BFGO: Bamboo Forest Growth Optimization Algorithm

Shu-Chuan Chu[1], Qing Feng[1], Jia Zhao[2], Jeng-Shyang Pan[1,3]*

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] School of Information Engineering, Nanchang Institute of Technology, China
[3] Department of Information Management, Chaoyang University of Technology, Taiwan
scchu0803@gmail.com, 202082060009@sdust.edu.cn, zhaojia@nit.edu.cn, jengshyangpan@gmail.com

Abstract

The heuristic optimization algorithm is a popular optimization method for solving optimization problems. However, the main disadvantage of this type of algorithm is the unstable performance, and the performance depends on the specific problem and the designer's experience, so inappropriate algorithms or parameters will lead to poor performance for solving optimization. In this paper, a new meta-heuristic algorithm—bamboo forest growth optimization (BFGO) algorithm is proposed. The BFGO algorithm refers to the growth law of bamboo: first, take root clearly, and then grow wildly. The growth characteristics of bamboo forests and the optimization process of the algorithm are integrated, and the test is carried out on 30 functions of the CEC2017 test set, and the algorithm is applied to four classical problems of engineering optimization. Compared with well-known heuristic algorithms, the BFGO algorithm has better performance.

Keywords: Heuristic optimization algorithm, Bamboo forest growth optimization algorithm, Engineering optimization

1 Introduction

Optimization problems widely exist in the fields of social production activities such as intelligent computing, mathematical science, engineering optimization, distribution scheduling, and so on. Past research has been looking for more efficient and accurate methods to solve optimization problems [1]. Among them, the heuristic algorithm [2] is a kind of solution method proposed to approximate the optimal solution as much as possible relative to the optimization algorithm.

Heuristic algorithms can be divided into three branches (see Figure 1): simple heuristics, meta-heuristics, and hyper-heuristics. Simple heuristic algorithms are generally deterministic algorithms for fixed structures and parameters, which have a unique global optimal solution [3]. Including local search algorithm, greedy algorithm [4], stereotype algorithm [5], hill-climbing algorithm [6], etc. Most of the traditional heuristic algorithms are based on the greedy idea, lack of consideration of the global optimum, and the solution result strongly depends on the initial value; meta-heuristic optimization algorithms [7] draw inspiration from behaviors, experiences, and rules in production activities, build optimization models for specific problems, and design optimization algorithms with certain natural laws. It is mainly divided into four categories: evolution-based algorithms (EA) [8] mainly represented by genetic algorithms (GA) [9], differential evolution (DE) [10] and quasi-affine transformation evolution (QUATRE) [11]; algorithms based on swarm intelligence such as particle swarm optimization (PSO) [12-15], monkey king evolution (MKE) [16], cuckoo search algorithm (CSA) [17], cat swarm optimization (CSO) [18], fish migration optimization (FMO) [19-20], flower pollination algorithm (FPA) [21], etc., by simulating the special behavior of the group, form a global optimization algorithm process; human-based algorithms [22] include teaching-learning-based optimization (TLBO) [23], group search optimizer (GSO) [24], social-based algorithm (SBA) [25] and other inspiration algorithms for human teaching, social and emotional behavior; algorithms based on physics and chemistry such as sine cosine algorithm (SCA) [26], simulated annealing (SA) [27], water cycle algorithm (WCA) [28], etc. come from physical rules and chemical reactions in the universe. The hyper-heuristic is a combination of multiple heuristic algorithms or multiple strategies and operations. Just like ensemble learning [29] in machine learning, the core is the feedback mechanism and selection mechanism, which can be roughly divided into random selection-based [30], tabu search-based [31], etc., it is suitable for solving cross-domain problems.

In this study, a novel meta-heuristic algorithm, the BFGO algorithm, is proposed. Its inspiration comes from the unique growth law of bamboo forests in nature [32]. The indifference between this job and likewise heuristic algorithm is that the cluster concept is introduced into the algorithm corresponding to the bamboo whip of bamboo forest, the method of communication and reference between clusters is added, and the stochastic process model of high growth of bamboo shoots is introduced into the algorithm's formula. We select 30 test functions of CEC2017 [33] as the achievement of the benchmarking algorithm, contrast the BFGO algorithm with other swarm intelligence optimization algorithms, and apply it to four engineering optimization problems [34-35]. Experiments show that the BFGO algorithm has strong competitiveness compared with other heuristic algorithms.

*Corresponding Author: Shu-Chuan Chu; E-mail: scchu0803@gmail.com
DOI: 10.53106/160792642023012401001

