


RESEARCH

Open Access



An improved wolf colony search algorithm based on mutual communication by a sensor perception of wireless networking

Chenghai Wu^{1*} , Kaiyu Qin¹, Penghui He² and Houbiao Li²

Abstract

Intelligence optimization algorithms have an important application in engineering calculation. Swarm intelligence optimization algorithms belong to the class of intelligent algorithms, which are derived from the simulation of natural biological evolution or foraging behaviours. But these algorithms have sometimes slow convergence, and for multi-peak problems, they are easy to fall into the local optimal solution in the later period of the algorithms. To solve these problems, in this paper, the advantages of these algorithms and the wolf colony search algorithm based on the strategy of the leader (LWCA) algorithm are used, to propose an improved wolf colony search algorithm with the strategy of the leader algorithm (ILWCA). ILWCA is based on mutual communication by the sensor perception of wireless networking, adding a global update strategy and a step acceleration network. In addition, by introducing the concept of individual density to depict the distribution density of the wolf, the problem of excessive input parameters of traditional wolf group algorithm is solved. Moreover, by adding the way of mutual migration, the algorithm increases the communication between wolves, and strengthens the overall performance of the optimization process. Finally, the experimental results show that the ILWCA algorithm achieves higher solution accuracy and better robustness compared with the accuracy and robustness of particle swarm optimization (PSO), gravitational search algorithm (GSA), swarm optimization algorithm (SOA), grey wolf optimizer (GWO), and genetic algorithms (GA).

Keywords: Intelligent algorithm, ILWCA algorithm, Mutual communication, Sensor network, Grey wolf optimizer

1 Introduction

Intelligent optimization techniques have become very popular over the past few decades. In engineering practice, “novel” algorithms or theories are often encountered. For example, simulated annealing, genetic algorithms, tabu search, and neural networks are representative of these innovations. These algorithms and theories have some common features, such as simulation of natural processes, which cast them as “intelligent algorithms”. They are very useful in solving some complex engineering problems.

Swarm intelligence optimization algorithm is a type of intelligent algorithm that is derived from the simulation of natural biological evolution or foraging behaviour. In recent years, intelligent algorithms have become very popular, including genetic algorithms (GA) [1, 2], particle

swarm algorithm (PSA) [3, 4], monkey search algorithm [5], and artificial bee colony algorithm (ABC) [6]. However, these algorithms sometimes encounter slow convergence, and for multi-peak problems, they are prone to fall into a local optimal solution in the later period of the algorithm.

On the basis of the hunting behaviour of the wolf group, the wolf swarm algorithm is proposed. The wolf swarm algorithm has both better robustness and global characteristic. The prey hunting behaviour of wolves consists of wandering, attack, and siege. There are now many algorithms related to the prey behaviour of wolves. For examples, the LWCA [7] incorporated the addition of a leader strategy to the original wolf pack algorithm (WPA) [8]. The GWPA [9, 10] approach added the direction dual chaos update strategy. The MWCA [11] method joined an interactive mode with an interactive movement feature for the walking of raiding wolves in the foraging process, and increased the exchanges to

* Correspondence: uestchwu@126.com

¹School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China

Full list of author information is available at the end of the article

enhance the overall algorithm. Additionally, MWCA [8] is aimed at multiple-peak problems, and MO-WCA [12] is used to solve a multi-objective problem of WCA. This paper primarily uses the double gauss function updating method in the small wolf pack algorithm (SWPA) [12]. The double gauss function update method is used to update the wolves, producing a better individual. Although the wolf swarm algorithm was proposed based on bionic wolves in 2007, it has been widely developed in recent years [13, 14]. Based on WPA, GWO [15, 16], and grey wolf optimization algorithm based on strengthening hierarchy of wolves (GWOSH) [17] divided the wolves into four grades, and different grades of wolves were capable of different ways to walk and run. In this paper, the advantages of these algorithms were extended to reduce the input of the parameters of the wolf swarm algorithm, join an interactive walking movement, change the manner of walking, improve the update strategy, and propose a new ILWCA algorithm. The experiment is performed with 22 test functions and compared with GWO, PSO, gravitational search algorithm (GSA) [18], swarm optimization algorithm (SOA) [19], and GA. Based on experimental results, the accuracy of the ILWCA algorithm is higher and shows better robustness.

The biological background of wolves is closely related to information science, as wolves in nature have a strict hierarchy. A pack of wolves usually has a leader wolf, an explorer wolf, and a fierce wolf, as well as other wolves apart from this hierarchy. Wolves, as the top group of predators at the peak of the natural food chain, have typical overall characteristics in their systems such as relevance and integrity [20, 21]. The wolf that finds the best game will howl and attract other wolves to round up the prey. They capture the prey, ensure their overall survival, safeguard the integrity of the pack, and exhibit a clear division of labour for each individual in the wolf pack. According to the call behaviour of the wolves and their mutual communication [22], the leader wolf, the explorer wolf, and the fierce wolf are able to communicate with each other in real time. While they work independently but interact with each other as a wolf pack, they must also work and hunt together so that they can accomplish what one wolf alone cannot do. With various kinds of behaviour, such as their visual observation, collaborative predatory detection, sense of smell, psychological induction, and body language, an intelligent sensor network of real-time communication with each other is established. Their communication methods are concise and efficient [23], and the wolf pack has strong abilities for seeking, coordinating, and controlling. These habits and characteristics of wolves are beneficial for a search algorithm based on the behaviour of wolves.

The rest of the paper is organised as follows: Section 2 outlines the proposed ILWCA algorithm. The results

and discussion of benchmark functions that are used comprise Section 3 and Section 4, respectively. Finally, Section 5 concludes the work and suggests some directions for future studies.

2 Problem formulation and methods

In this section, based on step acceleration technology and the global mutual communication update strategy, a new wolf group algorithm called ILWCA is presented that combines the advantages of other algorithms. ILWCA also adds an interactive way of walking, which can improve the wolf's communication and optimization process. The algorithm also introduces the double gauss update strategy, which makes the algorithm produce better individuals while optimising the entire group of wolves.

The algorithm flow of ILWCA algorithm is as follows in Fig. 1.

The pseudo code of the ILWCA algorithm is presented in Fig. 2.

2.1 Initialization of the wolf group

The location of wolves is randomly initialized in the search space.

$$X_i = (x_i^1, \dots, x_i^D) \quad (1)$$

$$x_i^d = x_{\min} + \mathbf{rand} \times (x_{\max} - x_{\min}) \quad (2)$$

where x_{\max} and x_{\min} are the maximum and minimum range of activity, respectively. Rand is a random number.

