Heliyon 10 (2024) e31631

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Population interaction network in representative gravitational search algorithms: Logistic distribution leads to worse performance

Haotian Li [a], Yifei Yang [b], Yirui Wang [c,d], Jiayi Li [a], Haichuan Yang [e], Jun Tang [f,*], Shangce Gao [a,*]

[a] Faculty of Engineering, University of Toyama, Toyama-shi, 930-8555, Japan
[b] Graduate School of Science and Technology, Hirosaki University, Hirosaki-shi, 036-8561, Japan
[c] Faculty of Electrical Engineering and Computer Science, Ningbo University, Zhejiang 315211, China
[d] Zhejiang Key Laboratory of Mobile Network Application Technology, Zhejiang 315211, China
[e] Graduate School of Technology, Industrial and Social Sciences, Tokushima University, Tokushima, 770-8506, Japan
[f] Wicresoft Co Ltd, 13810 SE Eastgate Way, Bellevue, WA 98005, USA

ARTICLE INFO

Keywords:
Population interaction network
Meta-heuristic algorithms
Gravitational search algorithm
Complex network
Cumulative distribution

ABSTRACT

In this paper, a novel study on the way inter-individual information interacts in meta-heuristic algorithms (MHAs) is carried out using a scheme known as population interaction networks (PIN). Specifically, three representative MHAs, including the differential evolutionary algorithm (DE), the particle swarm optimization algorithm (PSO), the gravitational search algorithm (GSA), and four classical variations of the gravitational search algorithm, are analyzed in terms of inter-individual information interactions and the differences in the performance of each of the algorithms on IEEE Congress on Evolutionary Computation 2017 benchmark functions. The cumulative distribution function (CDF) of the node degree obtained by the algorithm on the benchmark function is fitted to the seven distribution models by using PIN. The results show that among the seven compared algorithms, the more powerful DE is more skewed towards the Poisson distribution, and the weaker PSO, GSA, and GSA variants are more skewed towards the Logistic distribution. The more deviation from Logistic distribution GSA variants conform, the stronger their performance. From the point of view of the CDF, deviating from the Logistic distribution facilitates the improvement of the GSA. Our findings suggest that the population interaction network is a powerful tool for characterizing and comparing the performance of different MHAs in a more comprehensive and meaningful way.

1. Introduction

Optimization problems play a vital role in scientific and engineering researches [1–6]. Heuristic methods provide a way to rapidly arrive at a workable answer when finding the ideal solution to an issue is challenging or impossible, such as NP-hard problems [7–9]. These techniques deliver workable answers in a respectable amount of time and space, but they cannot ensure the

* Corresponding authors.
E-mail addresses: juntang@wicresoft.com (J. Tang), gaosc@eng.u-toyama.ac.jp (S. Gao).

https://doi.org/10.1016/j.heliyon.2024.e31631
Received 15 February 2024; Received in revised form 20 May 2024; Accepted 20 May 2024

Available online 22 May 2024

Nomenclature	Description
PIN	Population interaction network
MHAs	Meta-heuristic algorithms
DE	Differential evolution algorithm
GSA	Gravitational search algorithm
PSO	Particle swarm optimization
CDF	Cumulative distribution function
QBGSA	Quantum-inspired gravitational search algorithm
IGSA	An improved gravitational search algorithm
DPDE	Directional permutation differential evolution algorithm
PV	Photovoltaic
MaOEA	A many-objective evolutionary algorithm
SIS	Spatial information sampling algorithm
AGPSO	Adaptive replacement strategy-incorporated particle swarm optimizer
GGSA	Adaptive gbest-guided GSA
CGSA-M	Multiple chaos embedded GSA
HGSA	Hierarchical GSA with an effective gravitational constant
ALGSA	Aggregative learning GSA with self-adaptive gravitational constants
BSO	Brain storm optimization
SE	Spherical evolution
IEEE CEC	IEEE congress on evolutionary computation

finest answer. A form of generalized heuristic strategy that can be used to solve a wider variety of issues than just the conditions of a particular problem is known as a meta-heuristic method [10]. These methods are particularly useful when the optimization problem is complex and finding an optimal solution is not possible in polynomial time. While meta-heuristics do not provide an exact solution, they offer a good approximation. Consequently, they can be an effective instrument for resolving actual optimization issues, especially those that are time-consuming or impossible to solve numerically. Meta-heuristic methods are efficient techniques for solving optimization problems [11]. Even in circumstances where it is impossible to find the ideal answer, they effectively locate workable solutions [12]. However, the characteristics of NP-hard problems are difficult to obtain, thus researchers can only solve these problems by designing various meta-heuristics [13]. This scenario does not invariably have a favorable implication. Researchers frequently propose enhancements to established methodologies and conduct comparative analyses against non-improved versions or similar approaches to demonstrate superior convergence and properties. There are currently scholars in the meta-heuristic community who contend that applying unique techniques or newly suggested paradigms to well-known issues is not a productive study field [14–17]. The "alchemical dilemma" [18] refers to the scenario where individuals keep improving their meta-heuristics but are unable to describe why they are doing so. It has further resulted in the choice and development of algorithms that lack a theoretical foundation and are heavily reliant on scholar experience, eventually increasing the time and financial costs. So, are the ever-emerging meta-heuristics really useless? The answer is NO. For example, in 2021, Gao et al. used directional permutation differential evolution (DPDE) algorithm to optimize a solar photovoltaic (PV) model problem [19]. In 2022, Lei et al. solved a protein structure prediction problem using a many-objective evolutionary algorithm (MaOEA) [20]. In 2022, Yang et al. used spatial information sampling (SIS) algorithm to optimize a wave energy converter location problem [21]. In 2022, Lei et al. used adaptive replacement strategy-incorporated particle swarm optimizer (AGPSO) algorithm to optimize a wind farm layout problem [22]. These applications demonstrate that different types of algorithms make significance for real-world problems [23]. In the above examples, it can be seen that different algorithms are applied to different optimization problems. For different optimization problems, it becomes very meaningful to help the researchers filter out suitable algorithms.

Meanwhile, in the real world, there are complex relationships between people or things that are difficult to express in simple terms. Through collaborative communication, different types of feature networks are built, collectively called complex networks [24]. Dynamism, randomness, and intricacy are characteristics of complex networks. They can be seen in the connections between components and edges in the network [25]. A large number of existing studies have shown that various network structures can greatly affect network properties such as node degree, shortest path, and clustering coefficient [26,27]. It is the discovery of these features that can lead to a better understanding of the inner workings of the network. All in all, complex networks are considered to be an effective tool for describing and explaining elusive phenomena arising in the real world and are used in the theoretical study of algorithmic topologies [28–30]. In the meta-heuristic algorithm (MHA), individuals interact frequently during the retrieval process so that a certain structure is formed [31,32]. The efficiency of an algorithm and its search habits are greatly influenced by different population structures. Recently, several structures have been studied for meta-heuristic algorithms, such as cellular structures, distributed structures, and hierarchical structures [33]. Cellular structures allow the population to communicate only with their peers, such as cellular particle swarm optimization (PSO) [34]. Distributed structures have multiple populations that enhance the population diversity by exchanging information between sub-populations [35]. Hierarchical structures are utilized to assign individuals into different hierarchies. Qi et al. designed an interactive system network to analyze the effect of structure on PSO [36]. Finally, it was found that the interactive system network can improve the performance of algorithms [37]. In the network-centric unit of measure, a degree is used for differential evolution (DE) [38,39]. Numerous studies have shown that the structure of algorithms affects the interactions of individuals. However, the structural characteristics of algorithms are not clear for the effect of performance. Therefore, analyzing the structural characteristics of algorithms is an important topic to better understand their performance.

In recent years, researchers have applied gravitational search algorithm (GSA) algorithms to real optimization problems and achieved good results. For example, Ji et al. proposed a quantum-inspired gravitational search algorithm (QBGSA) and applied the algorithm to the optimization problem of wind farm unit combinations [40]. Chen et al. proposed an improved gravitational search algorithm (IGSA) and applied the algorithm to the water turbine regulation system optimization problem [41]. Despite the numerous excellent variants of GSA proposed, there has been a scarcity of analyses based on the structural characteristics of the GSA. Hence, this paper focuses on investigating the structural features of five classical GSA algorithms: the original GSA [42], adaptive gbest-guided gravitational search algorithm (GGSA) [43], multiple chaotic maps embedded gravitational search algorithm (CGSA-M) [44], hierarchical gravitational search algorithm with an effective gravitational constant (HGSA) [45], and aggregative learning GSA with self-adaptive gravitational constants (ALGSA) [46]. To conduct a deeper investigation into the GSA, this study also introduces different types of algorithms, such as DE and PSO, for comparison. In this research, we construct the structural features and degree distributions of the above seven MHA's Population Interaction Networks (PINs). The degree distributions of the PINs are modeled using seven distribution models. The experiments are conducted on 29 IEEE CEC2017 benchmark functions to compare the structural characteristics and distribution models of different algorithms.

The main contributions of this paper are summed up as follows:

1) For the first time, we use the PIN to analyze three different types of MHAs: the evolutionary mechanisms-based DE, the swarm intelligence theory-based PSO, and the physical law-inspired GSA. The experimental results show that the cumulative distribution of node degree for DE conforms to a Poisson distribution, and the cumulative node degree for PSO and GSA conforms to a Logistic distribution.
2) For the first time, we use the PIN to analyze classical GSA and its four variants. It is concluded that the more one deviates from the Logistic distribution, the higher the performance is for variant methods that improve the original GSA structure.
3) Experimental results on 29 functions of IEEE CEC2017 show that the algorithm conforming to the Poisson distribution is superior to the algorithm conforming to the Logistic distribution, thus concluding that the Poisson distribution is more suitable for improving the MHA than the Logistic distribution. The validity of the conclusions is verified on 22 real optimization problems.
4) In this study, by analyzing three different types of algorithms, two network distribution structures, Poisson distribution and Logistic distribution, are observed, and these two distribution structures are compared and analyzed. This study provides a new perspective for the theoretical analysis of MHA from the perspective of complex systems and, at the same time, provides valuable insights for the improvement and design of MHA.

The rest of this paper is organized as follows: The evolving processes of seven algorithms are shown in Section 2. We present the PIN in Section 3. In Section 4, seven algorithms are evaluated using 29 IEEE CEC2017 benchmark functions. Their performance is compared, and their cumulative distribution functions of degree of nodes in the PIN are analyzed. Finally, discussions and conclusions are given in Section 5.

2. Related work

In this session, the search process and updating methods of three different types of MHAs will be presented, namely DE [47] inspired by evolutionary mechanisms, PSO [48] inspired by group intelligence theory, and GSA inspired by laws of physics. Four mainstream variants of GSA are also presented in detail, namely CGSA-M, ALGSA, which changes the original parameters, and GGSA, HGSA, which changes its original structure.