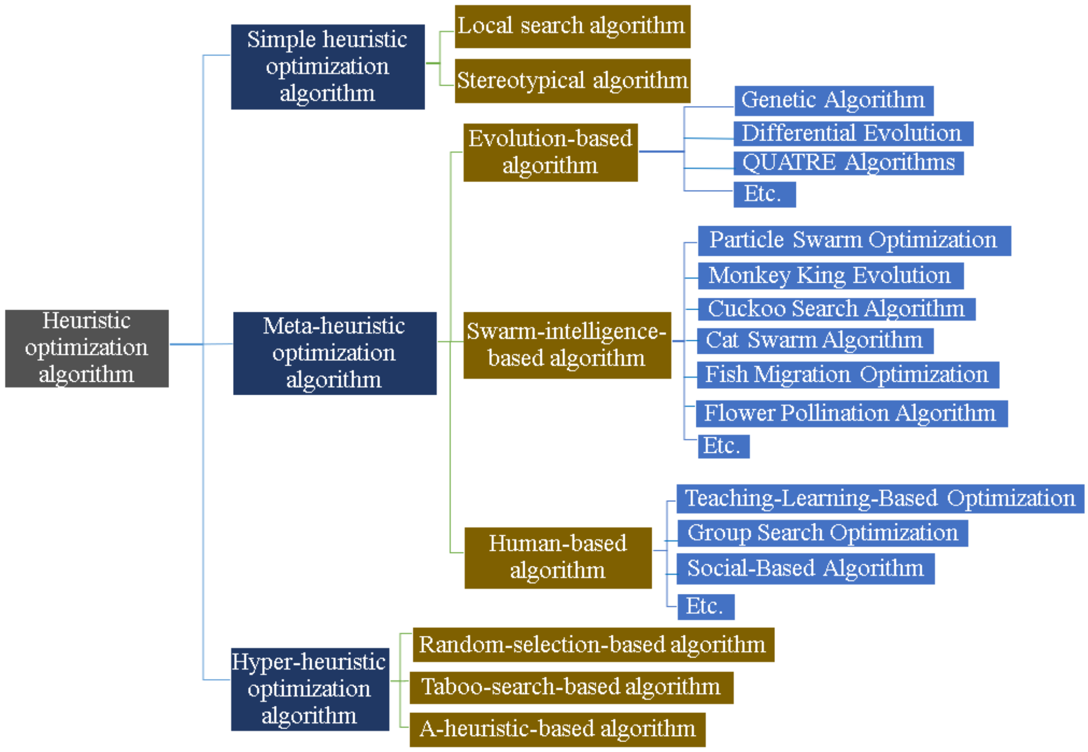

Figure 1. Division of heuristic optimization algorithms

Section 2 will briefly introduce the formula principle of the iterative search of the BFGO algorithm. Section 3 describes the optimization results of the algorithm under benchmark function tests in detail. Section 4 discusses the performance of the algorithm applied to four engineering optimization problems. Finally, in section 5, the experimental results are analyzed to draw conclusions and propose future research directions.

2 The Bamboo Forest Growth Optimization Algorithm

This section first explains the inspiration for the proposed algorithm, and then introduces the mathematical optimization model.

2.1 Inspiration

Bamboo is a grass plant of the Poaceae and Bamboo genus, but it can have the height of a tree. Young bamboo shoots can grow up to 1 meter a day. According to Guihua Jin [36], this rapid growth serves as a key innovative trait of woody bamboo, enabling it to compete with other trees to adapt to the forest environment.

Bamboo in the shooting stage grows in the rain, but when it grows into bamboo, it will not grow for three to five years. After that, the bamboo will suddenly exert force and grow at an astonishing rate. In the three or five years when the bamboo does not grow, its roots grow wildly underground - deep and wide. “Deep” refers to the depth of the ground. Bamboo roots can penetrate incredible stone bodies as strong as steel. “Wide” means that the root system of bamboo roots can be spread for several kilometers. On the land of several square kilometers, bamboo can easily obtain the nutrients and rainwater it needs.

A distinct unique physiological of bamboo is that the underground stem, also called a bamboo whip, grows reclining, with nodes and abounding and dense, with many fibrous roots and buds growing on the nodes. Figure 2 shows this. Bamboo whips not only store and provide a lot of nutrients for fast-growing bamboo forests but also are the core force for expanding the territory of bamboo forests. They can grow and extend randomly in all directions. These bamboos with connected rhizomes are crisscrossed and exchange nutrients. They will transfer nutrients to each other and share the environmental pressure together.

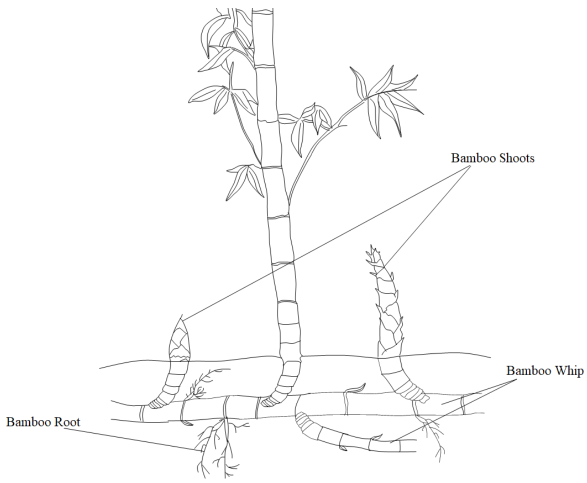

Figure 2. The structure of bamboo

2.2 Mathematical Model

The growth process of bamboo can be summarized into the stages of budding, shoot growth, rapid growth, adulthood, and flowering and death. This part maps the bamboo growth stages with the algorithm optimization process, constructs a mathematical model based on the bamboo growth process of bamboo root extension, bamboo forest growth, and bamboo flowering as the optimization principle, and proposes the BFGO algorithm.

2.2.1 The All-round Extension of The Bamboo Whip (Exploitation)

During this period, the bamboo whip will absorb nutrients in the soil, store energy, the meristem of the whip section begins to divide, and the underground stem begins to randomly lengthen and expand its territory. A part of the strong buds on the bamboo whip began to germinate and differentiate to become bamboo shoots. Another part of the bud does not grow out of the ground, but grows sideways and develops into a new underground stem. The root system on the bamboo whip extends in different directions. The main directions are the direction of the group cognition item, the direction of the bamboo whip memory item, and the direction of the central item of the bamboo forest. such as Equation (1) shown.

$$X_{t+1}=\begin{cases} X\_G+Q\times\left(c_1\times X\_G-X_t\right)\times\cos\alpha, & r_1<0.4 \\ X\_P(k)+Q\times\left(c_1\times X\_P(k)-X_t\right)\times\cos\beta, & 0.4\le r_2<0.7 \\ C(k)+Q\times\left(c_1\times C(k)-X_t\right)\times\cos\gamma, & else \end{cases},\tag{1}$$

$$\cos\alpha=\frac{\mathbf{X_t}\cdot\mathbf{X\_G}}{|\mathbf{X_t}|\times|\mathbf{X\_G}|},\tag{2}$$

$$\cos\beta=\frac{\mathbf{X_t}\cdot\mathbf{X\_P(k)}}{|\mathbf{X_t}|\times|\mathbf{X\_G}|},\tag{3}$$

$$\cos\gamma=\frac{\mathbf{X_t}\cdot\mathbf{C(k)}}{|\mathbf{X_t}|\times|\mathbf{X\_G}|},\tag{4}$$

$$Q=2-\frac{t}{T},\tag{5}$$

where $\mathbf{X\_G}$ represents the global optimal individual, $\mathbf{X\_P(k)}$ represents the optimal individual on the k-th bamboo whip, and $\mathbf{C(k)}$ represents the center point of the k-th bamboo whip. α, β and γ here represents the direction of the current individual on the group cognition term, the bamboo whip memory term, and the bamboo forest center term, as shown in Equation (2-4). Q is shown in Equation (5), which decreases from 2 to 1. c_1 to a random number between 0-2.

