



Dolphin swarm algorithm^{*}

Tian-qi WU[†], Min YAO, Jian-hua YANG

(School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

[†]E-mail: TainchWu@gmail.com

Received Sept. 4, 2015; Revision accepted Mar. 15, 2016; Crosschecked July 19, 2016

Abstract: By adopting the distributed problem-solving strategy, swarm intelligence algorithms have been successfully applied to many optimization problems that are difficult to deal with using traditional methods. At present, there are many well-implemented algorithms, such as particle swarm optimization, genetic algorithm, artificial bee colony algorithm, and ant colony optimization. These algorithms have already shown favorable performances. However, with the objects becoming increasingly complex, it is becoming gradually more difficult for these algorithms to meet human's demand in terms of accuracy and time. Designing a new algorithm to seek better solutions for optimization problems is becoming increasingly essential. Dolphins have many noteworthy biological characteristics and living habits such as echolocation, information exchanges, cooperation, and division of labor. Combining these biological characteristics and living habits with swarm intelligence and bringing them into optimization problems, we propose a brand new algorithm named the 'dolphin swarm algorithm' in this paper. We also provide the definitions of the algorithm and specific descriptions of the four pivotal phases in the algorithm, which are the search phase, call phase, reception phase, and predation phase. Ten benchmark functions with different properties are tested using the dolphin swarm algorithm, particle swarm optimization, genetic algorithm, and artificial bee colony algorithm. The convergence rates and benchmark function results of these four algorithms are compared to testify the effect of the dolphin swarm algorithm. The results show that in most cases, the dolphin swarm algorithm performs better. The dolphin swarm algorithm possesses some great features, such as first-slow-then-fast convergence, periodic convergence, local-optimum-free, and no specific demand on benchmark functions. Moreover, the dolphin swarm algorithm is particularly appropriate to optimization problems, with more calls of fitness functions and fewer individuals.

Key words: Swarm intelligence, Bio-inspired algorithm, Dolphin, Optimization

<http://dx.doi.org/10.1631/FITEE.1500287>

CLC number: TP391

1 Introduction

Swarm intelligence has become a hot topic in artificial intelligence and bionic calculation fields since its appearance in the 1980s. Bonabeau *et al.* (1999) defined swarm intelligence as "any attempt to design algorithms or distributed problem-solving strategies inspired by the collective behavior of social insect colonies and other animal societies". The main properties of swarm intelligence include decentralization, stigmergy, self-organization, positive feedback, negative feedback, fluctuation, and dif-

ferences (Garnier *et al.*, 2007). Many algorithms have been designed based on the definitions and properties of swarm intelligence. These algorithms can be divided into two categories: one is the bionic process algorithm, based on the genetic algorithm mainly imitating the process of population evolution; the other is the bionic behavior algorithm, which simulates the behavior model of different species searching for prey, and the main representatives are particle swarm optimization, artificial bee colony algorithm, ant colony optimization, and so on. These algorithms have been widely applied to the reconstruction of grids (Kennedy, 2010), telecom network routing (Ducattelle *et al.*, 2010), wireless sensor networks (Saleem *et al.*, 2011), cluster analysis (Cura, 2012), pattern recognition (Eberhart and Shi, 2001),

^{*} Project supported by the National Key Technology R&D Program of China (No. 2014BAD10B02)

ORCID: Tian-qi WU, <http://orcid.org/0000-0003-1547-3479>

© Zhejiang University and Springer-Verlag Berlin Heidelberg 2016

and other applications.

Currently, particle swarm optimization, genetic algorithm, artificial bee colony algorithm, and ant colony optimization are well implemented and have shown favorable performances. They use the behavior rules of individuals and the interactions between individuals to produce changes at the group level and achieve certain goals. The genetic algorithm simulates the mutations and exchanges of genes and finds the gene with the best environmental adaptation by the rule of 'survival of the fittest' (Whitley, 1994; Mitchell, 1998). Particle swarm optimization simulates the predatory behavior of bird swarms, which involves finding the prey by exchanging information and picking out the best among the search results, in addition to considering the information from other birds (Eberhart and Kennedy, 1995; Poli *et al.*, 2007; Kennedy, 2010). The artificial bee colony algorithm simulates the bees' honey-gathering behavior and finds better nectar sources through cooperation and division of labor (Karaboga, 2005; Karaboga and Basturk, 2007; Karaboga *et al.*, 2014). Ant colony optimization simulates the foraging behavior of ants, which involves finding the shortest path from the nest to the food using pheromones for information exchanges (Dorigo *et al.*, 1996; Dorigo and Birattari, 2010; Mohan and Baskaran, 2012).