2.2 Wolves moved to three leading wolves (raid)

First, the fitness of each wolf was calculated, and the wolf with the optimal fitness was marked as α . Then, the wolf with the second best fitness was selected as β , and the third best wolf was selected as γ . The values of the α, β , and γ wolves are recorded as $f_\alpha, f_\beta, f_\gamma$ respectively. The sum of their adaptive values is f .

$$f = f_\alpha + f_\beta + f_\gamma \quad (3)$$

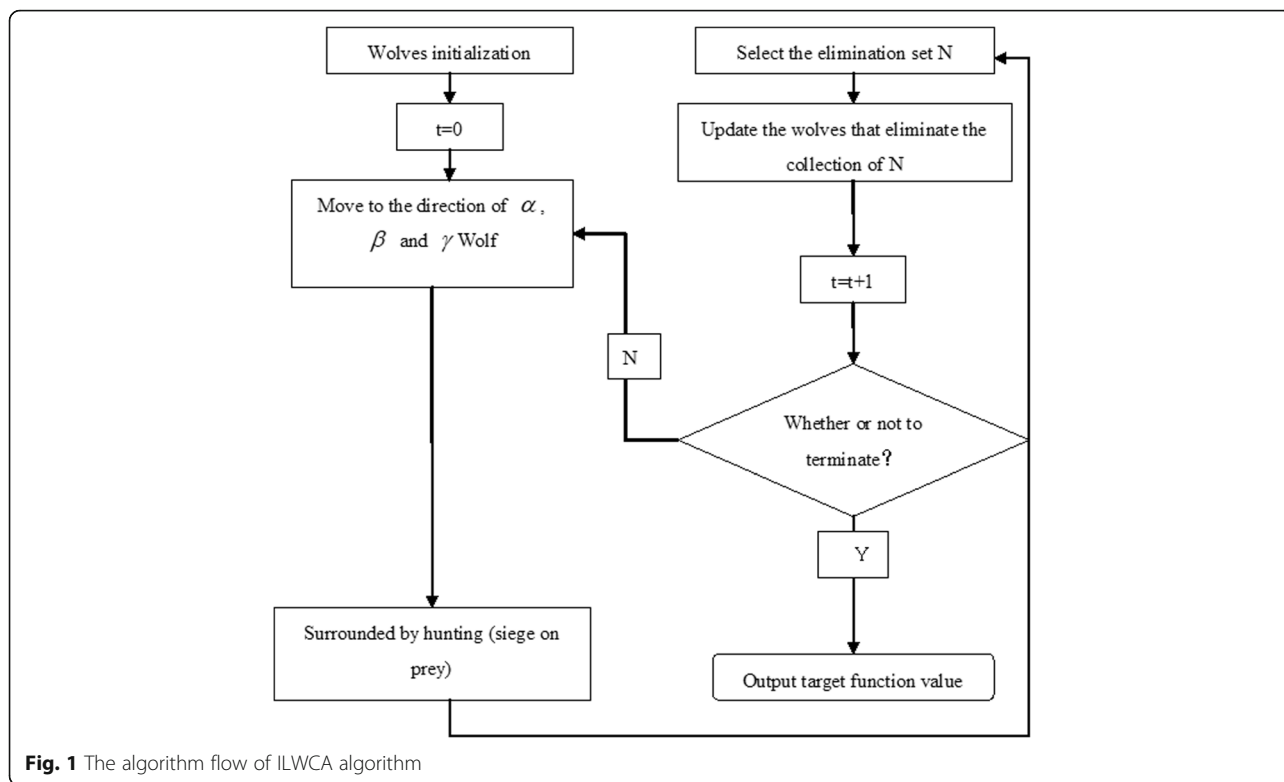
In this paper, the Euclidean distance is used to depict the distance between two wolves. The distance between the X_i wolf and the α wolf is d_α .

$$d_\alpha = \left(\sum_{i=1}^D (x_i^d - x_\alpha^d)^2 \right)^{\frac{1}{2}} \quad (4)$$

where x_i^d is the d dimension of the X_i wolf, and x_α^d is the d dimension of the α wolf. The distance between the X_i wolf and the β wolf is d_β .

$$d_\beta = \left(\sum_{i=1}^D (x_i^d - x_\beta^d)^2 \right)^{\frac{1}{2}} \quad (5)$$

Similarly, the distance between the X_i wolf and the γ wolf is d_γ .



$$d_\gamma = \left(\sum_{i=1}^D (x_i^d - x_\gamma^d)^2 \right)^{\frac{1}{2}} \tag{6}$$

The sum of $d_\alpha, d_\beta, d_\gamma$ is defined as d .

$$d = d_\alpha + d_\beta + d_\gamma \tag{7}$$

$$z_i^d = x_i^d + \left(1 - \frac{f_\alpha}{f^* e^{(t/t \max)^2}} \right) \left(1 - \frac{d_\alpha}{d} \right) (x_\alpha^d - x_i^d) + \left(1 - \frac{f_\beta}{f} \right) \left(1 - \frac{d_\beta}{d} \right) (x_\beta^d - x_i^d) + \left(1 - \frac{f_\gamma}{f} \right) \left(1 - \frac{d_\gamma}{d} \right) (x_\gamma^d - x_i^d) \tag{8}$$

Therefore, the mode of raid may be built as follows:

```

Initialize, m
Initialize the grey wolf population, (i=1,2,...,n)
While (t<tmax)
Sort by appropriate value.
Screen, and Wolf
Move to the direction of, and Wolf by equation (8)
Interacting with each other by equation (9).
Siege to the direction of the Wolf by equation (12).
Update the wolf pack by equation (16).
end
return [Xbest, pbest] % The position of leading the wolf and its moderate value
  
```

Fig. 2 The pseudo code of the ILWCA algorithm

The farther the α wolf is from the X_i wolf, the smaller the effect. The bigger the α wolf's fitness value is, the greater the influence on the X_i wolf. The β and γ wolves are the same. The introduction of $e^{(t/t_{\max})^2}$ is used to increase the influence of the α wolf with the increase in the number of iterations and to accelerate the speed and convergence trend of the algorithm. If the location of the wolf X_i moves to $Z_i((z_i^1, z_i^2, \dots, z_i^D))$, the fitness of Z_i is better than the current location. The position of the X_i wolf is updated to Z_i . On the other hand, the X_i wolf's position does not move. The adaptation values of the X_i wolf after updating were compared with the fitness values of the α , β , and γ wolves. The best three wolves were selected as the new α , β , and γ wolves to complete the update of α , β , and γ wolves.