2.2.2 The Bamboo Growth Stage (Exploration)

This period can be regarded as the selection stage of the algorithm because only a small part of the unearthed bamboo shoots can grow into bamboo, and the bamboo shoots that have no hope of growing can only fend for themselves. Bamboo shoots that have a chance to grow will gain sufficient energy to grow rapidly in a short period of time. The stochastic process model of bamboo shoots height growth is shown in Figure 3 [37].

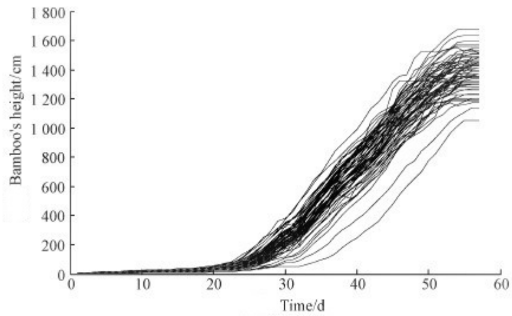

Figure 3. Measured bamboo shoots growth curve

Combined with the growth characteristics of bamboo shoots, Sloboda [38-39] used mathematical analysis methods to derive the following growth differential equation:

$$\frac{dy}{dt}=\frac{b\times y}{t^{\lambda}\ln\left(\frac{c}{y}\right)}.\tag{6}$$

Wherein: $\lambda>1$, b, $c>0$ are parameters, t is the time, for the high growth of the bamboo shoot stage, t is the number of growth days, and y is the bamboo height. Its integral form:

$$y=c\times e^{-d}\times e^{\left(\frac{b}{(\lambda-1)\times t^{(\lambda-1)}}\right)}.\tag{7}$$

Wherein: d is the integral constant, which can be given according to the environment in which the tree grows. The equation is then sorted out to:

$$X(\omega,t)=SI\times e^{\frac{b}{k\times t^{k}}}.\tag{8}$$

Wherein: SI is the maximum bamboo height under the condition of a certain standing ground, which changes with the change of the standing conditions; b and k is the shape parameter of the model. Therefore, the value of the SI can be given according to the growing environment of each bamboo plant.

According to the Equation (6-8), we define the temporary population of bamboo growth. Only when the individual is better than the original population will the position be replaced. The update formula is as follows,

$$X_{temp} = \begin{cases} X_t + X\_D \times \Delta H \\ X_t - X\_D \times \Delta H \end{cases}. \tag{9}$$

$$X\_D = 1 - \left| \frac{X_t - C(k) + 1}{X\_G - C(k) + 1} \right|. \tag{10}$$

$$\Delta H = \frac{q(t) - q(t-1)}{X\_G - X_t}. \tag{11}$$

$$q(t) = X\_G \times e^{-d} \times e^{\frac{b}{\varphi \times t^{\varphi}}}. \tag{12}$$

Where X_D represents the relationship between the distance from the particle to the center position and the distance between the optimal individual of the group and the center position, ΔH represents the difference between the two iterations of growth, and q represents the cumulative growth of the t-th generation. The range of d is (-1, 1), and both b and f are fixed parameters, which are equivalent to the site conditions of bamboo.

3 Experiment and Analysis

In this section, we choose the CEC2017 benchmark to test the optimization performance of the BFGO algorithm. There are four types of functions, namely unimodal function, multimodal function, hybrid function, and composite function. For functions, the variables are divided into subcomponents randomly.

The BFGO algorithm is run 30 times on each benchmark function, and the individual dimension is 10 dimensions. The final results are compared with typical swarm intelligence optimization algorithms such as PSO, BA [40], GWO [41], SSA [42], and ALO [43]. Count their average optimal values (mean) and the corresponding standard deviations (std), and the comparison results are shown in Table 1 to Table 4. The numbers in bold are the best value, the underlined numbers are the same optimal.

Table 1. Comparison of optimization results for unimodal benchmark functions

Function	BFGO		PSO		BA		GWO		SSA		ALO	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F1	576973	2266725	1.4E+08	4.2E+08	394459	102709	1.3E+07	6.5E+07	3021.69	2668.76	**1236.23**	1856.37
F2	316.983	219931	2.3E+08	7E+08	**205.126**	12.4499	9417346	1.7E+07	3874.03	4883.62	263.865	51.4504
F3	309.642	143.076	300.001	0.00575	301.065	0.3109	984.579	1209.85	300	7E-10	300	3.2E-08

Table 2. Comparison of optimization results for multimodal benchmark functions

Function	BFGO		PSO		BA		GWO		SSA		ALO	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F4	**400.848**	12.7981	428.166	39.996	406.978	16.6859	410.73	9.27256	411.855	19.345	407.272	12.0612
F5	528.503	8.19826	528.539	12.4368	556.381	18.8366	**513.105**	6.91138	523.62	10.3406	522.925	10.9237
F6	603.44	8.81937	606.998	4.18462	640.195	10.8192	**600.394**	0.65243	608.537	7.07277	606.255	4.85603
F7	**717.845**	9.58993	734.206	7.80555	827.697	36.8237	724.887	7.37586	735.71	12.5414	734.465	11.0117
F8	825.024	7.04583	824.407	7.30835	850.833	17.3578	**813.022**	6.72336	822.402	9.81936	820.596	10.2841
F9	940.679	80.3195	913.093	14.691	1703.46	511.944	**904.735**	12.3403	912.205	32.7125	979.477	116.741
F10	**1180.35**	244.907	1868.68	260.466	2344.71	325.935	1536.25	308.833	1864.16	324.609	1806.06	329.948

