadePSO
best	1.31	1.40	1.32	1.31	1.15	1.23	1.03	1.23
mean	1.55	1.75	1.78	1.60	1.77	1.61	1.55	1.50
std	0.19	0.12	0.26	0.15	0.23	0.14	0.23	0.11
rank	3	6	8	5	7	4	3	1

E. Ordinary differential equations models inference

Cause of the requirement of a higher computational, representation method of structure and design of the solution space, inferring the structure and parameters of the ordinary

differential equations models simultaneously is a more challenging task [2, 38]. To prove the effectiveness of SpadePSO, we infer the structure and parameters of the HIV model from scratch, which means we unknown anything about variables, parameters and interrelation between items in the initial stage. The HIV model is described by Eqs. (21).

$$\begin{aligned} \mathrm{d}T/\mathrm{d}t &= +\lambda - \rho * T - \beta * T * V, \\ \mathrm{d}I/\mathrm{d}t &= +\beta * T * V - \delta * I, \\ \mathrm{d}V/\mathrm{d}t &= +n * \delta * I - c * V - \beta * T * V. \end{aligned} \tag{21}$$

In the HIV model, the variables are T, I, and V, where T is the number of uninfected cells, I the number of infected CD4+T lymphocytes and V the number of free viruses. The parameters are λ, ρ, β, δ, n and c, where λ is the creation rate of uninfected cells, ρ the death rate of uninfected cells, β infection rate constant, δ the death rate of infected cells, n the number of viruses that are released during lysis of one infected cell, c the rate at which viruses disappear. $+$ and $-$ represent interrelations between items.

In the HIV model, λ is 80, ρ is 0.15, β is 0.00002, δ is 0.55, n is 900, c is 5.5. The initial conditions (variable values at time $t = 0$) is as follows: $T_0 = 100$, $I_0 = 150$, and $V_0 = 50,000$. The HIV model is shown in Eqs (22).

$$\begin{aligned} \mathrm{d}T/\mathrm{d}t &= 80 - 0.15 * T - 0.00002 * T * V, \\ \mathrm{d}I/\mathrm{d}t &= 0.00002 * T * V - 0.55 * I, \\ \mathrm{d}V/\mathrm{d}t &= 900 * 0.55 * I - 0.55 * V - 0.00002 * T * V. \end{aligned} \tag{22}$$

Tian et al. [32] defined the general form of an ODE as shown in Eq. (23). It is assumed that the equation of the HIV model is combined by four different items, including two single-variable items, one two-variable item and one item without variable. The interrelation between two items within an equation is defined by the addition and subtraction operator.

$$\mathrm{d}x_i/\mathrm{d}t = \pm k_1 * x_i \pm k_2 * x_a \pm k_3 * x_b * x_c \pm k_4, \tag{23}$$

where $i \in \{1, 2, 3\}$.

The representation method of Eq. (23) is $(k_1, k_2, k_3, k_4, (\pm, \pm, \pm, \pm, x_a, x_b, x_c))$ [32]. The first variable x_i must exist in Eq. (23) because Eq (23) is a differential equation about x_i, so we don't code x_i into the representation of Eq. (23). (k_1, k_2, k_3, k_4) are parameters of the equation, $(\pm, \pm, \pm, \pm)$ are interrelations between items and (x_a, x_b, x_c) are variables. $\pm$ represents one of the addition or subtraction operator. x_a represents a variable in the model except for x_i. x_b and x_c represent any variable in the model, respectively. For example, if $i = 1$, x_a represents one of the state variables x_2 and x_3, x_b and x_c represent one of the state variables x_1, x_2 and x_3. Therefore, there are 192 structures as shown in TABLE VII. To solve the problem, we use the serial number to represent structure as a dimension of the problem. Since the HIV model has three equations, the dimension of the problem is 15 of which 12 dimensions represent the parameters and 3 dimensions represent the structure of equations.

TABLE VII
ODE STRUCTURES LIST ABOUT EQ. (23).