Humans obtain favorable outcomes by imitating nature. The existing swarm intelligence algorithms can solve many practical problems. However, with the objects of optimization problems becoming increasingly complex and data becoming gradually enormous, traditional swarm intelligence algorithms cannot meet the practical demands in terms of time and accuracy. Because improvements in these traditional algorithms have gradually decreasing effect, designing a new algorithm to deal with optimization problems may be a better solution.

Apart from ants, bees, and birds, which have been successfully simulated, there are still many other creatures, such as bacteria and fireflies (Parpinelli and Lopes, 2011), worth our attention, and dolphin is one of them. The dolphin has many biological characteristics and living habits worth being learned from and simulated, such as echolocation, cooperation and division of labor, and information exchanges. Several phases, including the search phase, call phase, reception phase, and predation

phase, comprise the dolphin's predatory process, and these characteristics and habits help the dolphin achieve its goal during the predatory process. By simulating the biological characteristics and living habits shown in the dolphin's predatory process, we propose a new algorithm named the 'dolphin swarm algorithm'.

These characteristics and habits simulated in the dolphin swarm algorithm conform to the thoughts of swarm intelligence but differ from traditional swarm intelligence algorithms in their details. Take particle swarm optimization as an example. Although both methods use the concept of 'individual' and the interactions between multiple individuals to find the optimized solution, the way they get things done is totally different. Unlike particle swarm optimization (just moving forward to the solution), the dolphin swarm algorithm takes advantage of echolocation and adopts different strategies to obtain the solution more effectively, which may be a breakthrough. Furthermore, an algorithm based on dolphin's biological characteristics and living habits does not exist, which is why we make an attempt at simulating dolphins here.

2 Behavior of dolphin swarm

Dolphin is generally acknowledged as one of the smartest animals and it has a lot of interesting biological characteristics and living habits worth our attention.

1. Echolocation: Dolphin has good eyesight, but the good eyesight helps the dolphin little with predation in poor light conditions. Instead, the dolphin uses a special ability to search for prey, which is echolocation. Dolphin is one of the few creatures that are adept at using echolocation. It can make sounds and estimate the location, distance, and even the shape of the prey according to echo intensity. With the help of echo, the dolphin can have a better perception of the surrounding environment.

2. Cooperation and division of labor: In most cases, predatory behavior is not achieved by one dolphin alone but by the joint efforts of many dolphins through cooperation and division of labor. Facing large preys, the predatory behavior is unlikely to be achieved by only one dolphin. In such a case, the dolphin calls other dolphins for help with

predation. Moreover, there is a specific division of labor between the dolphins. For instance, the dolphins close to the prey are responsible for tracking the movements of the prey and the dolphins far away from the prey form a circle to surround the prey.

3. Information exchanges: Current studies show that dolphins have the ability to exchange information. They can express different ideas by using sounds at different frequencies and have their own language system. In the predatory process, especially under cooperation and division of labor, the ability of exchanging information is frequently used to call other dolphins and update the location of the prey. With the help of information exchanged, the dolphin can take better suited actions to make the predation more effective.

The whole process of dolphin's predation consists of three stages. In the first stage, each dolphin independently takes advantage of sounds to search for nearby preys and to evaluate the surrounding environment using echoes. In the second stage, dolphins exchange their information. The dolphins that find large preys call other dolphins for help. The dolphins that have received information move toward the prey and surround it along with other dolphins. In the last stage, the prey is surrounded by the dolphins and then what the dolphins need to do is to take turns to enjoy the food, which means that predation is accomplished.

3 Dolphin swarm algorithm

The dolphin swarm algorithm (DSA) is implemented mainly by simulating the biological characteristics and living habits shown in the dolphin's actual predatory process. The simulated predatory process is similar to the one described in Section 2. Main definitions and pivotal phases are introduced in Sections 3.1 and 3.2, respectively, and other definitions are introduced physically close to the part in which they are referred to in Section 3.2. The complete algorithm is expounded in Section 3.3.