Table 3. Comparison of optimization results for hybrid benchmark functions

Function	BFGO		PSO		BA		GWO		SSA		ALO	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F11	**1117.45**	64.6022	1170.17	52.6566	1205.06	71.1007	1130.12	26.8664	1193.42	83.588	1171.16	38.062
F12	4347221	7286925	826766	2506778	1121672	752256	**648852**	886667	1920354	1988137	1211446	122166
F13	20164.6	27321.8	**5158.53**	9426.94	15980.2	13670.1	12796.8	8755.05	14921.1	11850.2	16356.5	10407
F14	**1467.24**	189.979	1467.56	31.7151	2961.62	2135.16	2736.2	1736.99	1554.02	158.138	1504.19	33.651
F15	3465.99	1911.96	**1667.84**	347.146	14599.2	9685.87	3241.76	1624.18	4109.35	2533.67	6054.5	4799.7
F16	**1622.16**	101.493	1707.23	78.4871	1950.87	179.311	1702.29	104.922	1737.23	108.56	1811.76	133.62
F17	**1740.72**	21.1373	1785.48	47.4374	1830.17	71.8976	1743.37	16.6044	1756.19	17.5085	1766.3	33.620
F18	24842.8	46915.2	18132.8	17066.8	15638.6	13877.1	27335.7	16109.9	18473.4	13025	**14476.3**	11425
F19	4762.11	5154.9	**4007.7**	7863.52	5833.33	4676.98	4196.76	4603.56	4615.16	4189.62	9093.07	7262
F20	**2040.02**	37.723	2075.95	51.8192	2173.77	80.8977	2047.15	34.1945	2103.13	58.9355	2090.37	43.32

Table 4. Comparison of optimization results for composite benchmark functions

Function	BFGO		PSO		BA		GWO		SSA		ALO	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F21	**2213.82**	22.8623	2308.31	48.8981	2304.87	76.9744	2296.89	43.8736	2253.17	59.2294	2301.87	46.8713
F22	**2268.88**	23.1277	2316.37	16.1835	2371	309.973	2306.19	4.67076	2292.7	28.2099	2303.56	1.8565
F23	2635.9	14.3636	2640.66	14.8212	2673.72	28.8564	**2619.47**	10.4848	2620.5	8.66221	2623.06	11.5402
F24	**2647.11**	128.342	2746.55	76.3678	2820.84	35.753	2737.46	42.7075	2731.12	63.4308	2732.38	63.895
F25	**2899.91**	55.837	2940.76	33.2671	2922.29	43.9561	2937.16	13.5874	2926.1	24.469	2923.32	23.7021
F26	2999.54	149.447	3034.86	210.435	3350.63	398.188	2974.16	246.201	**2909.45**	32.582	2960.39	194.981
F27	**3078.35**	3.47375	3121.92	31.5101	3147.17	39.5008	3094.62	2.63972	3092.12	2.8296	3096.45	4.03932
F28	3272.57	7.19369	3361.34	90.0144	3240.87	75.6372	3310.49	115.973	**3191.32**	86.4504	3308.7	136.732
F29	3244	54.9167	**3217.18**	68.5378	3320.7	115.984	3184.22	35.2981	3190.66	33.4229	3232.62	51.5157
F30	**23293.9**	26515.3	972659	1919798	41192	49590.4	494143	654340	323502	539172	209422	466620

It can be seen from Table 1 that BFGO has no advantage in the face of unimodal functions. It is not as good as the average of BA, SSA, and ALO in 30 runs, but it can beat the PSO and GWO algorithms, so the BFGO algorithm has great development capabilities. lacking. From the comparison results in Table 2, it can be seen that the BFGO algorithm obtains the third best, second only to the GWO algorithm which obtains the fourth-best. As can be seen from Table 3, BFGO achieves the fifth optimality and is the best among the mixing functions. So the ability of BFGO to explore local optima is competitive. In the comparison of the test results of the composite function in Table 4, we can see that the BFGO algorithm is still in the leading state, and it has obtained six optimal values, surpassing other algorithms. From the test results of F1-F30 functions, BFGO has better performance on complex problems, and can well balance the ability of exploration and exploitation.

4 BFGO Algorithm for Classical Engineering Problems

In this paper, the BFGO algorithm experiments on four engineering optimization problems: tension spring design optimization [44], pressure vessel design optimization [45], welded beam design optimization [46], and speed reducer design optimization problem [47], each of which has a different constraint. Each experiment is run 30 times to take the optimal value to compare the BFGO algorithm with the PSO, BA, GWO, SSA, and ALO algorithms to detect the performance of the algorithm in practical applications.

4.1 Tension Spring Design Optimization Problem

The purpose of the optimized design of this engineering problem is to find a minimum mass of the tension/compression spring under certain limits of shear stress, surge frequency, and deflection. The three design optimization variables for this problem: wire diameter (l_1), average coil diameter (l_2), and the number of effective coils (l_3) are shown in Figure 4. The mathematical optimization process is described below.

$$f(\vec{l}) = (l_3 + 2)l_1^2 l_2$$

Subject to:

$$g_1(\vec{l}) = 1 - \frac{l_2^2 l_3}{7178 l_1^4} \le 0$$

$$g_2(\vec{l}) = \frac{4l_2^2 - l_1 l_2}{12566 l_1^3 l_2 - l_1^4} + \frac{1}{5108 l_1^2} - 1 \le 0$$

$$g_3(\vec{l}) = 1 - \frac{140.45 l_1}{l_2^2 l_3} \le 0$$

$$g_4(\vec{l}) = \frac{l_1 + l_2}{1.5} - 1 \le 0$$

With $0.05 \le l_1 \le 2.0$, $0.25 \le l_2 \le 1.3$, $2.0 \le l_3 \le 15.0$;

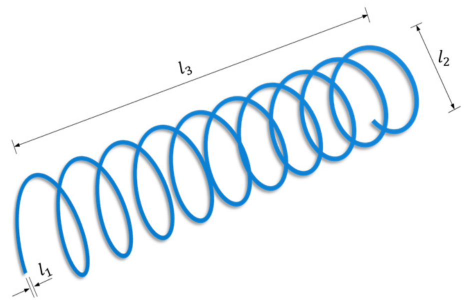

Figure 4. Tension spring design

Table 5. The optimization results for tension spring design optimization

Algorithm	Optimize variables			Minimum weight
	d(l1)	D(l2)	N(l3)	
BFGO	0.05	0.282	2	**0.00282**
PSO	0.0732	0.6191	6.3038	0.02758
BA	0.0607	0.3658	3.0579	0.00282
GWO	0.05	0.282	2	**0.00282**
SSA	0.0621	0.4054	1.6257	0.00567
ALO	0.05	0.282	2	**0.00282**

The optimization results of the BFGO algorithm for Tension Spring Design are compared with the five swarm intelligence algorithms of PSO, BA, GWO, SSA, and ALO, as shown in Table 5. It can be seen that the BFGO algorithm, GWO, and ALO have reached the optimal value, the minimum weight value is 0.00282, and the final three parameter values are 0.05, 0.282, and 2.