2.1. DE, PSO, and GSA

• **Differential evolution:** In 1997, DE as a successful evolutionary method was first proposed. Afterwards, many distinct differential evolution variations have been put forth, producing noteworthy findings on a variety of issues. DE comprises three primary operations: mutation, crossover, and selection. Based on the existing progenitor, DE creates new mutations in the population with iterations. As seen in Eq. (1), DE employs a straightforward mutation strategy in which three individuals, including $\vec{x}_{r0}$, $\vec{x}_{r1}$, and $\vec{x}_{r2}$, are chosen at random to produce a new one, $\vec{v}_i$. Then, the mutation individual crosses over with the progenitor to produce a new one $\vec{u}_{i,j}$ according to Eq. (2), meaning the crossover operation. Finally, DE uses a greedy selection technique to make sure that a newly produced individual $\vec{x}'_i$ is better than its progenitor, as shown in Eq. (3). In this selection method, an individual with a higher fitness value is chosen to be a part of the following population.

$$\vec{v}_i = x_{r0} + F \cdot (\vec{x}_{r1} - \vec{x}_{r2}) \tag{1}$$

$$\vec{u}_{i,j} = \begin{cases} \vec{v}_{i,j}, & rand(0,1) < CR \ or \ j = j_{rand} \\ \vec{x}_{i,j}, & otherwise \end{cases} \tag{2}$$

$$\vec{x}'_i = \begin{cases} \vec{u}_i, & f(\vec{u}_i) < f(\vec{x}_i) \\ \vec{x}_i, & otherwise \end{cases} \tag{3}$$

• **Particle swarm optimization:** PSO was first proposed in 1995. Numerous variations have been presented to produce impressive outcomes for various optimization problems. The concept and design of the PSO algorithm are motivated by the social behavior of fish schools and flocks of birds. In the wild, a flock of birds follows a leader who is in the best position to reach food as they fly through the air. In order to solve optimization problems, the social behavior of birds can be transformed into algorithmic operations. In PSO, each particle represents a potential solution. There are personal-best particle $p\vec{best}_i$ and global-best particle $g\vec{best}_i$ leading each particle's movement according to Eqs. (4) and (5). The population explores the area in the specified dimensions and finds the best solution to the current issue. $C1$ and $C2$ are acceleration constants, which take equal values $C1 = C2$.

$$\vec{v}_i = \vec{v}_i + C_1 \cdot rand\,(i) \cdot (p\vec{best}_i - \vec{x}_i) + C_2 \cdot rand\,(i) \cdot (g\vec{best}_i - \vec{x}_i) \tag{4}$$

$$\vec{x}_i' = \vec{x}_i + \vec{v}_i \tag{5}$$

• **Gravitational search algorithm:** GSA based on newtonian gravity theory, was introduced in 2009. Its variants have been developed widely and have shown outstanding outcomes for a variety of problems. In GSA, the gravitational force $\vec{F}_{ij}$ is generated by two particles' masses M and distance R, as shown in Eq. (6). The resultant force $\vec{F}_i$ produces an acceleration to update the velocity and position of the particle $\vec{x}_i$ according to Eqs. (8) and (9). In Eq. (7), where G_0 is the starting number, is a constant, and G_t is a gravitational constant linked to iteration t. t and T denote the number of current iterations and the maximum iteration, respectively.

$$\vec{F}_{ij} = G_t \cdot \frac{M_i \cdot M_j}{R^2} \cdot \left(\vec{x}_j - \vec{x}_i\right) \tag{6}$$

$$G_t = G_0 \cdot e^{-\alpha \frac{t}{T}} \tag{7}$$

$$\vec{v}_i = \vec{v}_i \cdot rand\,(i) + \frac{\vec{F}_i}{M_i} \tag{8}$$

$$\vec{x}_i' = \vec{x}_i + \vec{v}_i \tag{9}$$

2.2. GSA and its variants

GSA variants can be broadly classified into two categories. One is adjusting parameters to better balance exploitation and exploration, such as CGSA-M and ALGSA. The other is changing the population structure, and thus the way particles interact with each other is different, such as GGSA and HGSA.

• **Change original parameters:** 1) "CGSA-M": CGSA-M uses a memory-based strategy for adaptive selection of different chaotic maps. Chaotic operators as a means of improving the performance of GSA are commonly known as chaotic local search. In CGSA-M, the best chaotic map is selected from 12 different chaotic maps to act on the current best particle based on previous success and failure rates, and the latest generated particle is compared with the current best particle to determine which one is better. Initially, the selected probability of each chaotic map is 1/12. As the iteration time goes on, the selection probability is gradually updated. Finally, each particle's position is updated by Eqs. (8), (9), and (10). The search space's top and lower bounds are U and L, respectively. z is a chaotic variable.

$$\vec{x}_i = \vec{x}_i + rand\,(i) \cdot (U - L) \cdot (z - 0.5) \tag{10}$$

2) "ALGSA": ALGSA proposed an adaptive tunable gravitational constant strategy. With the help of an adjustable gravitational constant, ALGSA improves the algorithm's search capabilities and guards against populations stuck in local optima. The new gravity boosts particles' search efficiency. An adaptive gravitational constant that is tailored to each particle improves the search performance of the population. Each particle's position is updated by Eqs. (8), (9), (11), and (12). When $Counter$ exceeds θ and $rand$ exceeds p, the individual needs to expand the gravitational constant its exploratory capacity is enhanced. Otherwise, the gravitational constant of the original GSA is maintained. r is the gravitational constant and is used to enhance the exploration ability of individuals. In Eq. (12), the total gravitational force F is composed of several component forces Y.

$$G_{i(t)} = \begin{cases} G_{i(t)} \cdot r_{i(t)}, & if\ Counter > \theta\ \&\ rand < p \\ G_{i(t)}, & otherwise \end{cases} \tag{11}$$

$$F_i^d = \sum_{j=1}^{K} Y_i\,(j) \tag{12}$$

• **Change original structure:** 3) "GGSA": Based on the original GSA, GGSA adds the best particle $g\vec{best}$. The best particle $g\vec{best}$ can act on each particle and lead them to move towards it according to Eq. (13), providing a social intelligence guide. Moreover, the effect of $g\vec{best}$ on the current particles is independent of their masses, thus it can be seen as an external force independent of gravitational rules, which effectively prevents particles from trapping into local optima. c_1 and c_2 are linear coefficients that control how $g\vec{best}$ works during the development phase by adjusting their values.

$$\vec{v}_i = \vec{v}_i \cdot rand\,(i) + c_1 \cdot \frac{\vec{F}_i}{M_i} + c_2 \cdot \left(g\vec{best} - \vec{x}_i\right) \tag{13}$$

4) "HGSA": HGSA introduces a globally optimal individual to the original GSA and places this individual at the top layer to act as a guide to the current population. The traditional GSA views the $Kbest$ particles as facilitators of population evolution. It constructsa two-layer structure. Based on these characteristics, HGSA designs a three-layer hierarchy to improve the $Kbest$ particles and population. Hierarchical interaction among three layers effectively enhances particles' movement, thus the performance of algorithm is greatly reinforced. Population, $Kbest$ particles and global optimal particle are arranged in different layers. Particles velocity are updated as shown in Eq. (14). g_{opt} is the globally optimal individual. w_1 and w_2 are two weighting coefficients, respectively, whose values are adjusted to ensure an effective transition between the exploration and utilization capabilities of GSA in different search phases. Particles are finally updated by Eq. (9).

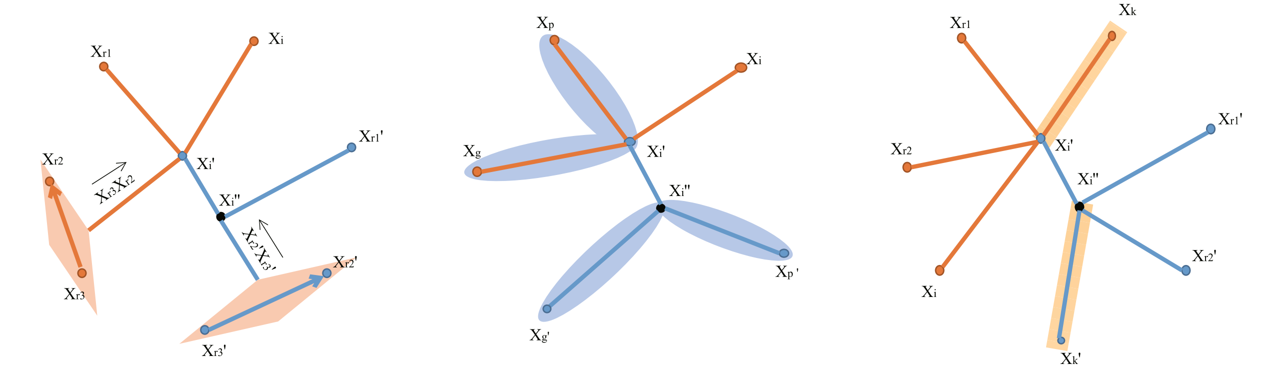
$$\vec{v}_{i\in K_b} = \vec{v}_i \cdot rand\,(i) + w_1 \cdot \frac{\vec{F}_i}{M_i} + w_2 \cdot \left(\vec{g}_{opt} - \vec{x}_{i\in K_b}\right) \tag{14}$$

3. Population interaction network

In this section, in order to better analyze the differences of different MHAs from the perspective of theoretical analysis, we use complex network theory as the basis for an in-depth analysis of the PINs of seven algorithms for three different types of MHAs, which are DE, PSO, GSA, and four variants of GSA.

In the iteration process of algorithm, there is the information interaction among individuals. It has been proven that the reasonable individual information interaction can enhance the performance of the algorithm in the optimization process [49–51]. When individuals with higher solution quality interact frequently with other individuals in the population, the population structure of the algorithm could have indicated that the Power-law distribution is generated after statistical and fitting analysis [50]. When the probability of individual interactions is more even, both good or poor, individuals have a similar number of information interactions. The population structure of the algorithm exhibits a Poisson distribution. However, analyzing them from a theoretical viewpoint remains a challenging task. Gao et al. proposed the PIN to efficiently evaluate DE [49]. The PIN keeps track of information about interactions between freshly created individuals and their parents, revealing patterns in how populations are created and enabling the accurate analysis of community traits. Wang et al. implemented experiments in brain storm optimization (BSO) with different dimensions and parameters. Results show that the BSO with Power-law distribution has the strongest performance [50]. Li et al. used PIN to examine DE and its variations. The algorithm with the Power-law distribution is superior than that with the Poisson distribution, it is concluded [51]. The three studies above have the same limitations. First, the analyzed algorithms are all based on evolutionary mechanisms and are of a relatively homogeneous kind. There is no comparative analysis of other different kinds of algorithms. Second, the conclusions are drawn by relying only on the experimental results of the standard function test set, and the validity of the conclusions is not verified on real-world problems. Zhang et al. utilized PIN theory-based MHAs for training dendritic neurons in time series prediction problems. The obtained results validated the superiority of algorithms based on Power-law distribution over those based on Poisson distribution, affirming the correctness of the conclusions [52]. Yang et al. proposed an enhanced version of the spherical evolution (SE) algorithm based on PIN theory and verified the effectiveness of the new algorithm in the context of the wind farm layout optimization problem. These demonstrations further affirm the universality of PIN theory in addressing practical optimization problems [53]. The two studies mentioned above also have the same limitations. First, the number of distribution functions provided in the fitting comparison of algorithm node degrees is small, which cannot accurately reflect the cumulative distribution of the algorithms. Second, the conclusions are drawn only on a single problem, and it is not possible to verify whether the conclusions are generalizable.