Serial number	ODE structures
1	$(+,+,+,+,I,T,T)$
2	$(+,+,+,+,I,T,I)$
3	$(+,+,+,+,I,T,V)$
4	$(+,+,+,+,I,I,I)$
5	$(+,+,+,+,I,I,V)$
6	$(+,+,+,+,I,V,V)$
7	$(+,+,+,+,V,T,T)$
...	.....
13	$(+,+,+,-,I,T,T)$
...	.....
192	$(-,-,-,-,I,V,V)$

The objective of an individual is defined as the sum of the squared error and shown in Eq. (24).

$$f(\mathbf{x}) = \sum_{i=1}^{n} \sum_{k=0}^{T} \left(\mathbf{x}'(t_0 + k\Delta t)\right) - \left(\mathbf{x}(t_0 + k\Delta t)\right)^2, \tag{24}$$

where, t_0 is the starting time, Δt the step size, n is the number of the state variables, and T the number of data points. $\mathbf{x}(t_0 + k\Delta t)$ is the given target time course data($k = 0, 1, 2, \ldots, T - 1$). $\mathbf{x}'(t_0 + k\Delta t)$ is the time course data acquired by calculating the system of ODE represented by a particle. If the inferred model is the same as the HIV model, the objective value is 0. On the contrary, the larger the objective value, the greater the difference between the inferred model and the HIV model.

The algorithms in the comparison include PSO, LatinPSO [32] and SpadePSO. LatinPSO is designed to infer the structure and parameters of the ordinary differential equations models. The parameter settings of each Algorithm are listed in TABLE IV. The iteration number is set to 3750 calculated according to [27].

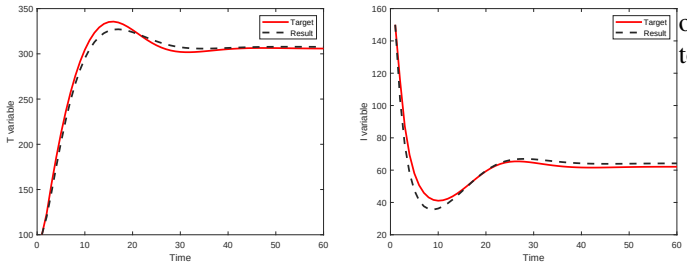
As shown in TABLE VIII, the results show that SpadePSO performs better than CLPSO and LatinPSO. The mean of SpadePSO is a half of CLPSO and one-fourth of LatinPSO. The performance of SpadePSO is similar to PSO. The best solution is very important because it is usually used to solve practical problems. The best solution shows that there is still a big gap between the model inferred from the three PSOs and the HIV model. So the population size is set to 400 to look for a better solution and the iteration number is also set to 3750. The mean of SpadePSO is one-sixth of LatinPSO and a half of PSO and CLPSO. What's more, the best solution of SpadePSO is significantly better than the other there PSOs. The best model inferred from SpadePSO is shown in Eqs. (25) and we compare the best model inferred from SpadePSO and the HIV model shown in Figs. 7.

$$\begin{aligned} \mathrm{d}T/\mathrm{d}t &= -0.2 * T - 0.32 * I + 81, \\ \mathrm{d}I/\mathrm{d}t &= -0.55 * I + 0.01 * T + 0.002T * I, \\ \mathrm{d}V/\mathrm{d}t &= -5.1 * V + 495 * I - 0.03 * T * I + 1. \end{aligned} \tag{25}$$

Tian et al. [32] got a better-inferred model in the experiment [32], but the population size is set 1000. For Ordinary differential equations models inference, large population size

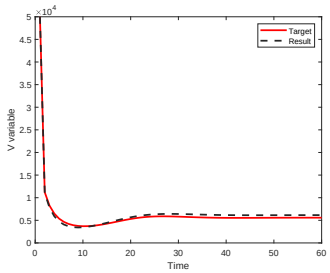
TABLE VIII
SPADEPSO VS. THE STATE-OF-THE-ART OPTIMIZATION ALGORITHMS ON ORDINARY DIFFERENTIAL EQUATIONS MODELS INFERENCE.