4.2 Pressure Vessel Design Optimization Problem

The purpose of the optimization design of this engineering problem is to find the minimum value of the total cost of a pressure vessel, such as a closed cylindrical vessel as shown in Figure 5, where the head has a hemispherical shape, where the total cost such as material, molding and welding costs. The problem has four optimization variables: l_1, l_2, l_3, l_4. Math The description is as follows.

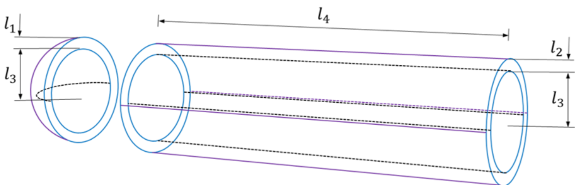

Figure 5. Pressure vessel design

$$f(\vec{l}) = 0.6224 l_1 l_3 l_4 + 1.7781 l_2 l_3^2 + 3.1661 l_1^2 l_4 + 19.84 l_1^2 l_3$$

Subject to:

$$g_1(\vec{l}) = -l_1 + 0.0193 l_3 \le 0$$

$$g_2(\vec{l}) = -l_2 + 0.00954 l_3 \le 0$$

$$g_3(\vec{l}) = -\pi l_3^2 l_4^2 - \frac{4}{3}\pi l_3^3 + 1296000 \le 0$$

$$g_4(\vec{l}) = l_4 - 240 \le 0$$

With

$1 \times 0.0625 \le l_1$, $l_2 \le 99 \times 0.0625$,

$10.0 \le l_3$, and $l_4 \le 200.0$;

Table 6. The optimization results for pressure vessel design optimization

Algorithm	Optimize variables				Minimum
	Ts(l1)	Th(l2)	R(l3)	L(l4)	
BFGO	0.193	0.095	10	64.14	**108.917**
PSO	2.932	10.05	19.47	55.7	13593.2
BA	0.285	0.211	10.56	62.74	19266.6
GWO	0.193	0.117	10	64.25	112.902
SSA	2.899	51.58	7.654	87	10165.5
ALO	0.193	0.096	10	64.13	108.928

The optimization results of the BFGO algorithm for pressure vessels are compared with the five swarm intelligence algorithms of PSO, BA, GWO, SSA, and ALO, as shown in Table 6. It can be seen that the BFGO algorithm is better than other algorithms, reaching the optimal value, the minimum cost is 108.917, and the final four parameter values are 0.193, 0.095, 10, and 64.14.

4.3 Welded Beam Design Optimization Problem

The purpose of optimization design of this engineering problem is to reduce the fabrication of welded beams under the constraints and find four optimization variables: the thickness of weld (l_1), the length of the clamped bar (l_2), height of the bar (l_3), and thickness of the bar (l_4). The structure diagram is shown in Figure 6, and the mathematical description of the engineering problem is as follows.

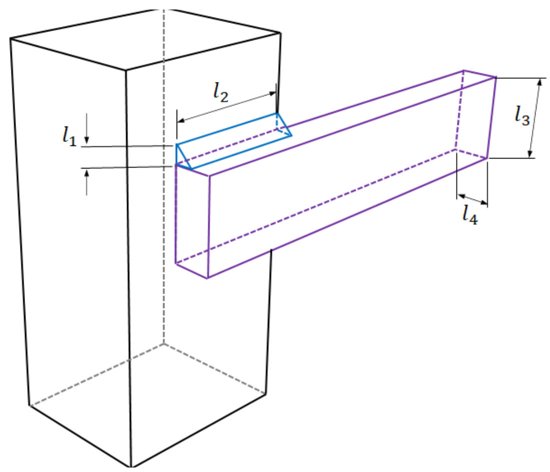

Figure 6. Welded beam design

$$f(\vec{l}) = 1.10471 l_1^2 l_2 + 0.04811 l_3 l_4 (14.0 + l_2)$$

Subject to:

$$g_1(\vec{l}) = \tau(\vec{l}) - 13600 \le 0$$

$$g_2(\vec{l}) = \sigma(\vec{l}) - 30000 \le 0$$

$$g_3(\vec{l}) = l_1 - l_4 \le 0$$

$$g_4(\vec{l}) = 0.10471 l_1^2 + 0.04811 l_3 l_4 (14 + l_2) - 5 \le 0$$

$$g_5(\vec{l}) = 0.125 - l_1 \le 0$$

$$g_6(\vec{l}) = \delta(\vec{l}) - 0.25 \le 0$$

$$g_7(\vec{l}) = 6000 - Pc(\vec{l}) \le 0$$

With

$$\tau(\vec{l}) = \sqrt{(\tau')^2 + (2\tau'\tau'')\frac{l_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{6000}{\sqrt{2} l_1 l_2}$$

$$\tau'' = \frac{MR}{J}$$

$$M = 6000\left(14 + \frac{l_2}{2}\right)$$

$$R = \sqrt{\frac{l_2^2}{4} + \left(\frac{l_1 + l_3}{2}\right)^2}$$

$$J = 2\left\{l_1 l_2 \sqrt{2}\left[\frac{l_2^2}{12} + \left(\frac{l_1 + l_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{l}) = \frac{504000}{l_3^2 l_4}$$

$$\delta(\vec{l}) = \frac{65856000}{(30\times 10^6) l_3^2 l_4}$$

$$Pc(\vec{l}) = \frac{4.013(30\times 10^6)\sqrt{\frac{l_3^2 l_4^6}{36}}}{196} \times \left(1 - \frac{l_3\sqrt{\frac{30\times 10^6}{4(12\times 10^6)}}}{28}\right)$$