Inspired by the above research, we will conduct a more in-depth study on MHAs. First of all, we will try to analyze in depth the PINs of seven algorithms for three different types of MHAs, which are DE, PSO, GSA, and four variants of GSA, respectively. The PIN keeps track of how recently created individuals interact with their parents. In this network, a node represents an individual, and an edge represents an exchange of information between two individuals. The recorded data in PIN provide a population generation pattern and enable an efficient analysis of the population's traits. Fig. 1 shows PINs of DE, PSO, and GSA. In Fig. 1, X_i'' is an offspring of X_i'. X_i' is an offspring of X_i. Orange lines represent the network connections of X_i' to its parents. Blue lines represent the network connections of X_i'' to its parents. In Fig. 1(a), the orange rhombus represents the search operation Eq. (1) of DE. In Fig. 1(b), the blue ellipse represents the search operation Eq. (4) of PSO. In Fig. 1(c), the yellow rectangle represents the X_k individuals in the $Kbest$ set acting as guides for the current individuals in the GSA. As can be seen from Eq. (2), the DE greedy selection strategy, compared to PSO and GSA, allows the algorithm to further filter out information about better individuals. As can be seen from Eq. (4), PSO uses information from $\vec{pbest}$ and $\vec{gbest}$ individuals compared to GSA, and the addition of outstanding individuals acts as a guide to the population. The variations between each type of algorithm's exploitation and exploration powers are discovered through the comparing of three search processes. In order to quantify the frequency of information exchange between individuals during the updating process, we introduce the numerical value of the degree to reflect the number of information interactions among individuals. In DE, a new individual is generated through three operations of mutation, crossover, and selection, which results in four opportunities for information interaction. Thus, there are a total of five-times information exchange if this new individual engages as a parent in the following iterations, and its degree is 5. In PSO, two optimal individuals $Pbest$ and $Gbest$ guide the generation of new individual, resulting in two opportunities for information interaction. If the newly generated individual participates in the subsequent iteration and successfully generates another individual, it will engage in a total of three information interactions. As a result, the degree of this individual in the PIN will be equal to 3. In GSA, an individual is guided by $Kbest$ individuals to generate



(a) The population interaction network of DE.

(b) The population interaction network of PSO.

(c) The population interaction network of GSA.

Fig. 1. The population interaction networks of DE, PSO, and GSA. (a) – (c) show schematic diagrams showing the population interaction networks of DE, PSO, and GSA.

new individuals, which means K opportunities for information interaction. Therefore, its degree is K and the value of K decreases gradually with iterations.

For GSA variants, they can be broadly classified into two categories. One category is to change the way information is exchanged between particles by introducing new particles, and the structure of the original GSA is changed due to the addition of new particles. Examples of such algorithms include GGSA and HGSA. Conversely, another category of algorithms alters the frequency of information exchanges among particles by introducing operators or adjusting parameters while retaining the original GSA structure. Representative algorithms in this category include CGSA-M and ALGSA. The PINs of GSA variants are built using the same method as the original GSA. It is significant to point out that the PIN only keeps track of the particle who engages with others. Their nodes and edges are recorded. Any operation's precise shape is not documented. Therefore, PIN can be used to analyze GSA variants that change the initial structure.

4. Fitting methods and experimental results

In this section, we first evaluate the performance of the seven algorithms on top of the IEEE CEC2017 test set. Secondly, we analyze and compare the relationship between the PINs of the algorithms and the distribution function. Finally, the validity of the conclusions is again verified on problems with 22 real-world problems.

The effectiveness of MHAs is evaluated using the following evaluation criteria:

1) The performance comparison among DE, PSO, GSA, and GSA variants is denoted by "+", "=", and "−" to indicate whether they perform better, similarly, and worse, respectively. The assessment is based on the Wilcoxon rank-sum test. The Wilcoxon rank-sum test yields $W/T/L$ as a measure of the statistical importance of performance disparities between two algorithms. The efficacy of various algorithms on benchmark functions is compared using this non-parametric measure. The number of functions where one algorithm performs considerably better, not significantly different, and significantly worse than the other algorithm is represented by the statistical result W, T, and L.

2) The Friedman test is used to compare the medians of results in two or more algorithms. A smaller ranking $Rank$ indicates the better performance of algorithm on test set.

3) The performance of an algorithm is shown using the box-and-whisker diagram and convergence graph. The components of box-and-whisker diagram stand for the extreme values, the first and third quartiles, the middle, the maximum, and the minimum values. If there is a significant difference between the maximum and minimum numbers, the algorithm's effectiveness is unstable. The convergence graph shows the convergence trait of the algorithm in order to observe the change of optimal solution during iterations.

4.1. Experiment setup

The parameter settings of the seven algorithms in this experiment follow the recommended parameter configuration in the literature, which by default maximizes the performance of the algorithms when used. The number of iterations with the same termination condition is set to 3000. To ensure the accuracy of the experimental results, each algorithm was independently performed 51 times on each function to obtain statistical results. The seven distribution functions use Statistics and Learning Toolbox in Matlab. Finally, MATLAB is used to conduct all tests on a computer with an Intel(R) Core(TM) i7-11700 processor running at 2.50 GHz and 32 GB of RAM.

Table 1
Results of DE, PSO, and GSA on IEEE CEC2017 functions according to Wilcoxon rank-sum test.

	DE		PSO			GSA		
	mean	std	mean	std		mean	std	
F1	8.081E-15	7.655E-15	1.098E+08	3.980E+07	+	1.652E+03	7.365E+02	+
F2	2.743E+01	2.299E+01	4.115E+03	1.204E+03	+	8.393E+04	6.794E+03	+
F3	3.749E+01	2.823E+01	3.023E+02	8.666E+01	+	1.367E+02	1.493E+01	+
F4	1.785E+02	1.053E+01	1.359E+02	3.092E+01	-	2.279E+02	1.692E+01	+
F5	2.548E-08	3.154E-08	2.729E+01	9.808E+00	+	5.104E+01	4.024E+00	+
F6	2.106E+02	9.184E+00	1.702E+02	3.478E+01	-	8.551E+01	1.224E+01	-
F7	1.794E+02	1.186E+01	1.061E+02	2.705E+01	-	1.520E+02	1.483E+01	-
F8	0.000E+00	0.000E+00	4.711E+02	4.151E+02	+	2.073E+03	2.709E+02	+
F9	6.977E+03	2.830E+02	4.099E+03	1.063E+03	-	3.812E+03	5.000E+02	-
F10	5.479E+01	2.371E+01	2.118E+02	4.886E+01	+	3.476E+02	8.227E+01	+
F11	6.870E+03	4.699E+03	7.226E+07	6.183E+07	+	1.933E+07	2.805E+07	+
F12	8.038E+01	1.529E+01	3.304E+06	8.916E+06	+	2.914E+04	6.218E+03	+
F13	6.112E+01	6.580E+00	2.677E+04	2.996E+04	+	4.819E+05	1.257E+05	+
F14	3.720E+01	7.000E+00	4.274E+04	6.755E+04	+	1.048E+04	2.115E+03	+
F15	1.166E+03	2.990E+02	1.149E+03	2.762E+02	=	1.519E+03	2.480E+02	+
F16	7.435E+01	7.864E+00	3.325E+02	1.401E+02	+	1.235E+03	1.613E+02	+
F17	3.592E+01	3.761E+00	2.224E+05	2.520E+05	+	3.164E+05	1.311E+05	+
F18	1.767E+01	5.084E+00	3.538E+04	1.165E+05	+	1.168E+04	4.931E+03	+
F19	3.917E+01	2.165E+01	3.558E+02	1.002E+02	+	1.077E+03	1.967E+02	+
F20	3.723E+02	8.539E+00	3.485E+02	3.138E+01	-	4.551E+02	2.132E+01	+
F21	1.000E+02	8.922E-14	2.002E+02	2.621E+01	+	3.759E+03	2.105E+03	+
F22	5.205E+02	1.353E+01	6.072E+02	3.463E+01	+	1.331E+03	1.182E+02	+
F23	5.876E+02	1.168E+01	6.502E+02	2.621E+01	+	8.764E+02	5.792E+01	+
F24	3.867E+02	2.604E-02	4.610E+02	3.044E+01	+	4.345E+02	8.907E+00	+
F25	2.364E+03	3.735E+02	2.969E+03	1.331E+03	+	3.790E+03	1.199E+03	+
F26	4.776E+02	8.534E+00	7.282E+02	4.753E+01	+	1.849E+03	3.235E+02	+
F27	3.256E+02	4.667E+01	5.762E+02	4.260E+01	+	5.218E+02	5.509E+01	+
F28	6.211E+02	1.604E+02	1.237E+03	2.144E+02	+	1.864E+03	1.970E+02	+
F29	2.160E+03	1.158E+02	3.119E+06	5.050E+06	+	1.419E+05	7.629E+04	+
$W/T/L$	-/-/-		5/1/23			3/0/26		

4.2. Experimental data and comparison results on IEEE CEC2017

The IEEE CEC standard test suite is a widely accepted benchmark. Using a common standard test suite facilitates comparisons and communication among researchers, leading to more objective and impartial performance evaluations of algorithms. Among these, CEC2017 exhibits superior generality and enjoys greater popularity, thus, CEC2017 is selected as the benchmark test suite. We test 29 IEEE CEC2017 benchmark functions to assess the efficacy of seven algorithms. Unimodal functions (F1–F2), basic multimodal functions (F3–F9), hybrid functions (F10–F19), and blended functions (F20-F29) are used. The dimension D of functions is set to 30. The population size N is 100. Seven algorithms including DE, PSO, GSA, CGSA-M, ALGSA, GGSA, and HGSA are compared.