The interaction between individuals in intelligent algorithms helps to enhance the optimization ability of the algorithm and the ability to jump out of the local extremum. To increase the interactivity of the wolves, the following search method is applied.

$$w_i^d = x_i^d + \phi_{id}(x_\alpha^d - x_i^d) + \phi_{id}(x_j^d - x_k^d) \tag{9}$$

where ϕ_{id} is a random number in $[0, 1]$, ϕ_{id} is a random number in $[-1, 1]$, and k, i , and j are three mutually unequal numbers. ϕ_{id} is the d dimension coordinate of the X_k wolf. x_α^d is the d dimensional coordinate of the best solution. The first part of formula (9) enhanced the local optimization ability of the algorithm, and the second part enhanced the global search ability of the algorithm. It is a good balance of the exploitation and exploration ability of walking behaviour, which not only embodies the influence of the wolf on the entire wolf group but also reflects the ability of each individual to learn from each other, improving the diversity of the population. If the raid interaction position is $W_i(w_i^1, w_i^2, \dots, w_i^D)$, the position of the X_i wolf is updated to W_i ; otherwise, the X_i wolf's position remains the same.

When moving to α , β , and γ , the locations of Z_i and W_i extend beyond the boundary; therefore, cross-border processing should be performed.

$$z_i^d = \begin{cases} x_{\max}^d, & \text{if } z_i^d > x_{\max}^d \\ x_{\min}^d, & \text{if } z_i^d < x_{\min}^d \end{cases}, \tag{10}$$

$$w_i^d = \begin{cases} x_{\max}^d, & \text{if } w_i^d > x_{\max}^d \\ x_{\min}^d, & \text{if } w_i^d < x_{\min}^d \end{cases}. \tag{11}$$

where x_{\max}^d and x_{\min}^d represent the maximum and minimum value of the d dimension of the search space, respectively.

2.3 Surrounded by hunting (siege)

After moving in the direction of the α , β , and γ wolves and given that the location of the α wolf stands for the location of the present optimal prey, the entire wolf pack is moved in the direction of the α wolf. If the position of a wolf in the siege is better than that of the α wolf, the wolf will take the place of the α wolf. The location of the α wolf is updated, and the remaining wolves who are not besieging continue to move to the new α wolf. The siege does not end until all wolves carry out the siege. After the X_i wolves carry out a siege, the d dimension will be defined as v_i^d :

$$v_i^d = x_i^d + r^*(x_\alpha^d - x_i^d) \tag{12}$$

where r is a random number in $[0, 2]$ so that the X_i wolf can attack the back area of prey. In addition, the trans-boundary treatment of $V_i(v_i^1, v_i^2, \dots, v_i^D)$ is considered. If the prey at V_i is better than the current location, the location of the X_i wolf is updated to V_i ; otherwise, the X_i wolf's position is kept unchanged.

2.4 Updated decision

In the early stage, wolves tend to search globally, so the wolves should explore more unknown areas. If the wolves in the region are denser in this area, each wolf in the region is well-adapted, and it can be inclined to select some of the poorly adapted wolves among the pack to search in other regions. Because wolves in the area are searching, they can avoid falling into a local optimum. In the later stage, the tendency is to search carefully to find better prey. Therefore, most of the wolves can be concentrated in a certain area for a detailed search. Every time the wolves are updated, the poorly adapted individuals tend to be eliminated.

Next, we introduce the method of the authors in [13] to depict the individual density of the X_i wolf.

The truncated distance is defined by dc , which is a parameter for determining whether a wolf is in the vicinity of another wolf. If the distance between the two wolves is smaller than the truncation distance, the distance between the two wolves is very close.

λ is a truncated distance parameter that determines the truncation distance, and it is determined by the user in advance. After sorting all the distances between all of the wolves, the cut-off distance $\lambda\%$ is defined.

The local density of the X_i wolves is denoted by ρ_i :

$$\rho_i = \sum_{j \in P/\{i\}} e^{-\left(\frac{d_{ij}}{dc}\right)^2} \tag{13}$$

If the wolf's distance from the X_i wolf is less than dc , the greater the number is, the greater the local density of the α wolf is. d_{ij} represents the Euclidean distance

between the X_i wolf and the X_j wolf, and $P = \{1, 2, \dots, n\}$, where n is the total number of wolves.

By sorting the local density of the wolves, 2 m wolves with the largest local density are selected. They are ordered by density from large to small in set A . Then, 2 m wolves are selected to adapt to the worst wolves, and they are put into set B according to their fitness values from small to large. The intersection of sets A and B is denoted by C :

$$C = A \cap B \tag{14}$$

$$C = \{c : c_i \in B, c_i \in A, i = 1, \dots, l\} \tag{15}$$

The wolf in set C has a large local density and poor fitness. The elimination of wolves can be selected at the early stage of collection C .

Two major elimination methods are adopted. Let N store the eliminated individuals.

$$N = \begin{cases} N_1, & \text{rand} > e^{(\frac{t}{r_{\max}})^2} \\ B[1 : m], & \text{other} \end{cases} \tag{16}$$

Random numbers are generated by the rand function for the interval $[0, 1]$. The probability of the establishment of $\text{rand} > e^{(\frac{t}{r_{\max}})^2}$ in the early stage is larger. With the increase in iteration steps, the probability will be increasingly smaller. The algorithm tends to search globally, and later, the algorithm tends to search locally.

Discussion of the cases:

$$N_1 = \begin{cases} C[1 : m], & l \geq m \\ C + (A/C)[1 : m - l], & \text{rand} > e^{(\frac{t}{r_{\max}})^2} \text{ and } l < m \\ C + (B/C)[1 : m - l], & \text{other} \end{cases} \tag{17}$$

N_1 is a collection of eliminated wolves. $C[1 : m]$ indicates that the former m elements are taken out in the collection C . $(A/C)[1 : m - l]$ means that after removing the intersection from set A to set B , we choose the first m elements in the remaining elements. $(B/C)[1 : m - l]$ is empathy. $\text{rand} > e^{(\frac{t}{r_{\max}})^2}$ has great probability in the early stage of the algorithm, and with the increase of

iteration steps, the probability will be smaller and smaller. This indicates that ILWCA algorithm tends to eliminate individuals with larger local density in the earlier stage. To facilitate the exploration of more unknown areas, in the later stage, ILWCA algorithm tends to eliminate individuals with poor fitness, which should increase local search speed, speed up convergence, and search for better prey.