With

$0.1 \le l_1$, $0.1 \le l_2$, $l_3 \le 10$, $l_4 \le 2$;

Table 7. The optimization results for welded beam design optimization

Algorithm	Optimize variables				Minimum cost
	l1	l2	l3	l4	
BFGO	0.203	3.539	9.047	0.206	**1.726085**
PSO	0.216	9.291	6.658	0.502	3.729981
BA	0.164	8.718	8.349	0.245	2.041104
GWO	0.204	3.514	9.054	0.206	1.72657
SSA	0.995	1.839	2	2	1.09E+15
ALO	0.203	3.616	10	0.203	1.826582

The optimization results of the BFGO algorithm on Welded Beam are compared with the five swarm intelligence algorithms of PSO, BA, GWO, SSA, and ALO, as shown in Table 7. It can be seen that the BFGO algorithm is better than other algorithms, reaching the optimal value, the minimum cost is 1.726085, and the final four parameter values are 0.203, 3.539, 9.047, and 0.206.

4.4 Speed Reducer Design Optimization Problem

The purpose of this engineering problem is to minimize the weight of the reducer. There are seven optimization variables. The construction diagram is shown in Figure 7, and its mathematical description is as follows.

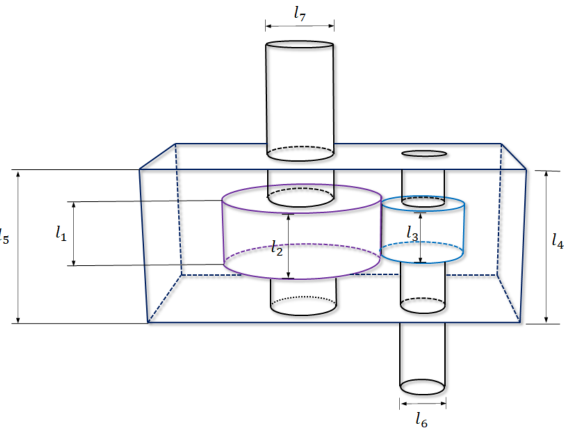

Figure 7. Speed reducer design

$$f(\vec{l}) = 0.7854 l_1 l_2^2 (3.3333 l_3^2 + 14.9334 l_3 - 43.0934) - 1.508 l_1 (l_6^2 + l_7^2) + 7.4777 (l_6^2 + l_7^2) + 0.7854 \begin{pmatrix} l_4 l_6^2 \\ + l_5 l_7^2 \end{pmatrix}$$

Subject to:

$$g_1(\vec{l}) = \frac{27}{l_1 l_2^2 l_3} - 1 \le 0$$

$$g_2(\vec{l}) = \frac{397.5}{l_1 l_2^2 l_3^2} - 1 \le 0$$

$$g_3(\vec{l}) = \frac{1.93 l_4^3}{l_2 l_3 l_6^4} - 1 \le 0$$

$$g_4(\vec{l}) = \frac{1.93 l_5^3}{l_2 l_3 l_7^4} - 1 \le 0$$

$$g_5(\vec{l}) = \frac{1.0}{110 l_6^3}\sqrt{\left(\frac{745.0 l_4}{l_2 l_3}\right)^2 + 16.9\times 10^6} - 1 \le 0$$

$$g_6(\vec{l}) = \frac{1.0}{85 l_7^3}\sqrt{\left(\frac{745.0 l_5}{l_2 l_3}\right)^2 + 157.5\times 10^6} - 1 \le 0$$

$$g_7(\vec{l}) = \frac{l_2 l_3}{40} - 1 \le 0$$

$$g_8(\vec{l}) = \frac{5 l_2}{l_1} - 1 \le 0$$

$$g_9(\vec{l}) = \frac{l_1}{12 l_2} - 1 \le 0$$

$$g_{10}(\vec{l}) = \frac{1.5 l_6 + 1.9}{l_4} - 1 \le 0$$

$$g_{11}(\vec{l}) = \frac{1.1 l_7 + 1.9}{l_5} - 1 \le 0$$

With

$2.6 \le l_1 \le 3.6$, $0.7 \le l_2 \le 0.8$, $17 \le l_3 \le 28.7$,

$3 \le l_4 \le 8.3$, $7.8 \le l_5 \le 8.3$, $2.9 \le l_6 \le 3.9$,

$5.0 \le l_7 \le 5.5$;

Table 8. The optimization results for speed reducer design optimization

Algorithm	Optimize variables							Minimum weight
	l1	l2	l3	l4	l5	l6	l7	
BFGO	3.6	0.8	28	7.3	7.8	3.9	5.2849	**201613.2**
PSO	3.0293	0.7705	27.8956	7.67	8.1004	3.8967	5.1882	437488.8
BA	3.6	0.8	28	7.3	8.293	3.9	5.2882	201614.2
GWO	3.6	0.8	28	7.3	7.8	3.9	5.2861	**201613.2**
SSA	5	5	5	5	5	5	5	17052488
ALO	3.6	0.8	28	7.3	7.9181	3.9	5.287	**201613.2**

The optimization results of Speed Reducer by BFGO algorithm are compared with the five swarm intelligence algorithms of PSO, BA, GWO, SSA, and ALO, as shown in Table 8. It can be seen that the BFGO algorithm, GWO, and ALO algorithms have reached the optimal value, the minimum cost is 201613.2, and the final four parameter values are 3.6, 0.8, 28, 7.3, 7.8, 3.9, and 5.2849 respectively.

5 Conclusion

Starting from the growth characteristics of bamboo forests, a new intelligent group optimization algorithm is proposed. Inspired by the underground bamboo whip, the proposed algorithm is divided into several bamboo whip groups, simulating the bamboo whip extension, bamboo height random growth, and underground root system crisscrossing several stages in the growth process of bamboo forest. Test on 30 mathematical functions, the BFGO algorithm is found to be highly competitive in local exploration by analyzing the algorithm's ability to exploit and explore. In addition, in order to test the performance of the BFGO algorithm in practical engineering applications, four typical engineering optimization problems are solved.