Population size.	40			400		
Stat. / Algorithm	Best	Mean	Std	Best	Mean	Std
PSO	1.18E+04	4.54E+04	6.08E+04	1.16E+04	2.36E+04	2.10E+04
CLPSO	1.77E+04	1.05E+05	9.32E+04	1.02E+04	2.57E+04	1.86E+04
LatinPSO	2.39E+04	2.04E+05	1.13E+05	2.12E+04	9.54E+04	7.76E+04
SpadePSO	1.17E+04	4.74E+04	6.40E+04	3.39E+03	1.49E+04	5.53E+03



(a)

(b)



(c)

Fig. 7. A comparison of time course data between the target and the inferred model on T, I and V variables.

seems to be necessary in recent studies. We think a better representation method of structure can reduce the demand of population size.

V. CONCLUSION

Inspired by the SPA in social science fields, Cui et al. [12] modelled the SPA in PSO scenarios to select the learning exemplars. We propose the SpadePSO with adaptive communication topology dynamic topology and compare different topologies to construct the best topology for SPA.

We evaluate the performance of SpadePSO with the CEC2014 benchmark suite on 10, 30, 50, and 100-dimensional functions in Experiment IV-C. The experimental results show that SpadePSO outperforms the state-of-the-art PSO variants, including OLPSO, GL-PSO, HCLPSO, PSO, TSLPSO and XPSO. In addition, we compared SpadePSO with L-SHADE which was one of the state-of-the-art optimization algorithms of the CEC2014 benchmark suite. There is an interesting phenomenon that L-shade, as a champion algorithm in CEC2014, performs still much better than PSO variants, including the proposed SpadePSO. The essential difference between DE and PSO is worth deliberate investigation. In Experiment IV-D, we evaluate the performance of SpadePSO on the spread spectrum radar polyphase coding design. The results show that SpadePSO performs better than other variants of PSO. Finally, we evaluate the performance of SpadePSO on the ordinary differential equations models inference in Experiment IV-E. The results show that SpadePSO performs better than LatinPSO which is specially designed for this problem. But in view of this problem, the performance of SpadePSO still has a lot of room to improve. Similar to the nonadjacency of 7 and 8 in binary coding, we think that the discontinuity of the representation method of the structure increases the difficulty of this problem. How to better represent the structure remains to be studied.

APPENDIX A

The CEC2014 benchmark suite are shown in TABLE IX.

TABLE IX
CEC2014 TEST SUITE.

	Function Name	Search Range	F(x^*)
Unimodal Functions	F1: Rotated High Conditioned Elliptic Function	$[-100, 100]^D$	100
	F2: Rotated Bent Cigar Function	$[-100, 100]^D$	200
	F3 Rotated Discus Function:	$[-100, 100]^D$	300
Simple Multimodal Functions	F4: Shifted and Rotated Rosenbrock's Function	$[-100, 100]^D$	400
	F5: Shifted and Rotated Ackley's Function	$[-100, 100]^D$	500
	F6: Shifted and Rotated Weierstrass Function	$[-100, 100]^D$	600
	F7: Shifted and Rotated Griewank's Function	$[-100, 100]^D$	700
	F8: Shifted Rastrigin's Function	$[-100, 100]^D$	800
	F9: Shifted and Rotated Rastrigin's Function	$[-100, 100]^D$	900
	F10: Shifted Schwefel's Function	$[-100, 100]^D$	1000
	F11 Shifted and Rotated Schwefel's Function	$[-100, 100]^D$	1100
	F12: Shifted and Rotated Katsuura Function	$[-100, 100]^D$	1200
	F13: Shifted and Rotated HappyCat Function	$[-100, 100]^D$	1300
	F14: Shifted and Rotated HGBat Function	$[-100, 100]^D$	1400
	F15: Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	$[-100, 100]^D$	1500
	F16: Shifted and Rotated Expanded Scaffer's F6 Function	$[-100, 100]^D$	1600
Hybrid Functions	F17: Hybrid Function 1 (N = 3)	$[-100, 100]^D$	1700
	F18: Hybrid Function 2 (N = 3)	$[-100, 100]^D$	1800
	F19: Hybrid Function 3 (N = 4)	$[-100, 100]^D$	1900
	F20: Hybrid Function 4 (N = 4)	$[-100, 100]^D$	2000
	F21: Hybrid Function 5 (N = 5)	$[-100, 100]^D$	2100
	F22: Hybrid Function 6 (N = 5)	$[-100, 100]^D$	2200
Composition Functions	F23: Composition Function 1 (N = 5)	$[-100, 100]^D$	2300
	F24 Composition Function 2 (N = 3)	$[-100, 100]^D$	2400
	F25: Composition Function 3 (N = 3)	$[-100, 100]^D$	2500
	F26: Composition Function 4 (N = 5)	$[-100, 100]^D$	2600
	F27: Composition Function 5 (N = 5)	$[-100, 100]^D$	2700
	F28: Composition Function 6 (N = 5)	$[-100, 100]^D$	2800
	F29: Composition Function 7 (N = 3)	$[-100, 100]^D$	2900
	F30: Composition Function 8 (N = 3)	$[-100, 100]^D$	3000