From Table 1, it can be seen that on the IEEE CEC2017 29 test functions, the number of functions for DE to win PSO and GSA is 23 and 26, respectively. Fig. 2 shows the convergence graphs and box-and-whisker plots of the three algorithms DE, PSO, and GSA on F8, F13, and F25 in 30 dimensions. Compared with PSO and GSA, DE not only performs better in terms of convergence speed and probability of finding a better solution, but also has more advantages in terms of stability.

Compared to GSA, the winning number of CGSA-M, GGSA, HGSA, and ALGSA is 2, 28, 28, and 29 in Table 2, respectively, suggesting that all four GSA variants are superior to GSA. Fig. 3 shows the convergence plots and box-and-whisker plots of five algorithms, GSA, CGSA-M, GGSA, HGSA, and ALGSA, on F11, F19, and F26 in 30 dimensions. Compared to the original GSA, the four variants show great advantages in terms of convergence speed and the probability of finding a better solution. Table 3 summarizes all results of $W/T/L$.

4.3. Fitting results of population interaction network

Fig. 4 illustrates a 2-dimensional visualization of the PIN of the GSA on the F4 benchmark function. This plot provides valuable insights into the information interaction and connectivity patterns among particles with iterations in the optimization process. The GSA iterates as more and more nodes and edges are generated, and their positions are constantly updated. Fig. 4(a) shows the positions of all initial particles in the first generation of GSA. Fig. 4(b) shows that at the beginning of the iteration, the GSA produces new particles, and the particles are connected to their ancestors based on their informative interactions. Fig. 4(c) shows that in the middle of the iteration, new particles continue to be generated by the GSA, and more and more points and lines are recorded as the particle-to-particle informational interactions become more frequent. Finally, Fig. 4(d) shows that in the later stages of the iteration, the particles gradually converge to a smaller range.

Since CGSA-M, ALGSA, GGSA, and HGSA are superior to GSA, it is worth analyzing their difference and characteristics. To be precise, we need to analyze how the population engages and how efficiency is related to the interaction within these algorithms. To

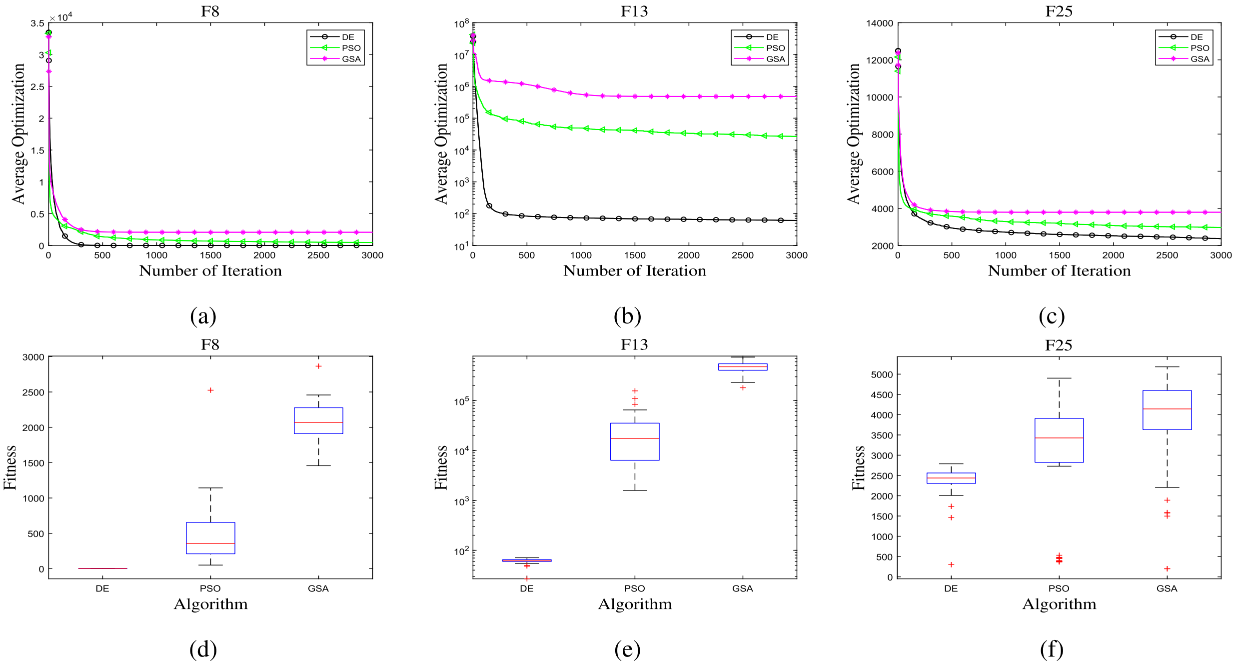


Fig. 2. Convergence graphs and box-and-whisker diagrams on IEEE CEC2017 of DE, PSO, and GSA.

Table 2
Results of GSA, CGSA-M, GGSA, HGSA, and ALGSA on IEEE CEC2017 functions according to Wilcoxon rank-sum test.

	GSA		CGSA-M			GGSA			HGSA			ALGSA		
	mean	std	mean	std		mean	std		mean	std		mean	std	
F1	1.652E+03	7.365E+02	1.799E+03	9.351E+02	=	1.934E+03	1.011E+03	=	1.978E+03	1.403E+03	=	1.426E+03	1.679E+03	+
F2	8.393E+04	6.794E+03	8.280E+04	6.211E+03	=	6.513E+04	8.112E+03	+	4.442E+04	4.558E+03	+	4.788E+04	1.327E+04	+
F3	1.367E+02	1.493E+01	1.350E+02	1.605E+01	=	1.270E+02	1.743E+01	+	1.194E+02	2.799E+00	+	1.129E+02	1.917E+01	+
F4	2.279E+02	1.692E+01	2.245E+02	1.738E+01	=	1.137E+02	1.103E+01	+	1.530E+02	1.542E+01	+	1.742E+01	3.804E+00	+
F5	5.104E+01	4.024E+00	5.005E+01	3.223E+00	=	7.893E+00	4.655E+00	+	9.119E+00	4.342E+00	+	1.461E-06	7.363E-07	+
F6	8.551E+01	1.224E+01	8.717E+01	1.107E+01	=	3.731E+01	1.881E+00	+	4.131E+01	2.764E+00	+	4.236E+01	2.750E+00	+
F7	1.520E+02	1.483E+01	1.525E+02	1.385E+01	=	8.810E+01	1.009E+01	+	1.076E+02	9.810E+00	+	1.522E+01	2.774E+00	+
F8	2.073E+03	2.709E+02	2.009E+03	2.613E+02	=	3.511E-03	1.755E-02	+	6.242E-14	5.713E-14	+	5.979E-12	8.845E-12	+
F9	3.812E+03	5.000E+02	3.847E+03	4.265E+02	=	3.441E+03	4.345E+02	+	3.169E+03	4.030E+02	+	1.368E+03	3.567E+02	+
F10	3.476E+02	8.227E+01	3.596E+02	1.106E+02	=	1.558E+02	3.686E+01	+	9.749E+01	3.129E+01	+	4.675E+01	3.052E+01	+
F11	1.933E+07	2.805E+07	1.127E+07	2.245E+07	=	5.395E+05	2.112E+05	+	1.200E+05	6.831E+04	+	2.314E+04	9.597E+03	+
F12	2.914E+04	6.218E+03	2.890E+04	5.595E+03	=	1.814E+04	4.724E+03	+	1.278E+04	5.020E+03	+	3.691E+03	1.983E+03	+
F13	4.819E+05	1.257E+05	4.750E+05	1.282E+05	=	2.339E+05	8.214E+04	+	6.467E+03	4.103E+03	+	1.667E+04	2.052E+04	+
F14	1.048E+04	2.115E+03	1.015E+04	2.031E+03	=	3.043E+03	2.044E+03	+	7.928E+02	7.178E+02	+	1.242E+03	1.475E+03	+
F15	1.519E+03	2.480E+02	1.548E+03	2.987E+02	=	1.229E+03	2.403E+02	+	1.215E+03	2.305E+02	+	6.762E+02	2.111E+02	+
F16	1.235E+03	1.613E+02	1.165E+03	2.041E+02	+	1.027E+03	2.095E+02	+	1.037E+03	2.115E+02	+	2.961E+02	1.164E+02	+
F17	3.164E+05	1.311E+05	2.864E+05	1.538E+05	=	1.667E+05	8.011E+04	+	6.079E+04	1.832E+04	+	6.941E+04	4.194E+04	+
F18	1.168E+04	4.931E+03	1.053E+04	3.279E+03	=	3.955E+03	1.396E+03	+	3.410E+03	1.381E+03	+	5.665E+03	4.865E+03	+
F19	1.077E+03	1.967E+02	9.756E+02	1.946E+02	+	8.911E+02	1.673E+02	+	8.555E+02	2.181E+02	+	1.821E+02	6.765E+01	+
F20	4.551E+02	2.132E+01	4.579E+02	2.081E+01	=	3.173E+02	2.127E+01	+	3.203E+02	3.476E+01	+	2.211E+02	5.571E+00	+
F21	3.759E+03	2.105E+03	4.172E+03	1.521E+03	=	1.000E+02	2.703E-08	+	2.648E +02	8.242E+02	+	1.000E+02	7.118E-06	+
F22	1.331E+03	1.182E+02	1.313E+03	1.302E+02	=	5.664E+02	3.664E+01	+	4.920E+02	1.348E+02	+	3.512E+02	1.632E+01	+
F23	8.764E+02	5.792E+01	8.843E+02	5.830E+01	=	5.073E+02	3.673E+01	+	5.200E+02	4.324E+01	+	3.729E+02	4.161E+01	+
F24	4.345E+02	8.907E+00	4.338E+02	8.944E+00	=	4.235E+02	1.284E+01	+	3.911E+02	8.752E+00	+	3.869E+02	6.918E-02	+
F25	3.790E+03	1.199E+03	4.080E+03	7.605E+02	=	2.975E+02	3.849E+02	+	2.510E+02	5.049E+01	+	2.644E+02	1.274E+02	+
F26	1.849E+03	3.235E+02	1.841E+03	3.309E+02	=	6.761E+02	4.348E+01	+	5.477E+02	1.545E+01	+	5.257E+02	1.397E+01	+
F27	5.218E+02	5.509E+01	5.236E+02	5.699E+01	=	4.355E+02	2.910E+01	+	3.052E+02	2.117E+01	+	3.684E+02	5.351E+01	+
F28	1.864E+03	1.970E+02	1.831E+03	2.235E+02	=	1.412E+03	2.025E+02	+	1.256E+03	2.272E+02	+	4.945E+02	9.260E+01	+
F29	1.419E+05	7.629E+04	1.449E+05	9.645E+04	=	3.955E+04	1.637E+04	+	7.545E+03	1.306E+03	+	4.221E+03	7.421E+02	+
$W/T/L$	-/-/-		2/27/0			28/1/0			28/1/0			29/0/0		

achieve it, we build PINs of the five algorithms and run 30 times on each function to record their nodes and edges. A degree of node which is identical to the number of connections indicates how many of its neighboring nodes are involved in interactions with it. The larger the value of degree, the higher the number of information interactions between nodes, and the smaller the value of degree, the lower the number of information interactions between nodes. The number of information interactions between nodes is reflected by the value of degree. We utilize seven distribution models to precisely model the frequency distribution of node degrees, thereby assessing the structural traits of PINs in diverse GSA variants. These models are widely employed in real-world scenarios to reflect certain phenomena or regulations. By fitting these distribution models to the degree frequency distribution, we can obtain valuable insights into the underlying network topology and connectivity patterns, which can enhance our comprehension of the optimization behavior of GSA algorithms.