According to the principle of survival of the fittest, we remove the wolves in set N . Then, m wolves are generated by the double Gauss function update method. Such population is not easy to fall into local optimum and makes the population diversity. The double Gauss function is constructed to make the location of W_i^* of new-born wolf X_i^* away from the location of the original wolf W_i , because the location of W_i does not have good population diversity. The d dimensional location of the new artificial wolf W_i^* is defined as X_i^{*z} :

$$X_i^{*z} = \delta_i^d \times N_2 + |1 - \delta_i^d| \times X_i^z \tag{18}$$

where δ_i^d is the d dimension updating coefficient of the artificial wolf W_i , which is a randomly generated 0 or 1. It is used to determine whether the location of the dimension is updated. N_2 is the updating coefficient of the position of the artificial wolf W_i , obtained by the double Gauss function update method. The difference between the newly generated wolves in the group of wolves is the result of the impact of many small, independent, and random factors from the perspective of probability and statistics. According to the central limit theorem, a new generation of an artificial wolf's difference in position can be considered to have normal distribution characteristics. At the same time, to promote the new generation of an artificial wolf who fled from the original artificial wolf's position, more artificial wolves should be made that are far away from the original position; therefore, a new generation of artificial wolves is more likely to be started. To accomplish this aim, the new artificial wolf is generated by the following double Gauss function:

Table 1 Multimodal benchmark functions

Function (Fn*)	Dim	Range	f_{\min}
$F_4(X) = \sum_{i=1}^n (x_i - 1/i)^2$	30	[- 10, 10]	0
$F_{10}(X) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[- 500, 50]	- 12,569.4
$F_{12}(X) = (\pi/n) \{ 10 \sin^2(\pi y_1) + \sum_{j=1}^{D-1} (y_j - 1)^2 [1 + \sin^2(\pi y_{j+1})] + (y_n - 1)^2 \} + \sum_{i=1}^D u(x_i, 10, 100, 4)$	30	[- 50, 50]	0
$F_{16}(X) = \frac{1}{n} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	100	[- 5, 5]	- 78.3323

*Fn—said $F_n(X)$

Table 2 Results of benchmark functions

F		ILWCA	GWO	PSO	GSA	SOA	GA
F4	Average	2.3098e-07	1.6809e-01	5.9244e+00	2.5968e+03	8.7209e+00	3.7533e-01
	Best	2.0231e-08	6.5745e-02	2.7091e+00	2.3267e+03	3.7315e+00	1.5339e-02
	Worst	1.0011e-06	8.6775e-01	2.2001e+01	2.7596e+03	1.7043e+01	1.2087e+00
F10	Average	-8.3013e+03	-5.9874e+03	-5.6100e+03	-2.7100e+03	-2.6426e+03	-2.0209e+03
	Best	-1.0709e+04	-7.1086e+03	-8.4041e+03	-4.0804e+03	-3.4663e+03	-3.3395e+03
	Worst	-7.0223e+03	-4.7540e+03	-2.8426e+03	-1.7474e+03	-2.0591e+03	-1.3446e+03
F12	Average	1.0207e+00	4.1775e-02	3.4556e-03	4.5266e+08	1.8131e+00	2.2822e+00
	Best	3.8368e-21	6.7666e-03	3.2972e-11	1.2343e+08	1.3187e-01	3.4683e-01
	Worst	9.8116e+00	1.2092e-01	1.0366e-01	6.5390e+08	3.8369e+00	3.9012e+00
F16	Average	-6.7145e+01	-4.6695e+01	-6.5902e+01	-3.0162e+01	-5.1201e+01	-2.0334e+01
	Best	-7.0698e+01	-5.3634e+01	-6.9157e+01	-3.7542e+01	-6.4863e+01	-2.7265e+01
	Worst	-6.4195e+01	-3.9875e+01	-6.3337e+01	-2.6260e+01	-3.8281e+01	-1.5495e+01

$$N_2 = (x_{\min}^d + \sigma_1 \times e) \times \text{heaviside}(r_1 - r) + (x_{\max}^d + \sigma_2 \times e) \times \text{heaviside}(r - r_1) \quad (19)$$

$$\text{heaviside}(x) = \begin{cases} 0, & x < 0 \\ 0.5, & x = 0 \\ 1, & x > 0 \end{cases} \quad (21)$$

$$r = \frac{x_i^d - x_{\min}^d}{x_{\max}^d - x_{\min}^d} \quad (20)$$

The variables σ_1 and σ_2 are the mean square variance of the double Gauss function. The d dimensional space is the position of the hunting grounds with the upper limit of x_{\max}^d and a lower limit of x_{\min}^d . The mean double Gauss function is used to promote the new generation of an artificial wolf who fled the original region. The variable e is used to comply with the standard normal distribution of the random number; r_1 belongs to a random number between [0,1], which is used to promote the new generation of artificial wolves who tend to stay away from the original position to escape. The unit step function used is $\text{heaviside}(x)$, defined as follows:

If the d dimensional position W_i^* of the X_i^* new generation artificial wolf W_i^* is beyond the scope of hunting, the new position of the wolf is regenerated with the double Gauss function method. The update process is not stopped until the new artificial wolf's d dimensional position is in the hunting space.

3 Experimental design

To verify the effectiveness of the algorithm, 22 benchmark functions [24] were selected for testing. The algorithm is compared with GWO, PSO, GSA, SOA, and GA. Some benchmark functions are shown in Table 1.