APPENDIX B

The columns show the best, mean and standard deviation of errors between the best fitness values found in each run and the true optimal values, respectively. The experimental results are shown in TABLE X.

REFERENCES

[1] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," *Evol. Comput.*, vol. 10, no. 2, Art. no. 2, Jun. 2002, doi: 10.1162/106365602320169811.

[2] M. Usman, W. Pang, and G. M. Coghill, "Inferring structure and parameters of dynamic system models simultaneously using swarm intelligence approaches," *Memetic Comput.*, vol. 12, no. 3, pp. 267–282, Sep. 2020, doi: 10.1007/s12293-020-00306-5.

TABLE X
THE RESULTS OF SPADEPSO ON THE CEC2014 BENCHMARK SUITE (D = 10, 30, 50 AND 100).

Dim.	10D			30D		
Stat. / Func.	Best	Mean	Std	Best	Mean	Std
1	7.75E+01	1.09E+04	1.34E+04	2.79E+04	2.11E+05	1.52E+05
2	5.32E-02	3.74E+01	5.00E+01	6.89E-05	1.11E+01	2.69E+02
3	9.59E-03	6.43E+01	9.44E+01	3.61E-01	1.28E+02	1.28E+02
4	2.05E-03	1.44E+00	3.71E+00	3.30E-02	3.91E+01	3.26E+01
5	0.00E+00	1.85E+01	5.40E+00	2.01E+01	2.02E+01	3.92E-02
6	1.13E-04	1.25E-02	2.17E-02	6.26E-01	2.11E+00	1.10E+00
7	3.25E-03	4.22E-02	2.28E-02	0.00E+00	2.87E-04	1.11E-03
8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9	1.53E+00	4.48E+00	1.68E+00	1.59E+01	4.35E+01	1.12E+01
10	0.00E+00	1.84E-01	6.55E-01	4.17E-02	4.91E-01	8.01E-01
11	1.87E+01	1.60E+02	1.21E+02	1.17E+03	1.91E+03	3.15E+02
12	5.60E-02	2.10E-01	5.90E-02	5.51E-02	2.39E-01	5.83E-02
13	2.54E-02	8.67E-02	3.37E-02	1.09E-01	2.27E-01	6.27E-02
14	2.61E-02	8.26E-02	3.43E-02	1.66E-01	2.36E-01	3.11E-02
15	3.99E-01	7.66E-01	2.15E-01	1.52E+00	4.32E+00	1.53E+00
16	2.07E-01	1.32E+00	4.28E-01	7.30E+00	9.36E+00	6.76E-01
17	1.88E+01	8.49E+02	7.01E+02	1.49E+03	8.82E+04	5.61E+04
18	4.24E+00	3.02E+02	6.94E+02	3.19E+01	1.49E+02	1.36E+02
19	3.91E-02	5.64E-01	3.66E-01	2.84E+00	4.71E+00	1.12E+00
20	2.44E+00	3.07E+01	4.62E+01	1.05E+02	8.56E+02	5.91E+02
21	6.62E+00	8.53E+01	8.13E+01	3.51E+03	3.86E+04	3.35E+04
22	1.98E-02	1.94E+00	5.12E+00	2.81E+01	1.91E+02	7.33E+01
23	3.29E+02	3.29E+02	2.59E-13	3.15E+02	3.15E+02	1.90E-12
24	1.00E+02	1.12E+02	2.93E+00	2.24E+02	2.25E+02	1.12E+00