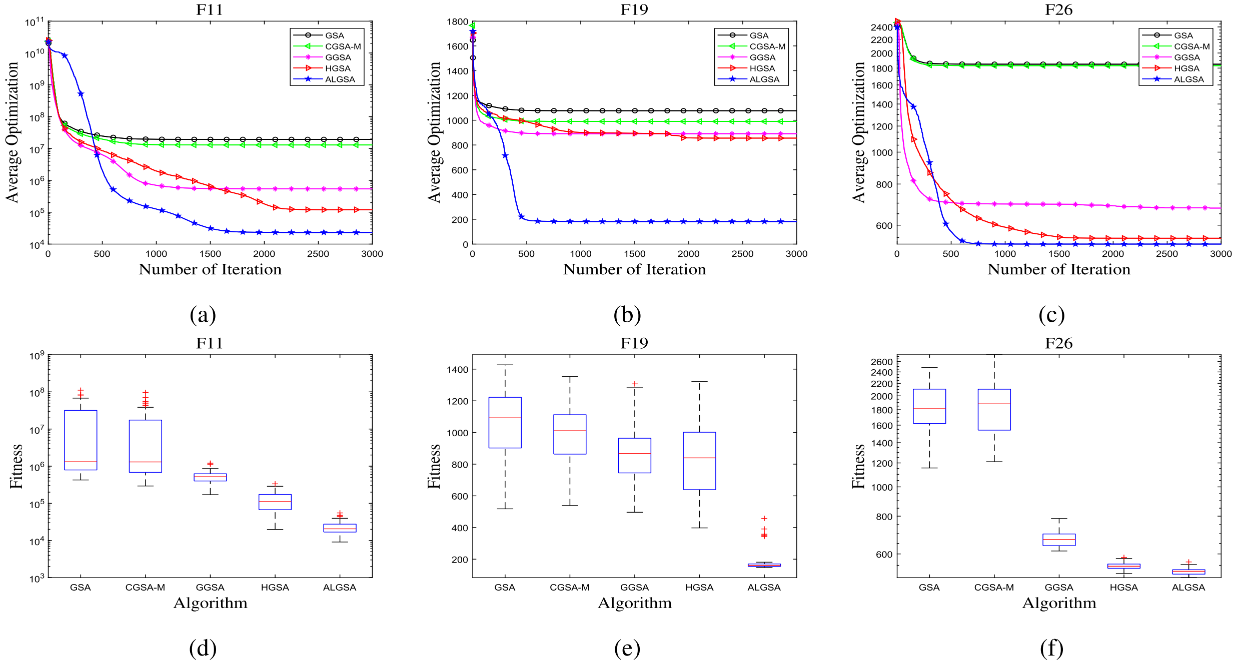


Fig. 3. Convergence graphs and box-and-whisker diagrams on IEEE CEC2017 of GSA, CGSA-M, GGSA, HGSA, and ALGSA.

Table 3
Results of DE, PSO, GSA, CGSA-M, GGSA, HGSA, and ALGSA on IEEE CEC2017 functions according to Wilcoxon rank-sum test.

DE *vs*	PSO	GSA	CGSA-M	GGSA	HGSA	ALGSA
$W/T/L$	23/1/5	26/0/3	26/0/3	21/1/7	19/2/8	18/0/11

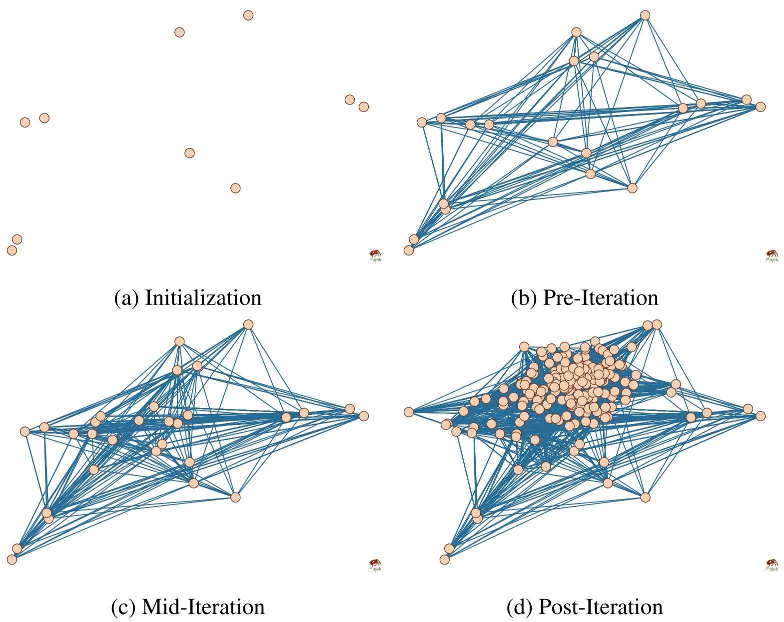
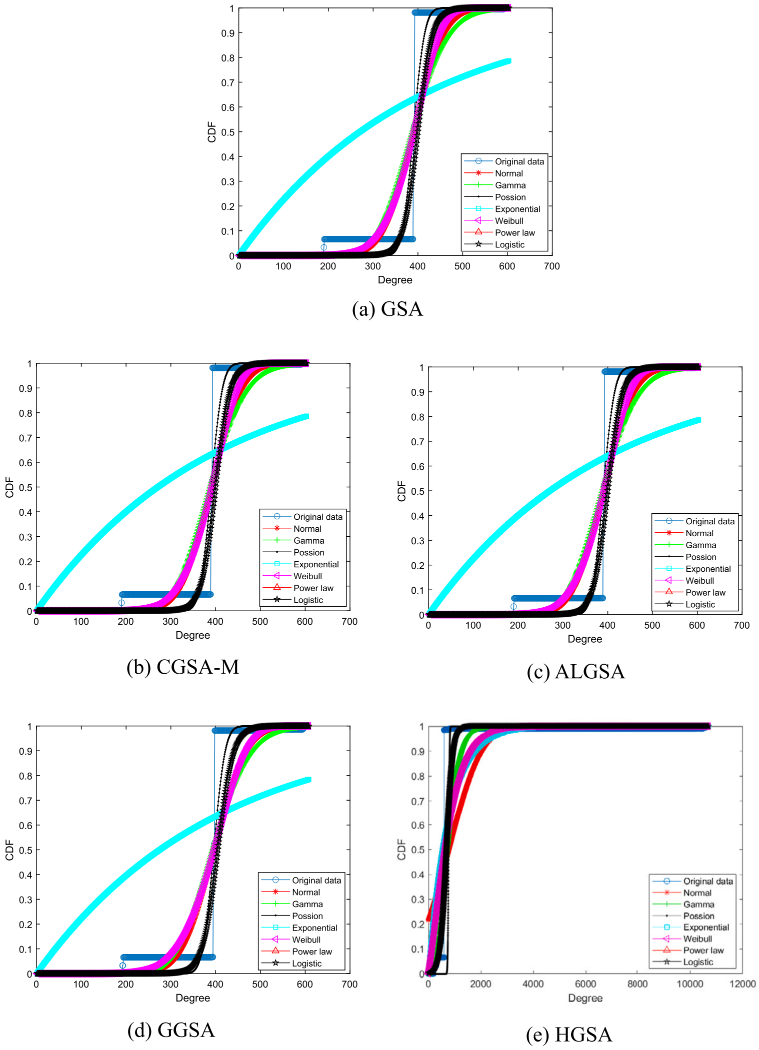


Fig. 4. Generation process of population interaction network on IEEE CEC2017 F4 function. Because of the gravitational effect of $Kbest$ individuals in the iterative process, more and more nodes and edges are generated. (a) – (d) show the positions of the particles under the four iteration periods.



(a) GSA

(b) CGSA-M

(c) ALGSA

(d) GGSA

(e) HGSA

Fig. 5. Plots of the seven model distribution functions fitted to the cumulative distribution functions of GSA, CGSA-M, GGSA, HGSA, and ALGSA nodal degrees. The raw data fitted poorly to the Exponential model and is better identified with the Logistic model fit.

Fig. 5 displays the fitting results of various GSA variants on the IEEE CEC2017 benchmark function. The horizontal axis represents the degree of the node. The vertical axis represents the cumulative distribution function (CDF) of the node degree. HGSA is obviously different from the other GSA variants because it has more degrees of nodes.

The results presented in Fig. 5 demonstrate that the Exponential distribution model is not a suitable fit for the CDF of node degrees in the five GSA algorithms under investigation. While the fitting results of the other distribution models are similar, we aim to more precisely distinguish between them. To achieve this goal, we employ two measures, namely the sum of squared errors (SSE) and the coefficient of determination (R^2), to assess the appropriateness of the different models for the observed data. Eq. (15) for SSE is calculated as follows:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{15}$$

where y_i and $\hat{y}_i$ represent the original data and fitting data, respectively, and n is the maximum degree of nodes. A smaller SSE value indicates a better fitting result.

The coefficient of determination R^2 shows how well the fitting data match the original data. The greatest R^2 which is close to 1, the better fitting result. Eq. (16) provides R^2 where $\bar{y}$ stands for the mean of the initial data.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{16}$$

Table 4
Fitting the results of GSA, GGSA, HGSA, CGSA-M, and ALGSA with seven models.

	Normal		Gamma		Poisson		Exponential		Weibull		Power law		Logistic	
	SSE	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2
GSA	9.468	0.912	11.425	0.888	5.301	0.958	70.801	-1.373	41.232	0.267	365.724	-82.258	**4.503**	0.962
CGSA-M	9.468	0.912	11.426	0.888	5.301	0.958	70.802	-1.373	41.233	0.267	365.724	-82.258	**4.503**	0.962
ALGSA	9.468	0.912	11.426	0.888	5.301	0.958	70.803	-1.373	41.233	0.267	365.726	-82.258	**4.503**	0.962
GGSA	9.934	0.906	11.758	0.884	5.079	0.960	71.067	-1.374	41.336	0.264	370.481	-83.330	**4.812**	0.959
HGSA	190.338	0.487	67.650	0.868	57.490	0.918	137.159	0.584	199.915	0.700	643.225	-143.260	**34.299**	0.935

Table 5
Fitting results of DE, PSO, and GSA, where DE has the smallest *SSE* value on Poisson, GSA and PSO have the smallest *SSE* value on Logistic.