The initialization scale of ILWCA and GWO, GSA, SOA, PSO, and GA is 30, and the maximum iterations are set to 800. The number of wolves in which other

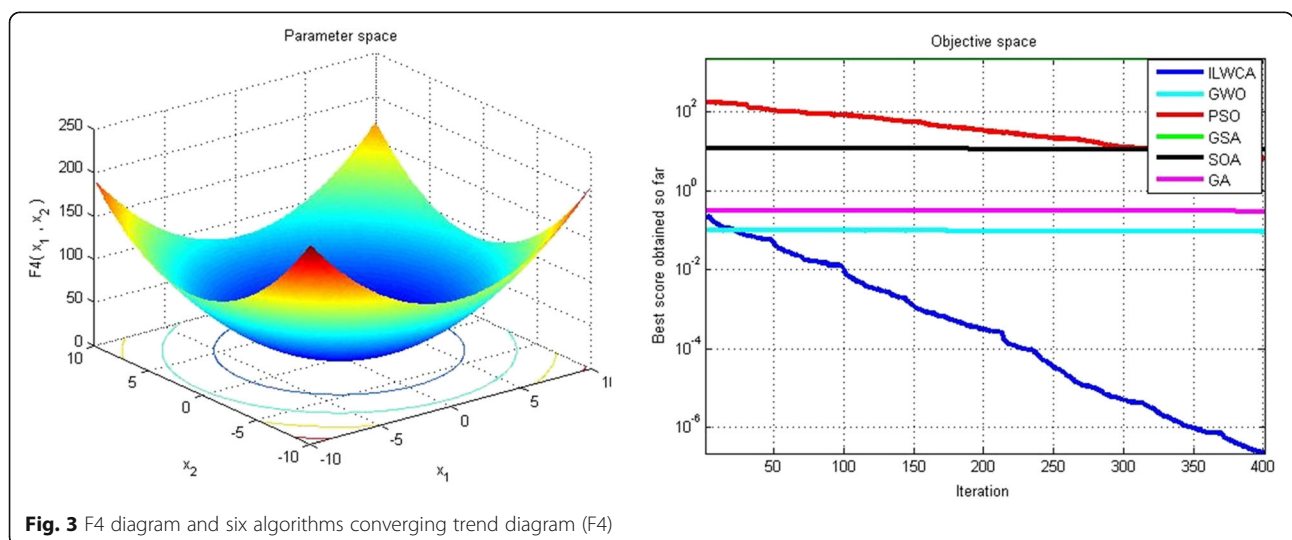
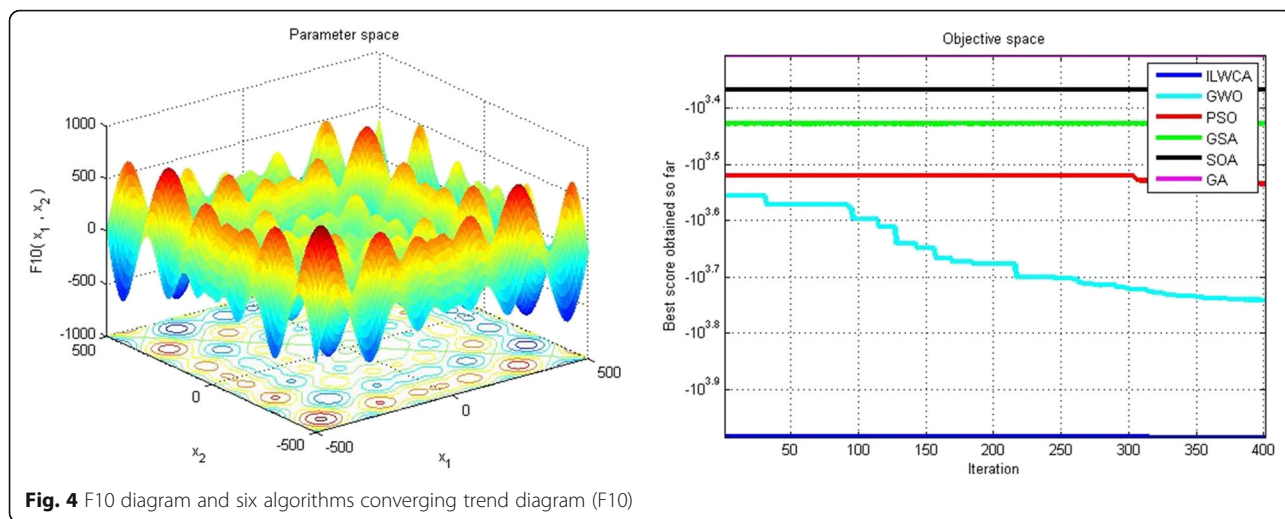


Fig. 3 F4 diagram and six algorithms converging trend diagram (F4)



parameters are eliminated is $m = 5$. The truncated distance parameter is $\lambda = 3$. The updated double Gauss function selects the Gauss function with a mean value of 0 and a variance of 1. For GWO, the parameter $\alpha = 2$. For GSA, the initial gravitational constant $G = 100$, $\alpha = 20$, and the speed range is $[-1, 1]$. For SOA, the maximum membership degree is $U_{max} = 0.9500$, and the minimum membership degree is $U_{min} = 0.0111$. For PSO, the inertia weight is $C_1 = C_2 = 2$. For GA, the cross probability is 0.75, and the mutation probability is 0.05.

4 Results and discussion

4.1 Results of benchmark functions

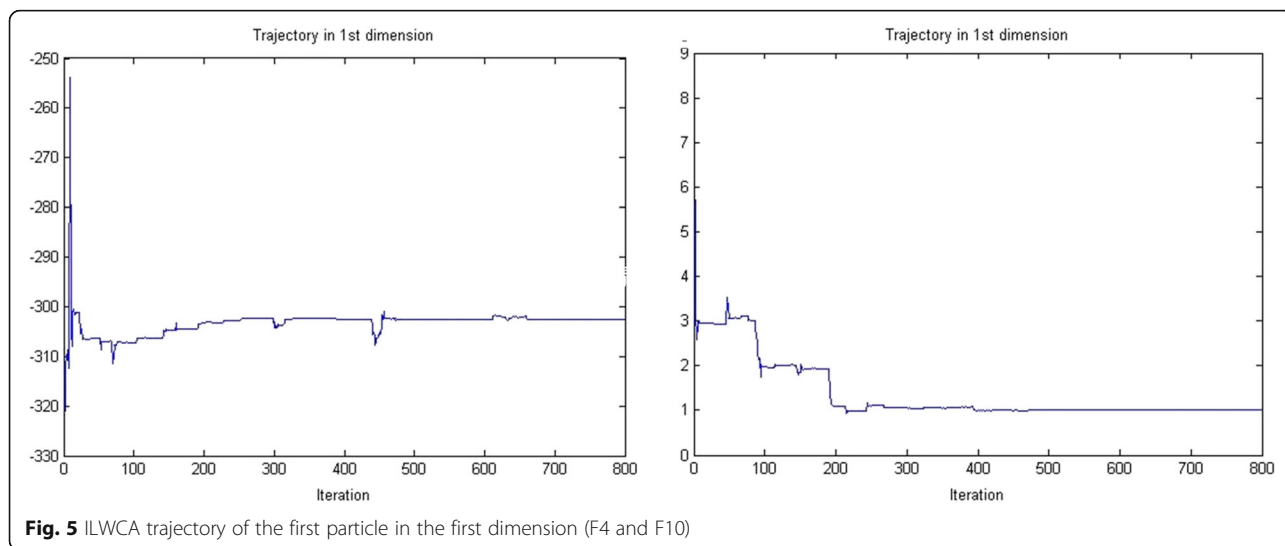
In the actual experiment, 22 independent functions were tested. To illustrate the convenience of the results, the results of four unimodal functions of F4, F10, F12, and F16 are shown below. The average value, the best value, and the worst value are shown in Table 2. The

converging trend diagram of some functions is drawn for six algorithms. Additionally, the first dimension variable of the α wolf position in the ILWCA algorithm is used to change the graph of the optimization process (see Figs. 3, 4, 5, 6, 7, and 8).