25	1.04E+02	1.31E+02	2.18E+01	2.03E+02	2.06E+02	1.50E+00
26	1.00E+02	1.00E+02	2.96E-02	1.00E+02	1.00E+02	5.51E-02
27	9.45E-01	4.44E+01	1.06E+02	3.45E+02	3.97E+02	1.52E+01
28	3.44E+02	3.75E+02	2.25E+01	7.66E+02	8.88E+02	4.81E+01
29	2.33E+02	2.79E+02	3.56E+01	7.95E+02	9.52E+02	1.05E+02
30	4.96E+02	6.58E+02	1.01E+02	1.11E+03	1.87E+03	4.42E+02
Dim.	50D			100D		
Stat. / Func.	Best	Mean	Std	Best	Mean	Std
1	3.43E+05	6.31E+05	2.45E+05	1.39E+06	3.05E+06	8.78E+05
2	3.58E+00	1.70E+02	3.33E+02	3.29E+00	5.62E+02	8.03E+02
3	1.37E+02	1.68E+03	9.72E+02	2.68E+02	1.89E+03	1.30E+03
4	2.17E+01	8.59E+01	2.01E+01	1.11E+02	1.98E+02	3.67E+01
5	2.00E+01	2.03E+01	9.49E-02	2.00E+01	2.02E+01	1.68E-01
6	2.31E+00	9.46E+00	3.04E+00	4.03E+01	5.55E+01	6.13E+00
7	0.00E+00	5.76E-03	7.24E-03	0.00E+00	1.39E-03	3.85E-03
8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.58E-02	2.34E-01
9	3.98E+01	9.22E+01	2.33E+01	1.84E+02	2.79E+02	3.66E+01
10	1.13E-01	1.22E+01	3.52E+01	2.52E-01	1.74E+01	4.69E+01
11	3.07E+03	4.22E+03	4.34E+02	9.30E+03	1.11E+04	7.00E+02
12	1.03E-01	2.13E-01	5.82E-02	1.75E-01	2.87E-01	7.33E-02
13	2.22E-01	3.30E-01	5.84E-02	2.84E-01	4.13E-01	5.10E-02
14	1.94E-01	2.69E-01	2.91E-02	2.50E-01	3.15E-01	2.60E-02
15	4.84E+00	9.05E+00	2.20E+00	2.19E+01	3.69E+01	8.04E+00
16	1.42E+01	1.77E+01	9.49E-01	3.89E+01	4.05E+01	6.24E-01
17	1.83E+04	1.32E+05	1.07E+05	2.64E+05	8.04E+05	3.24E+05
18	6.28E+01	1.52E+02	5.10E+01	2.51E+02	3.99E+02	1.48E+02
19	8.37E+01	1.34E+01	3.01E+00	2.94E+01	7.41E+01	1.81E+01
20	3.44E+02	1.29E+03	8.34E+02	1.62E+03	4.50E+03	1.94E+03
21	2.43E+04	1.72E+05	1.68E+05	1.29E+05	5.03E+05	3.74E+05
22	1.59E+02	6.93E+02	1.85E+02	1.12E+03	1.88E+03	3.13E+02
23	3.44E+02	3.44E+02	1.03E-12	3.48E+02	3.48E+02	2.93E-11
24	2.56E+02	2.60E+02	3.60E+00	3.57E+02	3.62E+02	1.98E+00
25	2.07E+02	2.14E+02	2.76E+00	2.46E+02	2.56E+02	5.10E+00
26	1.00E+02	1.06E+02	2.24E-02	1.00E+02	1.96E+02	1.94E-01
27	4.25E+02	5.45E+02	8.30E+02	5.09E+02	1.54E+03	4.88E+02
28	1.17E+03	1.52E+03	2.01E+02	3.46E+03	5.26E+03	7.65E+02
29	9.27E+02	1.21E+03	1.85E+02	1.36E+03	1.74E+03	3.06E+02
30	8.55E+03	1.05E+04	9.50E+02	7.87E+03	9.68E+03	8.32E+02