	Normal		Gamma		Poisson		Exponential		Weibull		Power law		Logistic	
	SSE	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2	*SSE*	R^2
DE	0.163	0.960	0.089	0.979	**0.059**	0.986	0.500	0.760	0.123	0.969	0.093	0.819	0.078	0.981
PSO	7.161	0.157	1.073	0.831	0.855	0.880	1.588	0.642	23.358	0.262	0.811	0.818	**0.646**	0.913
GSA	9.468	0.912	11.425	0.888	5.301	0.958	70.801	-1.373	41.232	0.267	365.724	-82.258	**4.503**	0.962

By using both statistical methods, *SSE* and R^2, we can more accurately assess the relationship between the GSA variants and each distribution function model. This enables us to select the most appropriate model. In order to study GSA in more depth, we have not only studied the original GSA and its two types of variants but also selected two other different types of original algorithms, DE and PSO, for comparison.

• **GSA and its variants:** In Table 4, we enumerate the average fitting result of each model for five algorithms. From this table, GSA variants which change the initial structure, such as GGSA and HGSA, are found to increasingly deviate from the Logistic model according to *SSE* and R^2. However, CGSA-M and ALGSA did not change the structure of the original GSA and therefore have the same logistic distribution as GSA. Accordingly, we can conclude that the CDF of node degree in GSA and its variants conforms to the Logistic model, but the GSA variant with improved structure deviates from the Logistic model.

• **DE, PSO, and GSA:** In Table 5, we enumerate the average fitting result of each model for three types of algorithms. From this table, it is evident that the Poisson model provides the best approximation for the cumulative distribution function of degree of nodes in DE, while the Logistic model better fits the distribution of degree of nodes in PSO and GSA.

The Logistic model can be expressed as: C is the correlation coefficient between y and $\hat{y}$. C value indicates the degree of correlation between y and $\hat{y}$, as shown in Eq. (17).

$$C = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y}) \left(y_i - \bar{y} \right)}{\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2 \cdot \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}} \tag{17}$$

$$\hat{y} = mt + b \tag{18}$$

where y is the original data. $\hat{y}$ is the standard Logistic function data. C is in the interval $(0, 1)$. $\hat{y}$ is calculated as shown in Eq. (18). t is the fitting data. m is the slope. b is the deviation.

Fig. 6 compares the variability between the fitting data and the standard Logistic distribution of degree of nodes in the five GSA algorithms. In Fig. 6, the standard Logistic distribution forms a dashed line. A blue line with a slope of m is fitted to the original data. C value is used to calculate the correlation between the fitting data and the standard Logistic. C value close to 1 indicates a strong correlation between the two variables. Otherwise, it indicates a weak correlation. C value in Fig. 6 shows that GSA, CGSA-M, and ALGSA are approximately equal. However, C value of GGSA is smaller than C value of GSA, and C value of HGSA is the smallest. From C value, it can be inferred that GGSA deviates more from the Logistic distribution in comparison to GSA. HGSA deviates the most from the Logistic distribution.

To evaluate the performance of GSA and its variants in more depth, Friedman test and Wilcoxon rank-sum test are used. Results are shown in Table 6. According to Table 6, it can be concluded that HGSA is better than GGSA and GGSA is better than GSA. From Tables 4 and 6, it can be concluded that GSA with improved structures becomes stronger when the frequency of relationship among individuals deviates more from the Logistic distribution. As can be seen from Tables 4 and 5, the population interaction networks formed by DE during the iteration process conform to a Poisson distribution, and the population interaction networks formed by PSO, GSA, CGSA-M, GGSA, HGSA, and ALGSA during the iteration process conform to a Logistic distribution. From the experimental results in Table 3, it can be seen that the DE conforming to Poisson distribution is superior to the other six algorithms conforming to Logistic distribution. Therefore, it can be concluded that the algorithm conforming to Poisson distribution is superior to the algorithm conforming to Logistic distribution.

4.4. Experimental data and comparison results on IEEE CEC2011

In this section, we present the performance evaluation of seven algorithms on the IEEE CEC2011 benchmark functions, which in turn validates the correctness of our proposed conclusions. The 22 real-world problems on the IEEE CEC2011 benchmarking

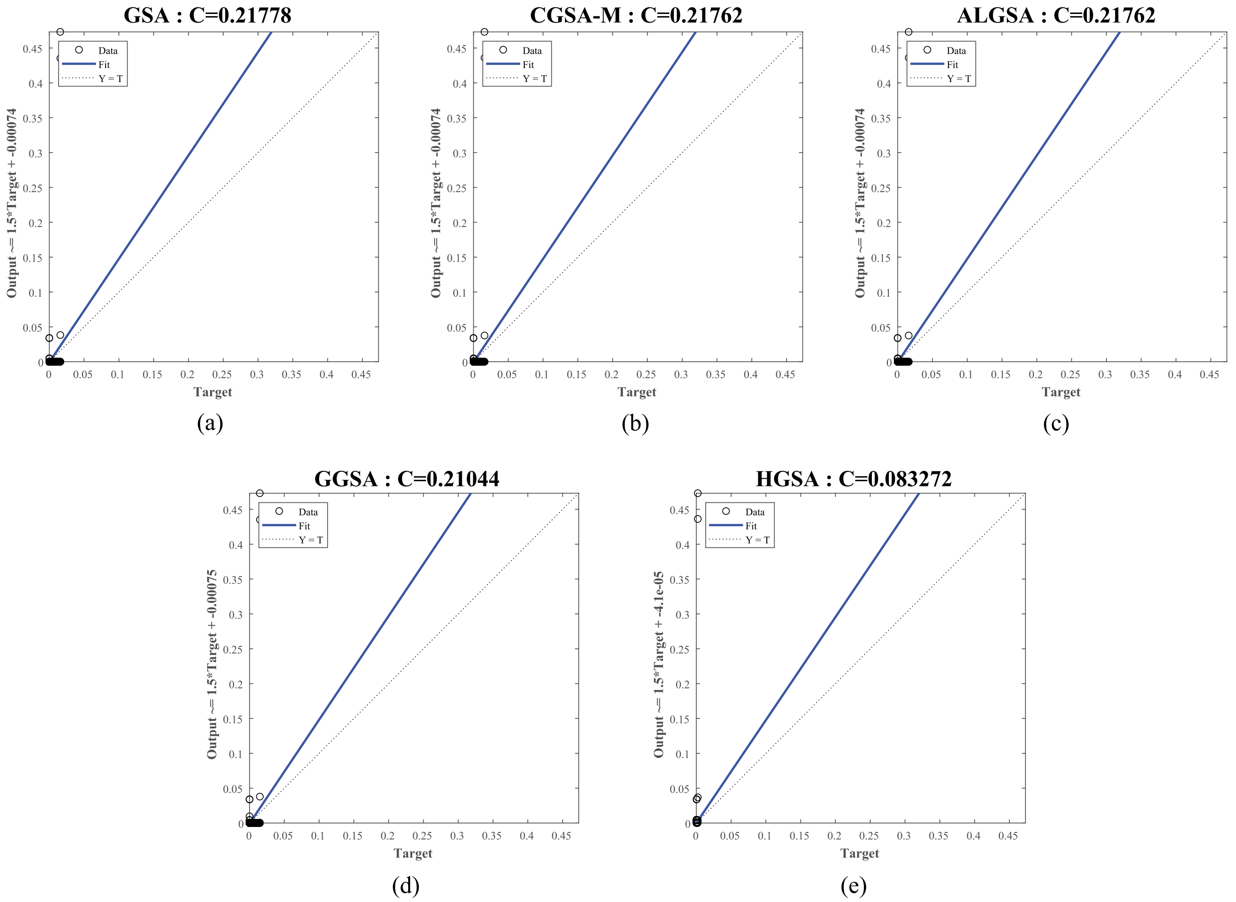


Fig. 6. Comparison of the correlations between the fitted data and the standard Logistic distributions for GSA, CGSA-M, ALGSA, GGSA, and HGSA node degrees. A larger C value proves a stronger correlation, and a smaller C value proves a weaker correlation.

Table 6
Friedman test and Wilcoxon rank-sum test of five GSA algorithms on 29 IEEE CEC2017 functions.

Algorithm	*SSE*	R^2	*Rank*	Comparison	$W/T/L$
ALGSA	4.503	0.962	1	ALGSA *vs* HGSA	17/1/11
HGSA	34.299	0.935	2	HGSA *vs* GGSA	17/7/5
GGSA	4.812	0.959	3	GGSA *vs* CGSA-M	26/3/0
CGSA-M	4.503	0.962	4	CGSA-M *vs* GSA	2/27/0
GSA	4.503	0.962	5	−−	−/−/−

set include problems in hydrothermal scheduling, static economic scheduling, dynamic economic scheduling, antenna array design, transmission pricing problems, and radar polyphase code design. The purpose of these issues, which include a broad variety of features and levels of difficulty, is to objectively assess how well optimization algorithms perform when faced with real-world situations. From Table 7, it can be seen that on the IEEE CEC2011 22 test functions, the number of functions for DE to win PSO and GSA is 20 and 13, respectively. According to Table 8, in comparison to GSA, the winning number of CGSA-M, GGSA, HGSA, and ALGSA is 2, 28, 28, and 29, respectively, suggesting that all four GSA variants are superior to GSA.

To evaluate the performance of GSA, GGSA, and HGSA in more depth, Friedman test and Wilcoxon rank-sum test are used. Results are shown in Table 9. According to Table 9, it can be concluded that HGSA is better than GGSA and GGSA is better than GSA. Therefore, the validity of the conclusion that the structurally improved GSA becomes stronger when the frequency of relationships between individuals deviates more from the Logistic distribution is verified.

From the experimental results in Table 10, it can be seen that the DE conforming to Poisson distribution is superior to the other six algorithms conforming to Logistic distribution. Therefore, the validity of the conclusion that the algorithm conforming to the Poisson distribution is superior to the algorithm conforming to the Logistic distribution is verified.

Table 7
Results of DE, PSO, and GSA on IEEE CEC2011 functions according to Wilcoxon rank-sum test.