3.1 Main definitions

3.1.1 Dolphin

Based on the idea of swarm intelligence, we need a certain number of dolphins to simulate the

biological characteristics and living habits shown in the dolphin's actual predatory process. In the optimization problem, each dolphin represents a feasible solution. Because the expressions of feasible solutions of different optimization problems vary, the dolphin in this study is defined as $\mathbf{Dol}_i = [x_1, x_2, \dots, x_D]^T$ ($i=1, 2, \dots, N$), for better understanding, namely, a feasible D -dimensional solution, where N is the number of dolphins and x_j ($j=1, 2, \dots, D$) is the component of each dimension to be optimized.

3.1.2 Individual optimal solution and neighborhood optimal solution

The individual optimal solution (denoted as \mathbf{L}) and neighborhood optimal solution (denoted as \mathbf{K}) are two variables associated with the dolphin. For each \mathbf{Dol}_i ($i=1, 2, \dots, N$), there are two corresponding variables \mathbf{L}_i ($i=1, 2, \dots, N$) and \mathbf{K}_i ($i=1, 2, \dots, N$), where \mathbf{L}_i represents the optimal solution that \mathbf{Dol}_i finds in a single time and \mathbf{K}_i the optimal solution of what \mathbf{Dol}_i finds by itself or gets from others.

3.1.3 Fitness

Fitness E is the basis for judging whether the solution is better. In DSA, E is calculated by a fitness function, and the closer it is to zero, the better it is. Because the fitness functions corresponding to different optimization problems are also different, for better understanding, the fitness function is represented as $\text{Fitness}(\mathbf{X})$ in this study, and specific examples of this fitness function can be found in Section 4.

3.1.4 Distance

In DSA, three types of distances are used in total. The first is the distance between \mathbf{Dol}_i and \mathbf{Dol}_j , named $DD_{i,j}$, which can be formulated as follows:

$$DD_{i,j} = \|\mathbf{Dol}_i - \mathbf{Dol}_j\|, \quad i, j=1, 2, \dots, N, \quad i \neq j. \quad (1)$$

The second is the distance between \mathbf{Dol}_i and \mathbf{K}_i , named DK_i , which can be formulated as follows:

$$DK_i = \|\mathbf{Dol}_i - \mathbf{K}_i\|, \quad i=1, 2, \dots, N. \quad (2)$$

The third is the distance between \mathbf{L}_i and \mathbf{K}_i , named DKL_i , which can be formulated as follows:

$$DKL_i = \|\mathbf{L}_i - \mathbf{K}_i\|, \quad i=1, 2, \dots, N. \quad (3)$$

DD_{ij} affects the information transfer between \mathbf{Dol}_i and \mathbf{Dol}_j ; DK_i and DKL_i thus influence the movement of \mathbf{Dol}_i in the predation phase.

3.2 Pivotal phases

DSA can be divided into six phases, including the initialization phase, search phase, call phase, reception phase, predation phase, and termination phase. This subsection is aimed to expound the four pivotal phases of DSA, i.e., search phase, call phase, reception phase, and predation phase.

3.2.1 Search phase

In the search phase, each dolphin searches its nearby area by making sounds toward M random directions. Similarly, the sound is defined as $V_i=[v_1, v_2, \dots, v_D]^T$ ($i=1, 2, \dots, M$) in this study, where M is the number of sounds and v_j ($j=1, 2, \dots, D$) is the component of each dimension, namely the direction attribute of the sound. In addition, sound satisfies $\|V_i\|=\text{speed}$ ($i=1, 2, \dots, M$), where ‘speed’ is a constant representing the speed attribute of sound. To prevent dolphins from getting stuck in the search phase, we set a maximum search time T_1 . Within the maximum search time T_1 , the sound V_j that \mathbf{Dol}_i ($i=1, 2, \dots, N$) makes at time t will search for a new solution X_{ijt} , which can be formulated as