[3] H. Wang, Y. Jin, and J. Doherty, "Committee-Based Active Learning for Surrogate-Assisted Particle Swarm Optimization of Expensive Problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, Art. no. 9, Sep. 2017, doi: 10.1109/TCYB.2017.2710978.

[4] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43, doi: 10.1109/MHS.1995.494215.

[5] Y. Li, Z.-H. Zhan, S. Lin, J. Zhang, and X. Luo, "Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems," *Inf. Sci.*, vol. 293, pp. 370–382, Feb. 2015, doi: 10.1016/j.ins.2014.09.030.

[6] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary Particle Swarm Optimization With Bottleneck Objective Learning Strategy for Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, Art. no. 4, Aug. 2019, doi: 10.1109/TEVC.2018.2875430.

[7] M. R. Tanweer, S. Suresh, and N. Sundararajan, "Self regulating particle swarm optimization algorithm," *Inf. Sci.*, vol. 294, pp. 182–202, Feb. 2015, doi: 10.1016/j.ins.2014.09.053.

[8] J. Lorenz, H. Rauhut, F. Schweitzer, and D. Helbing, "How social influence can undermine the wisdom of crowd effect," *Proc. Natl. Acad. Sci.*, vol. 108, no. 22, Art. no. 22, May 2011, doi: 10.1073/pnas.1008636108.

[9] J. P. Simmons, L. D. Nelson, J. Galak, and S. Frederick, "Intuitive Biases in Choice versus Estimation: Implications for the Wisdom of Crowds," in *J. Consum. Res.*, vol. 38, no. 1, pp. 1–15, Jun. 2011, doi: 10.1086/658070.

[10] K.-Y. Chen, L. R. Fine, and B. A. Huberman, "Eliminating Public Knowledge Biases in Information-Aggregation Mechanisms," *Manag. Sci.*, vol. 50, no. 7, pp. 983–994, Jul. 2004, doi: 10.1287/mnsc.1040.0247.

[11] D. Prelec, H. S. Seung, and J. McCoy, "A solution to the single-question crowd wisdom problem," *Nature*, vol. 541, no. 7638, pp. 532–535, Jan. 2017, doi: 10.1038/nature21054.

[12] Q. Cui et al., "Surprisingly Popular Algorithm-based Comprehensive Adaptive Topology Learning PSO," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 2603–2610, doi: 10.1109/CEC.2019.8790002.

[13] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal Learning Particle Swarm Optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, Art. no. 6, Dec. 2011, doi: 10.1109/TEVC.2010.2052054.

[14] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, Art. no. 3, Jun. 2006, doi: 10.1109/TEVC.2005.857610.

[15] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm Evol. Comput.*, vol. 24, pp. 11–24, Oct. 2015, doi: 10.1016/j.swevo.2015.05.002.

[16] G. Xu et al., "Particle swarm optimization based on dimensional learning strategy," *Swarm Evol. Comput.*, vol. 45, pp. 33–51, Mar. 2019, doi: 10.1016/j.swevo.2018.12.009.

[17] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, p. 3, 1998.

[18] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Phys. Lett. A*, vol. 263, no. 4–6, pp. 341–346, Dec. 1999, doi: 10.1016/S0375-9601(99)00757-4.