In the comparison of the results of the BFGO algorithm with other heuristic algorithms, it is concluded that the BFGO algorithm is very competitive in dealing with optimization problems, but the exploitation ability is not outstanding. In the future, we will continue to study the optimized version [48-51] of the BFGO algorithm and use it in more application areas.

References

[1] J. S. Logsdon, L. T. Biegler, Accurate solution of differential-algebraic optimization problems, *Industrial & engineering chemistry research*, Vol. 28, No. 11, pp. 1628-1639, November, 1989.

[2] S. Lin, B. W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem, *Operations research*, Vol. 21, No. 2, pp. 498-516, April, 1973.

[3] G. G. Wang, A. H. Gandomi, A. H. Alavi, G. S. Hao, Hybrid krill herd algorithm with differential evolution for global numerical optimization, *Neural Computing and Applications*, Vol. 25, No. 2, pp. 297-308, August, 2014.

[4] Z. Zhang, S. Schwartz, L. Wagner, W. Miller, A greedy algorithm for aligning DNA sequences, *Journal of Computational biology*, Vol. 7, No. 1-2, pp. 203-214, February, 2000.

[5] A. R. Wagner, Robots that stereotype: creating and using categories of people for human-robot interaction, *Journal of Human-Robot Interaction*, Vol. 4, No. 2, pp. 97-124, September, 2015.

[6] D. B. Skalak, Prototype and feature selection by sampling and random mutation hill climbing algorithms, In: W. W. Cohen, H. Hirsh (Eds.), *Machine Learning Proceedings*, Morgan Kaufmann, Skalak, 1994, pp. 293-301.

[7] M. Lukasiewycz, M. Glaß, F. Reimann, J. Teich, Opt4J: a modular framework for meta-heuristic optimization, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, Dublin Ireland, 2011, pp. 1723-1730.

[8] X. Xue, J. Chen, D. Chen, Matching Biomedical Ontologies Through Compact Hybrid Evolutionary Algorithm, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 10, No. 1, pp. 110-117, January, 2019.

[9] J. H. Holland, Genetic algorithms, *Scientific American*, Vol. 267, No. 1, pp. 66-73, July, 1992.

[10] K. V. Price, Differential Evolution, in: I. Zelinka, V. Snášel, A. Abraham (Eds.), *Handbook of optimization*, Springer, 2013, pp. 187-214.

[11] Z. Meng, J. S. Pan, QUasi-Affine TRansformation Evolution with External ARchive (QUATRE-EAR): an enhanced structure for differential evolution, *Knowledge-Based Systems*, Vol. 155, pp. 35-53, September, 2018.

[12] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95-international conference on neural networks*, Perth, WA, Australia, 1995, pp. 1942-

1948.

[13] J.-F. Chang, S.-C. Chu, J. F. Roddick, J.-S. Pan, A parallel particle swarm optimization algorithm with communication strategies, *Journal of Information Science and Engineering*, Vol. 21, No. 4, pp. 809-818, 2005.

[14] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, G. J. Kim, A PSO based energy efficient coverage control algorithm for wireless sensor networks, *Computers, Materials & Continua*, Vol. 56, No. 3, pp. 433-446, December, 2018.

[15] H. Wang, H. Sun, C. Li, S. Rahnamayan, J. S. Pan, Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, Vol. 223, pp. 119-135, February, 2013.

[16] J. S. Pan, Z. Meng, S. C. Chu, H. R Xu, Monkey king evolution: an enhanced ebb-tide-fish algorithm for global optimization and its application in vehicle navigation under wireless sensor network environment, *Telecommunication Systems*, Vol. 65, No. 3, pp. 351-364, July, 2017.

[17] J. S. Pan, P. C. Song, S. C. Chu, Y. J. Peng, Improved compact cuckoo search algorithm applied to location of drone logistics hub, *Mathematics*, Vol. 8, No. 3, Article No. 333, March, 2020.

[18] M. Bahrami, O. Bozorg-Haddad, X. Chu, Cat swarm optimization (CSO) algorithm, in: O. Bozorg-Haddad (Eds.), *Advanced optimization by nature-inspired algorithms*, Springer Singapore, 2018, pp. 9-18.

[19] J. S. Pan, P. Hu, S. C. Chu, Binary fish migration optimization for solving unit commitment, *Energy*, Vol. 226, Article No. 120329, July, 2021.

[20] B. Guo, Z. Zhuang, J. S. Pan, S. C. Chu, Optimal design and simulation for PID controller using fractional-order fish migration optimization algorithm, *IEEE Access*, Vol. 9, pp. 8808-8819, January, 2021.

[21] J. S. Pan, T. K. Dao, T. S. Pan, T. T. Nguyen, S. C. Chu, J. F. Roddick, An Improvement of Flower Pollination Algorithm for Node Localization Optimization in WSN, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 8, No. 2, pp. 486-499, March, 2017.

[22] X. Xue, H. Yang, J. Zhang, Using population-based incremental learning algorithm for matching class diagrams, *Data Science and Pattern Recognition*, Vol. 3, No. 1, pp. 1-8, March, 2019.

[23] R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, *Computer-aided design*, Vol. 43, No. 3, pp. 303-315, March, 2011.

[24] S. He, Q. H. Wu, J. R. Saunders, Group search optimizer: an optimization algorithm inspired by animal searching behavior, *IEEE transactions on evolutionary computation*, Vol. 13, No. 5, pp. 973-990, October, 2009.

[25] F. Ramezani, S. Lotfi, Social-based algorithm (SBA), *Applied Soft Computing*, Vol. 13, No. 5, pp. 2837-2856, May, 2013.

[26] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowledge-based systems*, Vol. 96, pp. 120-133, March, 2016.

[27] P. J. Van Laarhoven, E. H. Aarts, *Simulated annealing: Theory and application*, Springer, 1987, pp. 7-15.

[28] A. Sadollah, H. Eskandar, H. M. Lee, D. G. Yoo, J. H. Kim, Water cycle algorithm: a detailed standard code, *SoftwareX*, Vol. 5, pp. 37-43, 2016.

[29] R. Polikar, Ensemble Learning, In: C. Zhang, Y. Ma (Eds.), *Ensemble machine learning*, Springer, 2012, pp. 1-34.