$$X_{ijt} = \mathbf{Dol}_i + V_j t. \tag{4}$$

For the new solution X_{ijt} that \mathbf{Dol}_i gets, its fitness E_{ijt} is calculated as follows:

$$E_{ijt} = \text{Fitness}(X_{ijt}). \tag{5}$$

If

$$\begin{aligned} E_{iab} &= \min_{j=1,2,\dots,M;t=1,2,\dots,T_1} E_{ijt} \\ &= \min_{j=1,2,\dots,M;t=1,2,\dots,T_1} \text{Fitness}(X_{ijt}), \end{aligned} \tag{6}$$

then the individual optimal solution L_i of \mathbf{Dol}_i is determined as

$$L_i = X_{iab}. \tag{7}$$

If

$$\text{Fitness}(L_i) < \text{Fitness}(K_i), \tag{8}$$

then K_i is replaced by L_i ; otherwise, K_i does not change.

After all the \mathbf{Dol}_i ($i=1, 2, \dots, N$) update their L_i and K_i (if they can be updated), DSA enters the call phase.

3.2.2 Reception phase

Although the reception phase takes place after the call phase, it is necessary to expound on the reception phase first for better understanding. In DSA, the exchange process (including the call phase and the reception phase) is maintained by an $N \times N$ -order matrix named the ‘transmission time matrix’ (\mathbf{TS}), where TS_{ij} represents the remaining time for the sound to move from \mathbf{Dol}_j to \mathbf{Dol}_i .

When DSA enters the reception phase, all the terms TS_{ij} ($i=1, 2, \dots, N; j=1, 2, \dots, N$) in the transmission time matrix decrease by one to indicate that the sounds spread over one unit of time. Then DSA needs to check every term TS_{ij} in the matrix, and if

$$TS_{ij} = 0, \tag{9}$$

then it means that the sound sent from \mathbf{Dol}_j to \mathbf{Dol}_i can be received by \mathbf{Dol}_i . Then we need to replace TS_{ij} by a new time term named ‘maximum transmission time’ (T_2), to indicate that the corresponding sound has been received. Comparing K_i and K_j , if

$$\text{Fitness}(K_i) > \text{Fitness}(K_j), \tag{10}$$

then K_i is replaced by K_j ; otherwise, K_i does not change.

After all the terms in matrix \mathbf{TS} satisfying Eq. (9) are handled, DSA enters the predation phase.

3.2.3 Call phase

In the call phase, each dolphin makes sounds to inform other dolphins of its result in the search phase, including whether a better solution is found and the better solution’s location. The transmission time matrix \mathbf{TS} needs to be updated as follows:

For K_i, K_j , and TS_{ij} , if

$$\text{Fitness}(K_i) > \text{Fitness}(K_j), \tag{11}$$

and

$$TS_{i,j} > \left\lceil \frac{DD_{i,j}}{A \cdot \text{speed}} \right\rceil, \tag{12}$$

where ‘speed’ is a constant representing the speed attribute of sound, as mentioned in Section 3.2.1, and A is a constant representing the acceleration, which can make sounds spread faster in case the speed is rather slow. Then, $TS_{i,j}$ is updated as follows:

$$TS_{i,j} = \left\lceil \frac{DD_{i,j}}{A \cdot \text{speed}} \right\rceil. \quad (13)$$

After all the $TS_{i,j}$ ($i=1, 2, \dots, N; j=1, 2, \dots, N$) terms are updated (if they can be updated), DSA enters the reception phase.

3.2.4 Predation phase

In the predation phase, each dolphin needs to calculate the encircling radius R_2 , which determines the distance between the dolphin’s neighborhood optimal solution and its position after the predation phase according to the known information, and then gets a new position. For each dolphin, the known information contains its own position, its individual optimal solution L , its neighborhood optimal solution K , distance DK , and distance DKL ; the search radius R_1 , which represents the maximum range in the search phase, can be calculated as follows:

$$R_1 = T_1 \times \text{speed}. \quad (14)$$

Generally, the calculation of encircling radius R_2 and the update of the dolphin’s position should be discussed in three cases. In the following we take \mathbf{Dol}_i ($i=1, 2, \dots, N$) as an example to illustrate these three cases.