The above six algorithms were analysed in terms of the following two aspects: the best value and the average value.

For function F4, the best value found by the ILWCA algorithm is the best among the algorithms, reaching $1E-08$. The optimal values for GWO and GA are only $1E+00$, and the other three algorithms achieve only up to $1E+00$. For the functions F10, F12, and F16, the ILWCA is superior to the other algorithms in finding the optimal values of these functions.

Comparison of mean value, as for functions F4, F10, and F16, the average value that ILWCA finds has good results, which are better than the other algorithms. For



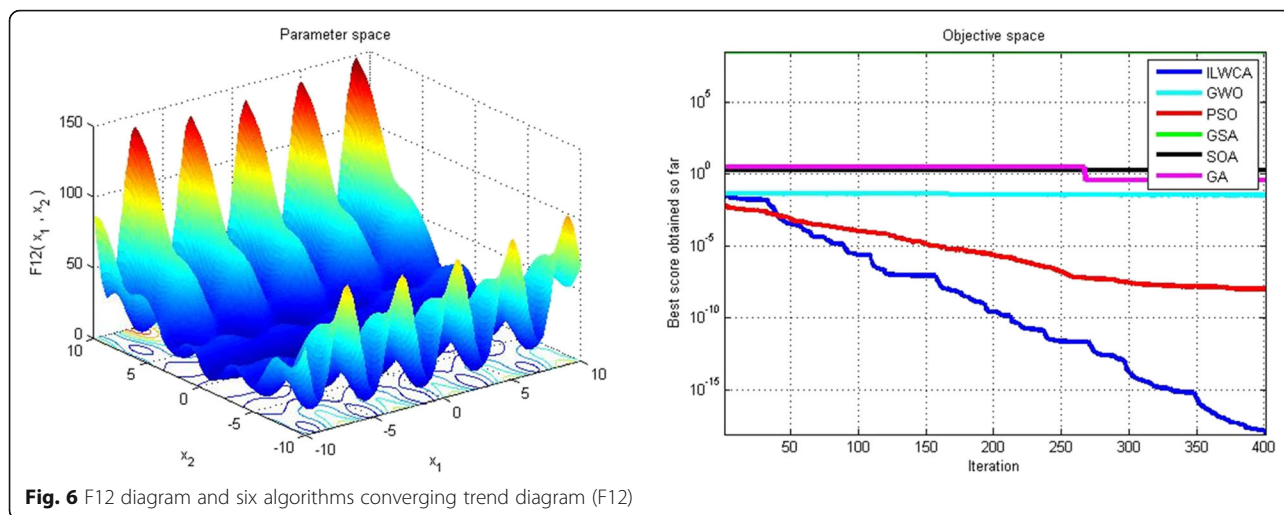


Fig. 6 F12 diagram and six algorithms converging trend diagram (F12)

function F12, the average value searched by GWO and PSO is the best, and ILWCA is second. For function F16, the average value obtained by ILWCA and PSO is $1E-30$, and the exact solution is 0.

To make a clearer comparison of these algorithms, some functions within these twenty-two test functions were chosen. The images of the functions were observed, and the algorithm converges in the contrast graph and the track of the first-dimension variables in the search process of the ILWCA algorithm, see Figs. 3, 4, 5, 6, 7, and 8.

4.2 Convergence behaviour analysis

In order to get a clearer understanding of the gaps in these algorithms and to make a clearer comparison of these algorithms, we choose some functions in these 22 test functions. The images of the functions are observed. We also draw six algorithms converging trend diagram of some functions, as well as the convergent contrast diagram (see Figs. 3, 4, 5, 6, 7, and 8), and the first

dimension variable of α wolf position in ILWCA algorithm is used to change the graph of optimization process (see Fig. 5 and Fig. 8).

From the convergence diagram of ILWCA algorithm, we can know that as the number of iterations increases, the value of functions presents monotonically increasing trend and approaches the global optimal solution. In order to further observe the convergence behaviour of the algorithm, we also drew the search history track of the first search agent in the first search dimension. From the first dimension of the first dimension of the above seven functions, we can learn that the trend of the first dimension of the first dimension of the wolf is very large in the early period of the search, but the search trajectory tends to be stable in the later period of the iteration. This behaviour guarantees that a SI algorithm will eventually converge to the point in search space. This gives a good indication of the stability of the ILWCA algorithm. For Fig. 8 F12, to the later stage, the change trend graph is also unstable, because the function is a process