	DE		PSO		GSA	
	mean	std	mean	std	mean	std
G1	1.410E+00	3.747E+00	2.374E+01	1.746E+00	2.583E+01	1.556E+00
G2	-9.802E+00	7.620E-01	-4.386E+00	4.846E-01	-1.855E+01	3.930E+00
G3	1.151E-05	2.336E-19	1.151E-05	1.847E-12	1.151E-05	4.123E-19
G4	1.786E+01	3.288E+00	1.556E+01	1.510E+00	1.953E+01	2.156E+00
G5	-2.143E+01	1.168E+00	-1.953E+01	1.336E+00	-3.028E+01	3.806E+00
G6	-1.607E+01	9.587E-01	-1.352E+01	1.293E+00	-1.845E+01	2.951E+00
G7	1.719E+00	1.181E-01	1.723E+00	1.181E-01	9.193E-01	2.275E-01
G8	2.200E+02	0.000E+00	2.335E+02	1.076E+01	2.830E+02	7.052E+01
G9	3.960E+03	7.271E+02	2.449E+06	6.657E+04	1.362E+03	1.825E+02
G10	-2.172E+01	9.663E-02	-8.441E+00	9.112E-01	-1.126E+01	5.182E-01
G11	6.655E+04	3.336E+03	1.876E+08	1.000E+07	5.218E+04	3.627E+02
G12	1.742E+07	5.123E+04	5.201E+07	6.485E+05	3.224E+07	1.040E+06
G13	1.544E+04	6.094E-06	1.584E+04	3.431E+02	1.203E+05	5.726E+04
G14	1.840E+04	1.535E+02	1.960E+04	3.202E+02	1.920E+04	1.150E+02
G15	3.279E+04	1.835E+01	1.539E+05	4.449E+04	3.324E+05	3.840E+04
G16	1.359E+05	1.909E+03	2.258E+06	1.629E+06	1.464E+05	2.182E+03
G17	2.090E+06	1.081E+05	5.175E+09	7.772E+08	2.720E+06	6.020E+05
G18	1.302E+06	1.139E+05	8.007E+07	6.636E+06	9.420E+05	1.257E+03
G19	1.918E+06	1.600E+05	7.947E+07	6.531E+06	9.907E+05	4.785E+04
G20	1.303E+06	1.284E+05	8.063E+07	5.422E+06	9.421E+05	1.353E+03
G21	1.617E+01	2.923E+00	3.946E+01	3.315E+00	4.164E+01	8.120E+00
G22	1.799E+01	3.255E+00	3.896E+01	2.838E+00	5.341E+01	6.618E+00
$W/T/L$	-/-/-		0/2/20		9/0/13	

Table 8
Results of GSA, CGSA-M, GGSA, HGSA, and ALGSA on IEEE CEC2011 functions according to Wilcoxon rank-sum test.

	GSA		CGSA-M		GGSA		HGSA		ALGSA	
	mean	std	mean	std	mean	std	mean	std	mean	std
G1	2.583E+01	1.556E+00	2.525E+01	1.323E+00	2.589E+01	8.920E-01	1.417E+01	6.503E+00	2.507E+01	1.264E+00
G2	-1.855E+01	3.930E+00	-1.838E+01	3.772E+00	-2.218E+01	3.805E+00	-2.456E+01	2.283E+00	-2.563E+01	2.136E+00
G3	1.151E-05	4.123E-19	1.151E-05	4.293E-19	1.151E-05	4.434E-19	1.151E-05	1.927E-12	1.151E-05	4.207E-15
G4	1.953E+01	2.156E+00	1.804E+01	2.811E+00	1.889E+01	2.489E+00	1.552E+01	1.377E+00	1.801E+01	2.632E+00
G5	-3.028E+01	3.806E+00	-2.995E+01	3.382E+00	-3.086E+01	3.436E+00	-3.301E+01	2.108E+00	-3.316E+01	2.031E+00
G6	-1.845E+01	2.951E+00	-1.846E+01	2.765E+00	-1.911E+01	2.647E+00	-2.190E+01	2.297E+00	-2.122E+01	2.262E+00
G7	9.193E-01	2.275E-01	9.808E-01	1.814E-01	8.542E-01	1.600E-01	7.128E-01	1.318E-01	6.921E-01	1.172E-01
G8	2.830E+02	7.052E+01	2.772E+02	4.405E+01	2.483E+02	2.839E+01	2.204E+02	2.135E+00	2.307E+02	1.285E+01
G9	1.362E+03	1.825E+02	1.331E+03	2.190E+02	1.984E+03	2.015E+03	2.101E+05	3.875E+04	7.086E+03	8.039E+03
G10	-1.126E+01	5.182E-01	-1.129E+01	4.344E-01	-1.189E+01	4.820E-01	-1.286E+01	6.137E-01	-1.472E+01	9.478E-01
G11	5.218E+04	3.627E+02	5.216E+04	4.054E+02	5.231E+04	4.108E+02	5.120E+04	4.833E+02	5.130E+04	5.963E+02
G12	3.224E+07	1.040E+06	3.231E+07	1.064E+06	2.507E+07	6.543E+05	2.050E+07	1.768E+05	1.837E+07	7.973E+04
G13	1.203E+05	5.726E+04	9.092E+04	4.944E+04	8.325E+04	2.787E+04	4.673E+04	3.697E+04	1.579E+04	1.339E+03
G14	1.920E+04	1.150E+02	1.917E+04	1.049E+02	1.923E+04	1.133E+02	1.914E+04	1.348E+02	1.921E+04	1.661E+02
G15	3.324E+05	3.840E+04	3.265E+05	4.379E+04	2.832E+05	3.333E+04	3.325E+04	2.413E+01	3.325E+04	2.114E+01
G16	1.464E+05	2.182E+03	1.458E+05	1.936E+03	1.454E+05	1.758E+03	1.430E+05	2.014E+03	1.409E+05	1.630E+03
G17	2.720E+06	6.020E+05	2.743E+06	5.348E+05	2.311E+06	3.949E+05	1.941E+06	6.685E+03	2.018E+06	1.116E+05
G18	9.420E+05	1.257E+03	9.421E+05	1.628E+03	9.421E+05	1.288E+03	9.430E+05	1.688E+03	9.433E+05	1.840E+03
G19	9.907E+05	4.785E+04	9.899E+05	5.383E+04	9.731E+05	4.410E+04	1.189E+06	1.018E+05	9.438E+05	4.544E+03
G20	9.421E+05	1.353E+03	9.418E+05	1.250E+03	9.420E+05	1.480E+03	9.433E+05	1.906E+03	9.440E+05	2.482E+03
G21	4.164E+01	8.120E+00	4.029E+01	6.913E+00	3.476E+01	7.752E+00	2.650E+01	5.342E+00	1.739E+01	2.311E+00
G22	5.341E+01	6.618E+00	5.251E+01	7.455E+00	5.099E+01	9.774E+00	3.740E+01	5.824E+00	2.918E+01	2.885E+00
$W/T/L$	-/-/-		4/15/3		10/11/1		16/1/5		17/2/3	

Table 9
Friedman test and Wilcoxon rank-sum test of five GSA algorithms on 22 IEEE CEC2011 functions.

Algorithm	*Rank*	Comparison	$W/T/L$
HGSA	1	HGSA *vs* GGSA	17/0/5
GGSA	2	GGSA *vs* GSA	10/11/1
GSA	3	−−	−/−/−

4.5. Some general remarks

In the past two to three decades of research, researchers have categorized MHAs into three types: evolution-inspired, swarm-inspired, and physical law-inspired. Although this classification system can indicate the origins of the algorithms, it does not clearly

Table 10
Results of DE, PSO, GSA, CGSA-M, GGSA, HGSA, and ALGSA on IEEE CEC2011 functions according to Wilcoxon rank-sum test.

DE *vs*	PSO	GSA	CGSA-M	GGSA	HGSA	ALGSA
$W/T/L$	20/2/0	13/0/9	12/1/9	12/1/9	11/1/10	11/2/9

elucidate their essential differences. As a result, researchers find it challenging to obtain sufficient information from these classifications and struggle to choose appropriate MHAs for different optimization problems. To address this issue and accommodate various types of optimization problems, researchers have proposed a multitude of diverse algorithms, a practice often called the "alchemical dilemma". In contrast, our classification method focuses on the complex network structures formed by algorithms during the iteration process and establishes a clear classification approach. In this study, the evolution-inspired DE, swarm intelligence-inspired PSO, and physics-inspired GSA are classified into two categories using the PIN theory: algorithms based on Poisson distribution and algorithms based on Logistic distribution. The former includes DE, while PSO and GSA is the algorithm based on Logistic distribution. This method facilitates a more effective selection of algorithms for different optimization problems, making a novel and valuable contribution to the field of meta-heuristic research.

5. Conclusions

In this paper, we have constructed PINs of seven meta-heuristic algorithms and have evaluated their structural characteristics by fitting their cumulative distribution functions of node degrees using seven different distribution models. The analysis is performed on 29 IEEE CEC2017 benchmark functions. The conclusions are given from the trial findings: 1) The relationship among individuals in DE conforms to the Poisson distribution. However, the relationship among individuals in PSO, GSA, and GSA variants meets the Logistic distribution. 2) In terms of algorithm performance, the algorithm whose inter-individual relationships conform to a Poisson distribution outperforms the algorithm whose inter-individual relationships conform to a Logistic distribution. 3) In the structurally improved variant of GSA, the more the frequency of individual-to-individual relationships deviates from the Logistic distribution, the stronger the performance. These findings offer valuable insights into the optimization performance of MHAs from the viewpoint of complex networks, which can inform future efforts to improve algorithm design. Therefore, our study provides crucial guidance towards developing more effective optimization algorithms in the field of computer science.

However, there are still drawbacks in this paper: 1) There are many algorithms whose performance changes because of parameter changes. The current PIN method is still imperfect, as it can only analyze algorithms whose structure has changed and cannot reflect the effect of parameter changes on the performance of algorithms. In future work, more evaluation indicators should be entered to further improve the PIN method. 2) There are still many MHAs that have not been investigated by the PIN method. It is possible to discover additional types of distribution to match the population structure of MHAs. 3) There is also a need to further extend the application of PIN to real optimization problems. Currently, we use PIN to analyze the algorithms, but the same algorithm performs differently on different optimization problems, so in future work, we will try to use PIN to analyze real optimization problems and filter more suitable algorithms for optimization problems.

CRediT authorship contribution statement

Haotian Li: Writing – review & editing, Writing – original draft, Software, Methodology. **Yifei Yang:** Software, Methodology, Conceptualization. **Yirui Wang:** Writing – review & editing, Methodology. **Jiayi Li:** Software, Investigation. **Haichuan Yang:** Writing – review & editing, Visualization. **Jun Tang:** Writing – review & editing, Software, Project administration, Conceptualization. **Shangce Gao:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization.