(a) For the currently known information of \mathbf{Dol}_i ($i=1, 2, \dots, N$), if

$$DK_i \leq R_1, \quad (15)$$

then it means that the neighborhood optimal solution K_i of \mathbf{Dol}_i is within the search range. For simplicity, in this case, DSA also regards the individual optimal solution L_i as K_i (Fig. 1).

In this case, the encircling radius R_2 can be calculated as follows:

$$R_2 = \left(1 - \frac{2}{e}\right)DK_i, \quad e > 2, \quad (16)$$

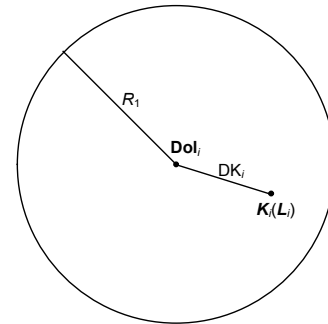


Fig. 1 Case (a) in the predation phase

where e is a constant named the ‘radius reduction coefficient’, which is greater than two and usually set as three or four. It can be easily seen that R_2 gradually converges to zero.

After obtaining the encircling radius R_2 , we can obtain \mathbf{Dol}_i ’s new position \mathbf{newDol}_i :

$$\mathbf{newDol}_i = K_i + \frac{\mathbf{Dol}_i - K_i}{DK_i} R_2. \quad (17)$$

Namely, \mathbf{Dol}_i moves toward K_i and stops at the position that is R_2 distance away from K_i (Fig. 2).

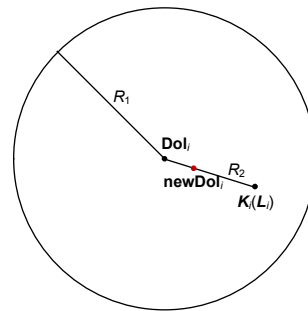


Fig. 2 The result of \mathbf{Dol}_i ’s movement in case (a)

(b) For the currently known information of \mathbf{Dol}_i ($i=1, 2, \dots, N$), if

$$DK_i > R_1, \quad (18)$$

and

$$DK_i \geq DKL_i, \quad (19)$$

then it means that \mathbf{Dol}_i updates K_i by receiving information from others and L_i is closer to \mathbf{Dol}_i than K_i is (Fig. 3).

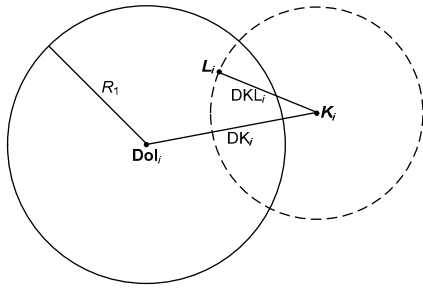


Fig. 3 Case (b) in the predation phase

In this case, the encircling radius R_2 can be calculated as follows:

$$R_2 = \left(1 - \frac{\frac{DK_i}{\text{Fitness}(K_i)} + \frac{DK_i - DKL_i}{\text{Fitness}(L_i)}}{e \cdot DK_i \frac{1}{\text{Fitness}(K_i)}} \right) DK_i, \quad e > 2. \quad (20)$$

After obtaining the encircling radius R_2 , we can obtain \mathbf{Dol}_i 's new position \mathbf{newDol}_i :

$$\mathbf{newDol}_i = K_i + \frac{\mathbf{Random}}{\|\mathbf{Random}\|} R_2. \quad (21)$$

Namely, \mathbf{Dol}_i moves to a random position that is R_2 distance away from K_i (Fig. 4).

(c) For the currently known information of \mathbf{Dol}_i ($i=1, 2, \dots, N$), if it satisfies inequality (18) and

$$DK_i < DKL_i, \quad (22)$$

then it means that \mathbf{Dol}_i updates K_i by receiving information from others and K_i is closer to \mathbf{Dol}_i than L_i is (Fig. 5).