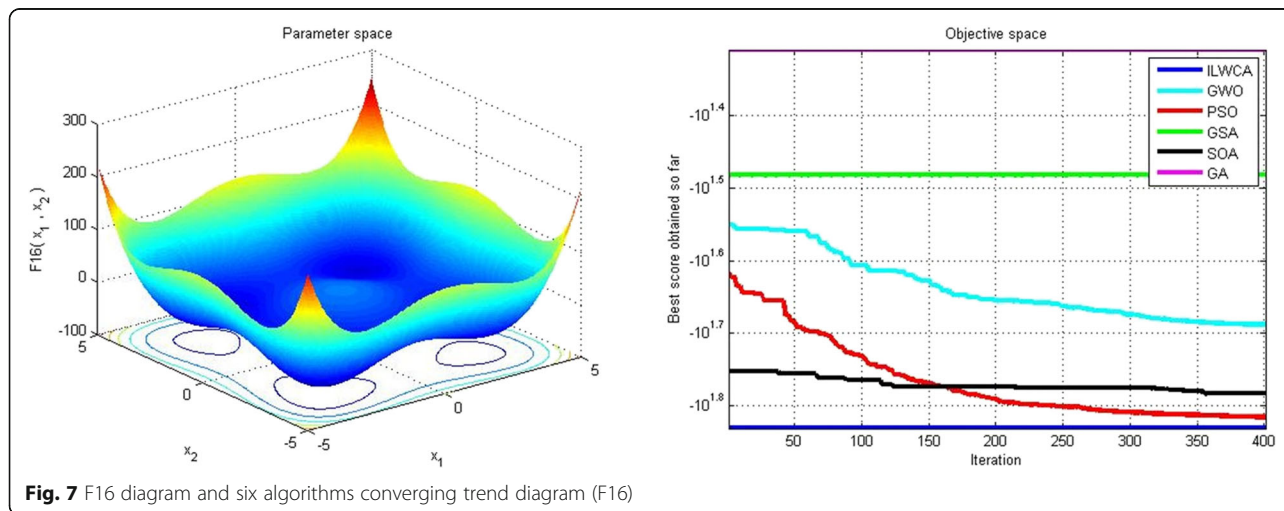


Fig. 7 F16 diagram and six algorithms converging trend diagram (F16)

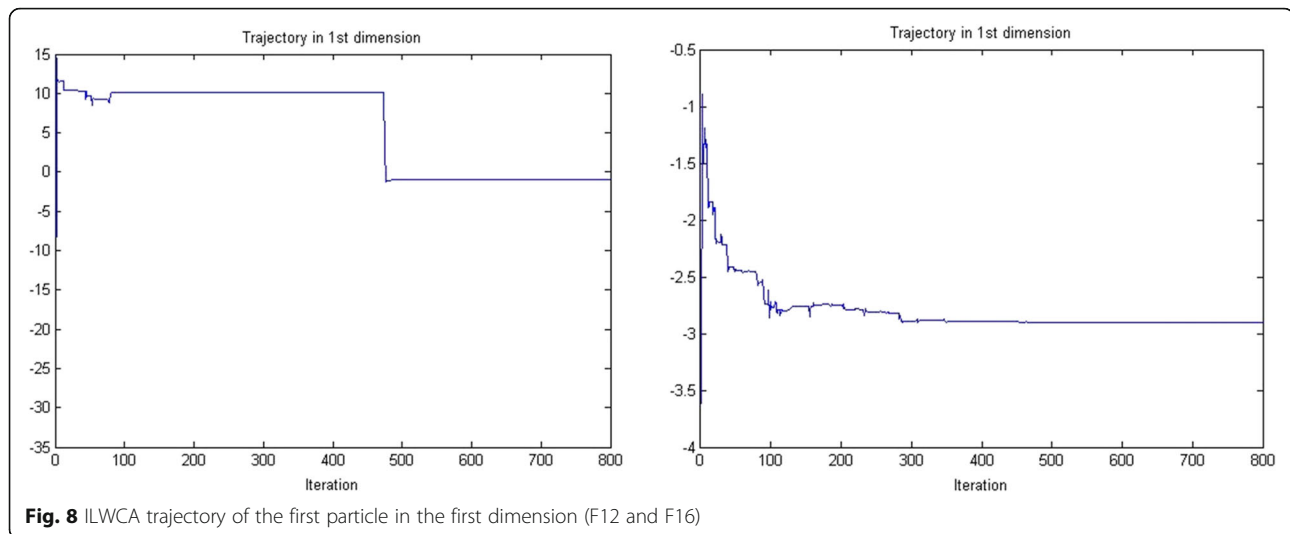


Fig. 8 ILWCA trajectory of the first particle in the first dimension (F12 and F16)

overlap, and the point of the function image is stereoscopic in the vicinity of the global best advantage. The optimal prey searched by ILWCA algorithm is generally better than other algorithms.

5 Conclusions

This article presents a swarm intelligence algorithm based on the predation behaviour of wolves. The survival rule of the wolf group through the survival of the fittest was evaluated to find the global optimal solution. IWCA was combined with the characteristics of other algorithms, and added an interactive walk to increase the overall algorithm. To reduce the probability of leading wolves to bring the entire group of wolves to the local solutions, an improved manner of wolves' wandering was adopted.

In addition, this paper introduced a second excellent moderate value and the third best fitness value of a wolf. Together with the leader wolf, they led the entire group of wolves to find the best global value. The other wolves were affected by the three wolves, which were closely related to the wolves' adaptation value and the distance from the wolves. Therefore, the greater the adaptation value is, the closer the distance is, and the greater the effect on the wolf is.

To make the wolves explore more unknown areas, if the wolves in a region were denser in a particular area, each wolf in the region had a better fitness value and was inclined to select some of the poorly adapted wolves to go to other regions and search (elimination). Since there were already wolves in the area, falling into a local optimum could be avoided. In the later period of the algorithm, the search tended to be more careful to facilitate the search for better prey. Therefore, most of the wolves were concentrated in a certain area for a detailed search. Every time the wolves were updated, the poorly adapted individuals tended

to be eliminated. In addition, the double Gauss function update method was introduced to generate better individuals.

Acknowledgements

The authors would like to thank Dr. Seyedali Mirjalili and other authors for providing us some codes during the research and preparation of the manuscript.

Funding

The authors are partially supported by NSFC (61702083), The Doctoral Programme Foundation of State Education Ministry Grant (20130185110023) and Fundamental Research Funds for the Central Universities (ZYGX2016J131, ZYGX2016J138).

Authors' contributions

CW and PH contributed to the conception and algorithm design of the study. PH and KQ contributed to the acquisition of simulation. CW and HL contributed to the analysis of the simulation data and approved the final manuscript. All authors read and approved the final manuscript.

Authors' information

Chenghai Wu received an M.S. degree and PhD as a student from the University of Electronic Science and Technology of China, in 2010 and 2012, respectively. He is a Senior Engineer in Beijing Special Vehicle Institute, Beijing, China. His research interests lie in Multi-Agent System, Coordinated Control, Vehicle Formation, Modelling and Simulation, Computational Intelligence, Communication and Electronics, Wireless Sensor Networks, Software Engineering and Internet of Things. (e-mail: uestchwu@126.com). Kaiyu Qin received the M.S. and PhD degrees from University of Electronic Science and Technology of China, in 1994 and 1999, respectively. He is a professor in the School of Astronautics and Aeronautics, University of Electronic Science and Technology of China, Chengdu, Sichuan, China. His research interests lie in the Simulation, Testing, Verification and Evaluation of Complex Electronic Systems, Space Electronic Systems, Microwave Modules and Component Technologies. (e-mail: kyqin@uestc.edu.cn). Penghui He is currently pursuing the M.S. degree with the School of Mathematical Science, University of Electronic Science and Technology of China, Chengdu, China. His research interests lie in artificial intelligence and image restoration. (e-mail: m18380132550@163.com). Houbiao Li received the M.S. and PhD degrees in computational mathematics from the School of Mathematical Science, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2004 and 2007, respectively. He is currently an Associate Professor with the School of Mathematical Sciences, UESTC. He has authored over 40 scientific papers. His research interests lie in numerical algebra and parallel computing. (e-mail: hlb0189@uestc.edu.cn).