Declaration of competing interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Data availability

The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

Acknowledgement

This research was partially supported by the National Natural Science Foundation of China (Grant No. 62203238), Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP22H03643, Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145, and JST through the Establishment of University Fellowships towards the Creation of Science Technology Innovation under Grant JPMJFS2115.

References

[1] M. Chumburidze, I. Basheleishvili, A. Khetsuriani, Dynamic programming and greedy algorithm strategy for solving several classes of graph optimization problems, BRAIN. Broad Res. Artif. Intell. Neurosci. 10 (1) (2019) 101–107.
[2] I. Grossmann, R. Apap, B. Calfa, P. Garcia-Herreros, Q. Zhang, Mathematical programming techniques for optimization under uncertainty and their application in process systems engineering, Theor. Found. Chem. Eng. 51 (6) (2017) 893–909.
[3] Y. Wang, S. Gao, M. Zhou, Y. Yu, A multi-layered gravitational search algorithm for function optimization and real-world problems, IEEE/CAA J. Autom. Sin. 8 (1) (2021) 94–109.
[4] B.A. Hassan, T.A. Rashid, H.K. Hamarashid, A novel cluster detection of covid-19 patients and medical disease conditions using improved evolutionary clustering algorithm star, Comput. Biol. Med. 138 (2021) 104866.
[5] S.M. Qader, B.A. Hassan, T.A. Rashid, An improved deep convolutional neural network by using hybrid optimization algorithms to detect and classify brain tumor using augmented mri images, Multimed. Tools Appl. 81 (30) (2022) 44059–44086.
[6] M.T. Abdulkhaleq, T.A. Rashid, A. Alsadoon, B.A. Hassan, M. Mohammadi, J.M. Abdullah, A. Chhabra, S.L. Ali, R.N. Othman, H.A. Hasan, et al., Harmony search: current studies and uses on healthcare systems, Artif. Intell. Med. 131 (2022) 102348.
[7] A. Guo, Y. Wang, L. Guo, R. Zhang, Y. Yu, S. Gao, An adaptive position-guided gravitational search algorithm for function optimization and image threshold segmentation, Eng. Appl. Artif. Intell. 121 (2023) 106040.
[8] L. Abualigah, M.A. Elaziz, A.M. Khasawneh, M. Alshinwan, R.A. Ibrahim, M.A. Al-qaness, S. Mirjalili, P. Sumari, A.H. Gandomi, Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results, Neural Comput. Appl. 34 (2022) 4081–4110.
[9] B.B. Maaroof, T.A. Rashid, J.M. Abdulla, B.A. Hassan, A. Alsadoon, M. Mohammadi, M. Khishe, S. Mirjalili, Current studies and applications of shuffled frog leaping algorithm: a review, Arch. Comput. Methods Eng. 29 (5) (2022) 3459–3474.
[10] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. 52 (4) (2019) 2191–2233.
[11] P. Winker, M. Gilli, Applications of optimization heuristics to estimation and modelling problems, Comput. Stat. Data Anal. 47 (2) (2004) 211–223.
[12] B.A. Hassan, T.A. Rashid, S. Mirjalili, Performance evaluation results of evolutionary clustering algorithm star for clustering heterogeneous datasets, Data Brief 36 (2021) 107044.
[13] J. Jiang, X. Yang, X. Meng, K. Li, Enhance chaotic gravitational search algorithm (CGSA) by balance adjustment mechanism and sine randomness function for continuous optimization problems, Phys. A, Stat. Mech. Appl. 537 (2020) 122621.
[14] K. Sörensen, Metaheuristics—the metaphor exposed, Int. Trans. Oper. Res. 22 (1) (2015) 3–18.
[15] C.L. Camacho Villalón, T. Stützle, M. Dorigo, Grey wolf, firefly and bat algorithms: three widespread algorithms that do not contain any novelty, in: International Conference on Swarm Intelligence, Springer, 2020, pp. 121–133.
[16] B.A. Hassan, CSCF: a chaotic sine cosine firefly algorithm for practical application problems, Neural Comput. Appl. 33 (12) (2021) 7011–7030.
[17] B.A. Hassan, T.A. Rashid, A multidisciplinary ensemble algorithm for clustering heterogeneous datasets, Neural Comput. Appl. 33 (17) (2021) 10987–11010.
[18] M. Hutson, Has artificial intelligence become alchemy?, Science 360 (6388) (2018) 478.
[19] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, J. Cheng, A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models, Energy Convers. Manag. 230 (2021) 113784.
[20] Z. Lei, S. Gao, Z. Zhang, M. Zhou, J. Cheng, Mo4: a many-objective evolutionary algorithm for protein structure prediction, IEEE Trans. Evol. Comput. 26 (3) (2022) 417–430.
[21] H. Yang, Y. Yu, J. Cheng, Z. Lei, Z. Cai, Z. Zhang, S. Gao, An intelligent metaphor-free spatial information sampling algorithm for balancing exploitation and exploration, Knowl.-Based Syst. 250 (2022) 109081.
[22] Z. Lei, S. Gao, Y. Wang, Y. Yu, L. Guo, An adaptive replacement strategy-incorporated particle swarm optimizer for wind farm layout optimization, Energy Convers. Manag. 269 (2022) 116174.
[23] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, J. Wang, Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction, IEEE Trans. Neural Netw. Learn. Syst. 30 (2) (2019) 601–614.
[24] P. Chunaev, Community detection in node-attributed social networks: a survey, Comput. Sci. Rev. 37 (2020) 100286.
[25] S.H. Strogatz, Exploring complex networks, Nature 410 (6825) (2001) 268–276.
[26] A.-L. Barabási, Scale-free networks: a decade and beyond, Science 325 (5939) (2009) 412–413.
[27] M. Li, R.-R. Liu, L. Lü, M.-B. Hu, S. Xu, Y.-C. Zhang, Percolation on complex networks: theory and application, Phys. Rep. 907 (2021) 1–68.
[28] P. Holme, Rare and everywhere: perspectives on scale-free networks, Nat. Commun. 10 (1) (2019) 1–3.
[29] N. Koroniotis, N. Moustafa, E. Sitnikova, A new network forensic framework based on deep learning for Internet of Things networks: a particle deep framework, Future Gener. Comput. Syst. 110 (2020) 91–106.
[30] J.J. Cheng, M.D. Ju, M.C. Zhou, C. Liu, S.C. Gao, A. Abusorrah, C.J. Jiang, A dynamic evolution method for autonomous vehicle groups in a highway scene, IEEE Int. Things J. 9 (2021) 1445–1457.
[31] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[32] B.A. Hassan, T.A. Rashid, S. Mirjalili, Formal context reduction in deriving concept hierarchies from corpora using adaptive evolutionary clustering algorithm star, Complex Intell. Syst. 7 (5) (2021) 2383–2398.
[33] N. Lynn, M.Z. Ali, P.N. Suganthan, Population topologies for particle swarm optimization and differential evolution, Swarm Evol. Comput. 39 (2018) 24–35.
[34] Y. Shi, H. Liu, L. Gao, G. Zhang, Cellular particle swarm optimization, Inf. Sci. 181 (20) (2011) 4460–4493.
[35] Y. Wang, S. Gao, Y. Yu, Z. Cai, Z. Wang, A gravitational search algorithm with hierarchy and distributed framework, Knowl.-Based Syst. 218 (2021) 106877.
[36] J. Qi, Z. Rong, The emergence of scaling laws search dynamics in a particle swarm optimization, Phys. A, Stat. Mech. Appl. 392 (6) (2013) 1522–1531.
[37] W. Du, M. Zhang, W. Ying, M. Perc, K. Tang, X. Cao, D. Wu, The networked evolutionary algorithm: a network science perspective, Appl. Math. Comput. 338 (2018) 33–43.
[38] V. Homolya, T. Vinkó, Leveraging local optima network properties for memetic differential evolution, in: World Congress on Global Optimization, Springer, 2019, pp. 109–118.
[39] V. Homolya, T. Vinkó, Memetic Differential Evolution Using Network Centrality Measures, AIP Conference Proceedings, vol. 2070, AIP Publishing LLC, 2019, p. 020023.
[40] B. Ji, X. Yuan, Z. Chen, H. Tian, Improved gravitational search algorithm for unit commitment considering uncertainty of wind power, Energy 67 (2014) 52–62.
[41] Z. Chen, X. Yuan, H. Tian, B. Ji, Improved gravitational search algorithm for parameter identification of water turbine regulation system, Energy Convers. Manag. 78 (2014) 306–315.
[42] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[43] S. Mirjalili, A. Lewis, Adaptive gbest-guided gravitational search algorithm, Neural Comput. Appl. 25 (2014) 1569–1584.
[44] Z. Song, S. Gao, Y. Yu, J. Sun, Y. Todo, Multiple chaos embedded gravitational search algorithm, IEICE Trans. Inf. Syst. 100 (4) (2017) 888–900.
[45] Y. Wang, Y. Yu, S. Gao, H. Pan, G. Yang, A hierarchical gravitational search algorithm with an effective gravitational constant, Swarm Evol. Comput. 46 (2019) 118–139.
[46] Z. Lei, S. Gao, S. Gupta, J. Cheng, G. Yang, An aggregative learning gravitational search algorithm with self-adaptive gravitational constants, Expert Syst. Appl. 152 (2020) 113396.

[47] B. Dorronsoro, P. Bouvry, Improving classical and decentralized differential evolution with new mutation operator and population topologies, IEEE Trans. Evol. Comput. 15 (1) (2011) 67–98.
[48] J. Kennedy, R. Eberhart, Particle Swarm Optimization, Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, IEEE, 1995, pp. 1942–1948.
[49] S. Gao, Y. Wang, J. Wang, J. Cheng, Understanding differential evolution: a Poisson law derived from population interaction network, J. Comput. Sci. 21 (2017) 140–149.
[50] Y. Wang, S. Gao, Y. Yu, Z. Xu, The discovery of population interaction with a power law distribution in brain storm optimization, Memetic Comput. 11 (1) (2019) 65–87.
[51] X. Li, J. Li, H. Yang, Y. Wang, S. Gao, Population interaction network in representative differential evolution algorithms: power-law outperforms Poisson distribution, Phys. A, Stat. Mech. Appl. 603 (2022) 127764.
[52] Y. Zhang, Y. Yang, X. Li, Z. Yuan, Y. Todo, H. Yang, A dendritic neuron model optimized by meta-heuristics with a power-law-distributed population interaction network for financial time-series forecasting, Mathematics 11 (5) (2023) 1251.
[53] H. Yang, S. Gao, Z. Lei, J. Li, Y. Yu, Y. Wang, An improved spherical evolution with enhanced exploration capabilities to address wind farm layout optimization problem, Eng. Appl. Artif. Intell. 123 (2023) 106198.