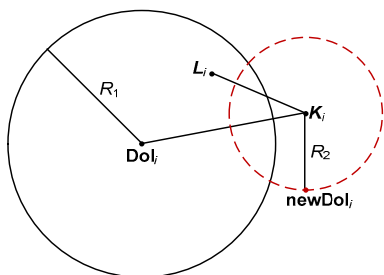


Fig. 4 The result of \mathbf{Dol}_i 's movement in case (b)

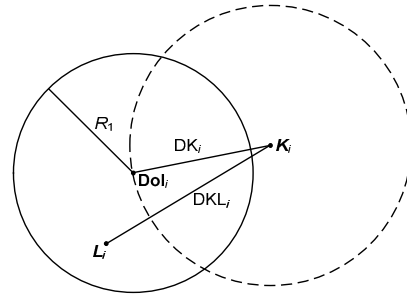


Fig. 5 Case (c) in the predation phase

In this case, the encircling radius R_2 can be calculated as

$$R_2 = \left(1 - \frac{\frac{DK_i}{\text{Fitness}(K_i)} - \frac{DKL_i - DK_i}{\text{Fitness}(L_i)}}{e \cdot DK_i \frac{1}{\text{Fitness}(K_i)}} \right) DK_i, \quad e > 2. \quad (23)$$

After obtaining the encircling radius R_2 , we can obtain \mathbf{Dol}_i 's new position \mathbf{newDol}_i by Eq. (21). Namely, \mathbf{Dol}_i moves to a random position that is R_2 distance away from K_i (Fig. 6).

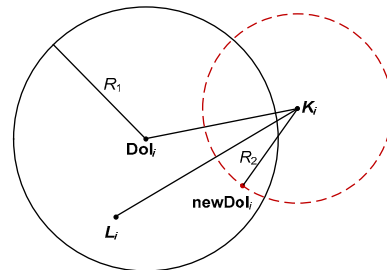


Fig. 6 The result of \mathbf{Dol}_i 's movement in case (c)

After \mathbf{Dol}_i moves to the position \mathbf{newDol}_i , comparing \mathbf{newDol}_i with K_i in terms of fitness, if

$$\text{Fitness}(\mathbf{newDol}_i) < \text{Fitness}(K_i), \quad (24)$$

then K_i is replaced by \mathbf{newDol}_i ; otherwise, K_i does not change.

After all the \mathbf{Dol}_i ($i=1, 2, \dots, N$) update their positions and K_i (if it can be updated), determine whether DSA meets the end condition. If the end condition is satisfied, DSA enters the termination

phase. Otherwise, DSA enters the search phase again.

3.3 Overall implementation

Algorithm 1 shows the overall DSA. It can be seen that the four pivotal phases are included, as well as the initialization phase and the termination phase. The initialization phase of DS contains the initialization of dolphins and the initialization of parameters. The initialization of dolphins is better when distributed randomly and evenly, and the initialization of parameters needs to be set according to different optimization problems. There are numerous conditions that can be used as the end condition, such as running out of given time, satisfying a certain precision, using up the calls of fitness functions, and so on. When the end condition is satisfied, the best one of \mathbf{K}_i ($i=1, 2, \dots, N$) will be the output.

Algorithm 1 Dolphin swarm algorithm

Step 1: initialize
 Randomly and evenly generate the initial dolphin swarm $\mathbf{Dol}=\{\mathbf{Dol}_1, \mathbf{Dol}_2, \dots, \mathbf{Dol}_N\}$ in the D -dimensional space.
 Calculate the fitness for each dolphin, and obtain $\text{Fit}_k=\{\text{Fit}_{k,1}, \text{Fit}_{k,2}, \dots, \text{Fit}_{k,N}\}$.
 Step 2: start loop
while the end condition is not satisfied **do**
 Step 2.1: search phase

$$E_{ijt}=\text{Fitness}(\mathbf{Dol}_i+\mathbf{V}_{jt})$$

$$\text{Fit}_{L_t}=\{\min\{E_{1jt}\}, \min\{E_{2jt}\}, \dots, \min\{E_{Njt}\}\}$$

$$\text{Fit}_{K,i}=\begin{cases} \text{Fit}_{L,i}, & \text{if } \text{Fit}_{L,i} < \text{Fit}_{K,i}, \\ \text{Fit}_{K,i}, & \text{otherwise.} \end{cases}$$