[30] B. Du, L. Zhang, Random-selection-based anomaly detector for hyperspectral imagery, *IEEE Transactions on Geoscience and Remote sensing*, Vol. 49, No. 5, pp. 1578-1589, May, 2011.

[31] H. C. Huang, S. C. Chu, J. S. Pan, C. Y. Huang, B. Y. Liao, Tabu search based multi-watermarks embedding algorithm with multiple description coding, *Information Sciences*, Vol. 181, No. 16, pp. 3379-3396, August, 2011.

[32] B. Wang, W. J. Wei, C. J. Liu, W. Z. You, X. Niu, R. Z. Man, Biomass and carbon stock in Moso bamboo forests in subtropical China: characteristics and implications, *Journal of Tropical Forest Science*, Vol. 25, No. 1, pp. 137-148, January, 2013.

[33] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, K. M. Jambi, LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, *2017 IEEE Congress on Evolutionary Computation (CEC)*, Donostia, Spain, 2017, pp. 145-152.

[34] J. R. Buck, T. W. Hill Jr, Laplace transforms for the economic analysis of deterministic problems in engineering, *The Engineering Economist*, Vol. 16, No. 4, pp. 247-263, 1971.

[35] Y. F. Huang, C. H. Hsu, J. W. Wang, H. L. Hung, J. Jenq, Energy Efficiency of Dynamically Distributed Clustering Routing for Naturally Scattering Wireless Sensor Networks, *Journal of Network Intelligence*, Vol. 3, No. 1, pp. 50-57, February, 2018.

[36] G. Jin, P. F. Ma, X. Wu, L. Gu, M. Long, C. Zhang, D. Z. Li, New Genes Interacted With Recent Whole-Genome Duplicates in the Fast Stem Growth of Bamboos, *Molecular biology and evolution*, Vol. 38, No. 12, pp. 5752-5768, December, 2021.

[37] Y. Shi, E. Liu, G. Zhou, Z. Shen, S. Yu, Bamboo shoot growth model based on the stochastic process and its application, *Scientia Silvae Sinicae*, Vol. 49, No. 9, pp. 89-93, September, 2013.

[38] B. Sloboda, Zur Darstellung von Wachstumsprozessen mit Hilfe von Differentialgleichungen erster Ordnung, *Mitteilungen der Baden-Württembergischen Forstlichen Versuchs-und Forschungsanstalt*, 1971.

[39] L. Rasche, L. Fahse, A. Zingg, H. Bugmann, Enhancing gap model accuracy by modeling dynamic height growth and dynamic maximum tree height, *Ecological Modelling*, Vol. 232, pp. 133-143, May, 2012.

[40] X. S. Yang, X. He, Bat algorithm: literature review and applications, *International Journal of Bio-inspired computation*, Vol. 5, No. 3, pp. 141-149, July, 2013.

[41] E. Emary, H. M. Zawbaa, A. E. Hassanien, Binary grey wolf optimization approaches for feature selection,

Neurocomputing, Vol. 172, pp. 371-381, January, 2016.

[42] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, Vol. 114, pp. 163-191, December, 2017.

[43] S. Mirjalili, P. Jangir, S. Saremi, Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems, *Applied Intelligence*, Vol. 46, No. 1, pp. 79-95, January, 2017.

[44] L. Wang, L. P. Li, An effective differential evolution with level comparison for constrained engineering design, *Structural and Multidisciplinary Optimization*, Vol. 41, No. 6, pp. 947-963, June, 2010.

[45] S. Shadravan, H. R. Naji, V. K. Bardsiri, The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, *Engineering Applications of Artificial Intelligence*, Vol. 80, pp. 20-34, April, 2019.

[46] Z. Zhang, S. Ding, W. Jia, A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems, *Engineering Applications of Artificial Intelligence*, Vol. 85, pp. 254-268, October, 2019.

[47] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Advances in engineering software*, Vol. 95, pp. 51-67, May, 2016.

[48] T. Lykouris, V. Syrgkanis, É. Tardos, Learning and efficiency in games with dynamic population, *In Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms Society for Industrial and Applied Mathematics*, Arlington, Virginia, USA, 2016, pp. 120-129.

[49] Z. G. Du, J. S. Pan, S. C. Chu, H. J. Luo, P. Hu, Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks, *IEEE Access*, Vol. 8, pp. 8583-8594, January, 2020.

[50] C. Sun, J. Ding, J. Zeng, Y. Jin, A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems, *Memetic Computing*, Vol. 10, No. 2, pp. 123-134, June, 2018.

[51] X. Kou, J. Feng, Matching ontologies through compact monarch butterfly algorithm, *Journal of Network Intelligence*, Vol. 5, No. 4, pp. 191-197, November, 2020.

Biographies

Shu-Chuan Chu received a Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She has been a Research Fellow and Associate Professor in the College of Science and Engineering of Flinders University, Australia since December 2009. Currently, she is a Research Fellow with a Ph.D. advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. She also serves as an editorial board member for Engineering Applications of Artificial Intelligence (EAAI), Journal of Internet Technology (JIT) and Research Reports on Computer Science (RRCS). Her research interests are mainly in Swarm Intelligence, Intelligent Computing, and Wireless Sensor Networks.

Qing Feng received her B.S. degree from Qingdao Binhai University, China, in 2020. She is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. Her recent research interests include swarm intelligence, clustering, and FPGA.

Jia Zhao received the B.S. degree in computer science and technology from Nanchang Institute of Aeronautical Technology, Nanchang, China, in 2004, and the M.E. degree in computer application technology from Nanchang Hangkong University, Nanchang, China, in 2011, and the Ph.D. degree in information and communication engineering from Hohai University, Nanjing, China, in 2020.He is currently a professor with the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. He is also the Director of Nanchang Key Laboratory of Big Data and Computational Intelligence. His research interests include big data analysis; artificial intelligence theory; deep learning and reinforcement learning.

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Professor of Shandong University of Science and Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section and Tainan Chapter Chair of IEEE Signal Processing Society. His current research interest includes the information hiding, artificial intelligence and wireless sensor networks.