[19] Y. Gong and J. Zhang, "Small-world particle swarm op-

timization with topology adaptation," in *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference - GECCO '13*, Amsterdam, The Netherlands, 2013, p. 25, doi: 10.1145/2463372.2463381.

[20] T. Qiu, B. Li, X. Zhou, H. Song, I. Lee, and J. Lloret, "A Novel Shortcut Addition Algorithm With Particle Swarm for Multisink Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 16, no. 5, pp. 3566–3577, May 2020, doi: 10.1109/TII.2019.2925023.

[21] Q. Liu, S. Du, B. J. van Wyk, and Y. Sun, "Niching particle swarm optimization based on Euclidean distance and hierarchical clustering for multimodal optimization," *Nonlinear Dyn.*, vol. 99, no. 3, pp. 2459–2477, Feb. 2020, doi: 10.1007/s11071-019-05414-7.

[22] M. D. Lee, I. Danileiko, and J. Vi, "Testing the ability of the surprisingly popular method to predict NFL games.," Judgm. Decis. Mak., vol. 13, no. 4, pp. 322–333, 2018.

[23] T. Luo and Y. Liu, "Machine Truth Serum," ArXiv190913004 Cs Stat, Sep. 2019, Accessed: Mar. 11, 2021. [Online]. Available: http://arxiv.org/abs/1909.13004.

[24] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Jun. 2008, pp. 1128–1134. doi: 10.1109/CEC.2008.4630938.

[25] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," *Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Rep. Nanyang Technol. Univ. Singap.*, 2013.

[26] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, Art. no. 1, Mar. 2011, doi: 10.1016/j.swevo.2011.02.002.

[27] P. N. Suganthan, N. Hansen, J. J. Liang, and K. Deb, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," *Nat. Comput.*, pp. 341–357, 2005.

[28] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 1128–1134, doi: 10.1109/CEC.2008.4630938.

[29] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, Jul. 2014, pp. 1658–1665, doi: 10.1109/CEC.2014.6900380.

[30] Y.-J. Gong et al., "Genetic Learning Particle Swarm Optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, Art. no. 10, Oct. 2016, doi: 10.1109/TCYB.2015.2475174.

[31] X. Xia et al., "An expanded particle swarm optimization based on multi-exemplar and forgetting ability," Inf. Sci., vol. 508, pp. 105–120, Jan. 2020, doi: 10.1016/j.ins.2019.08.065.

[32] X. Tian, W. Pang, Y. Wang, K. Guo, and Y. Zhou, "LatinPSO: An algorithm for simultaneously inferring structure and parameters of ordinary differential equations models," *Biosystems*, vol. 182, pp. 8–16, Aug. 2019, doi: 10.1016/j.biosystems.2019.05.006.

[33] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Inf. Sci.*, vol. 423, pp. 172–186, Jan. 2018, doi: 10.1016/j.ins.2017.09.053.

[34] J. Brest, M. S. Maucec, and B. Boskovic, "iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 1188–1195, doi: 10.1109/CEC.2016.7743922.

[35] N. Mladenović, J. Petrović, V. Kovačević-Vujčić, and M. Čangalović, "Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search," *Eur. J. Oper. Res.*, vol. 151, no. 2, pp. 389–399, Dec. 2003, doi: 10.1016/S0377-2217(02)00833-0.

[36] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution Using a Neighborhood-Based Mutation Operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009, doi: 10.1109/TEVC.2008.2009457.

[37] M. L. Dukic and Z. S. Dobrosavljevic, "A method of a spread-spectrum radar polyphase code design," *IEEE J. Sel. Areas Commun.*, vol. 8, no. 5, pp. 743–749, Jun. 1990, doi: 10.1109/49.56381.

[38] M. Heinonen, C. Yildiz, H. Mannerström, J. Intosalmi, and H. Lähdesmäki, "Learning unknown ODE models with Gaussian processes," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholmsmässan, Stockholm Sweden, Jul. 2018, vol. 80, pp. 1959–1968, [Online]. Available: http://proceedings.mlr.press/v80/heinonen18a.html.