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China. ²School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China.

Received: 2 May 2018 Accepted: 30 May 2018

Published online: 15 June 2018

References

- PM Pardalos, HE Romeyn, Recent developments and trends in global optimization. *J. Comput. Appl. Math.* **124**(1–2), 209–228 (2000). [https://doi.org/10.1016/S0377-0427\(00\)00425-8](https://doi.org/10.1016/S0377-0427(00)00425-8)
- X Yan, D Peng, J Ma, Analog circuit diagnosis based on wolf pack algorithm radical basis function network. *Comput. Eng. Appl.* **53**(19), 152–156 (2017). <https://doi.org/10.3778/j.issn.1002-8331.1607-0182>
- Hollandj, Outline for a logical theory of adaptive systems. *J. Assoc. Comput. Mach.* **9**(3), 297–314 (1962)
- S Mirjalili, SM Mirjalili, A Lewis, Grey wolf optimizer[J]. *Adv. Eng. Softw.* **69**(3), 46–61 (2014). <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- LIU Chang-an, YAN Xiao—hu, LIU Chun-yang, et al., The wolf colony algorithm and its application. *Chin. J. Electron.* **20**(2), 212–216 (2011)
- D Karaboga, B Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **39**(3), 459–471 (2007). <https://doi.org/10.1007/s10898-007-9149-x>
- Q Zhou, Y Zhou, Wolf colony search algorithm based on leader strategy. *Appl. Res. Comput.* **30**(9), 2629–2632 (2013). <https://doi.org/10.3969/j.issn.1001-3695.2013.09.018>
- J bin, Z Jin, Multi-objective sink nodes coverage algorithm based on quantum wolf pack evolution[J]. *J. Electron. Inf.* **39**(5), 1–7 (2017). <https://doi.org/10.11999/JEIT160693>
- X Hui, G Qing, Z Yu, et al., An improved wolf pack algorithm. *Control and Decision* **32**(7), 1164–1172 (2017). <https://doi.org/10.13195/j.kzyjc.2016.0690>
- X Zhang, TU Qiang, Q Kang, et al., Grey wolf optimization algorithm with double-hunting modes and its application to multi-threshold image segmentation. *J. Shanxi Univ.* **39**(3):378-85 (2016). [https://doi.org/10.13451/j.cnki.shanxi.univ\(nat.sci\).2016.03.006](https://doi.org/10.13451/j.cnki.shanxi.univ(nat.sci).2016.03.006)
- Y Liu, J Yong, W Song, et al., Track planning for unmanned aerial vehicles based on wolf pack algorithm. *J. Syst. Simul.* **27**(8), 1838–1843 (2015). <https://doi.org/10.16182/j.cnki.joss.2015.08.027>
- J Xue, W Ying, J Xiao, et al., A small wolf pack algorithm and its convergence analysis. *Control and Decision* **31**(12), 2131–2139 (2016). <https://doi.org/10.13195/j.kzyjc.2016.0690>
- H Zheng, Chenkeng, An incremental dynamic clustering method base on the representative points and the density peaks. *J. Zhejiang Univ. Technol.* **45**(4), 427–433 (2017). <https://doi.org/10.3969/j.issn.1006-4303.2017.04.014>
- T Wu, F Zhang, L Wu, New swarm intelligence algorithm-wolf pack algorithm. *Syst. Eng. Electron.* **35**(11), 2431–2438 (2013). <https://doi.org/10.3969/j.issn.1001-506X.2013.11.33>
- Z Qiang, Y Zhou, Wolf colony search algorithm based on leader strategy. *Appl. Res. Comput.* **30**(9), 2630–2632 (2013). <https://doi.org/10.3969/j.issn.1001-3695.2013.09.018>
- PC Pinto, TA Runkler, JM Sousa, in *Wasp swarm algorithm for dynamic MAX-SAT problems*, ed. by Adaptive and Natural Computing Algorithms. (Springer, 2007), pp. 350–357. https://doi.org/10.1007/978-3-540-71618-1_39
- X Zhang, T Qiang, K Qiang, et al., Grey wolf optimization algorithm based on strengthening hierarchy of wolves. *J. Date Acquis. Process.* **32**(5), 880–889 (2017). <https://doi.org/10.16337/j.1004-9037.2017.05.004>
- NM Sabri, M Puteh, MR Mahmood, A review of gravitational search algorithm. *Int. J. Adv. Soft Comput. Appl.* **5**(3), 1–39 (2013)
- KK Seo, Content-based image retrieval technique by combining swarm optimization algorithm and support vector machines[J]. *J. Comput. Theor. Nanosci.* **10**(8), 1693–1700 (2013)
- X. Lu and Y. Zhou, A novel global convergence algorithm: bee collecting pollen algorithm. *Advanced Intelligent Computing Theories and Applications*. ed. by With Aspects of Artificial Intelligence (Springer, 2008), pp. 518–525. DOI: https://doi.org/10.1007/978-3-540-85984-0_62
- AH Gandomi, AH Alavi, Krill herd: A new bio-inspired optimization algorithm[J]. *Commun. Nonlinear Sci. Numer. Simul.* **17**(12), 4831–4845 (2012)
- X-S Yang, *Test problems in optimization*. *arXiv preprint arXiv:1008.0549*, 2010. *al Review Letters*, vol. 116, no. 6, Article ID 061102 (2016). <https://doi.org/10.1002/9780470640425>
- J Huang, Z Liang, Q Zang, Dynamics and swing control of double-pendulum bridge cranes with distributed-mass beams. *Mech. Syst. Signal Process.* **54–55**, 357–366 (2015)
- X-S Yang, Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio-Inspired Comput.* **2**, 78–84 (2010). <https://doi.org/10.1504/IJBIC.2010.032124>

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com