Step 2.2: call phase

$$\text{TS}_{i,j}=\begin{cases} \left\lceil \frac{\text{DD}_{i,j}}{A \cdot \text{speed}} \right\rceil, & \text{if } \text{Fit}_{K,j} < \text{Fit}_{K,i} \text{ and } \text{TS}_{i,j} > \left\lceil \frac{\text{DD}_{i,j}}{A \cdot \text{speed}} \right\rceil, \\ \text{TS}_{i,j}, & \text{otherwise.} \end{cases}$$

Step 2.3: reception phase
 $\text{TS}_{i,j}$ reduces one unit time

$$\text{Fit}_{K,i}=\begin{cases} \text{Fit}_{K,j}, & \text{if } \text{TS}_{i,j} = 0 \text{ and } \text{Fit}_{K,j} < \text{Fit}_{K,i}, \\ \text{Fit}_{K,i}, & \text{otherwise.} \end{cases}$$

Step 2.4: predation phase
 Calculate DK_i and DKL_i
if $\text{DK}_i \leq R_1$

$$R_2 = \left(1 - \frac{2}{e}\right) \text{DK}_i$$

else if $\text{DK}_i \geq \text{DKL}_i$

$$R_2 = \left(1 - \frac{\frac{\text{DK}_i}{\text{Fitness}(\mathbf{K}_i)} + \frac{\text{DK}_i - \text{DKL}_i}{\text{Fitness}(\mathbf{L}_i)}}{e \cdot \text{DK}_i \frac{1}{\text{Fitness}(\mathbf{K}_i)}}\right) \text{DK}_i$$

else

$$R_2 = \left(1 - \frac{\frac{\text{DK}_i}{\text{Fitness}(\mathbf{K}_i)} - \frac{\text{DKL}_i - \text{DK}_i}{\text{Fitness}(\mathbf{L}_i)}}{e \cdot \text{DK}_i \frac{1}{\text{Fitness}(\mathbf{K}_i)}}\right) \text{DK}_i$$

end if

\mathbf{Dol}_i gets a new position, calculates its fitness, and updates $\text{Fit}_{K,i}$

end while

output the best one of \mathbf{K}_i ($i=1, 2, \dots, N$)

4 Experiments

The experiments are divided into two parts. In the first part, DSA is compared with particle swarm optimization (PSO), the artificial bee colony algorithm (ABC), and the genetic algorithm (GA) in terms of the convergence curves of four benchmark functions with different extreme value distributions. In the second part, these four algorithms are compared in terms of the results of 10 benchmark functions.

In the experiments, 10 benchmark functions with different properties are used, which can be seen in Eqs. (25)–(34) (Yao *et al.*, 1999). Considering the distribution of extreme values, there are unimodal functions ($f_1(x)$ – $f_4(x)$), step functions ($f_5(x)$), functions with random numbers ($f_6(x)$), and multimodal functions ($f_7(x)$ – $f_{10}(x)$). Considering whether the variables are separable, $f_1(x)$, $f_5(x)$, $f_6(x)$, and $f_8(x)$ are separable and others are nonseparable. Apart from that, some benchmark functions have certain properties different from others, such as the discontinuity of $f_5(x)$ and the asymmetry of $f_8(x)$. These properties can help us testify the performances and characteristics of DSA such as the convergence rate and the possibility of falling into local optimum solutions, in various situations.

$$f_1(x) = \sum_{i=1}^n x_i^2, \tag{25}$$

$$f_2(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|, \tag{26}$$

$$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2, \tag{27}$$

$$f_4(x) = \max \{ |x_i| : 1 \leq i \leq n \}, \tag{28}$$

$$f_5(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2, \tag{29}$$

$$f_6(x) = \sum_{i=1}^n i x_i^4 + (\text{random} - \lfloor \text{random} \rfloor), \tag{30}$$

$$f_7(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], \tag{31}$$

$$f_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10], \tag{32}$$

$$f_9(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1, \tag{33}$$

$$f_{10}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] \right. \\ \left. + (x_n - 1) [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4),$$

$$\text{where } \sum_{i=1}^n u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < a. \end{cases} \tag{34}$$

Apart from the 10 benchmark functions, the initial distributions and the parameters that these four algorithms use in the experiments are also very important. To obtain better results, the initial distributions of individuals of these four algorithms are always the same in a single experiment. Moreover, the parameters of PSO, GA, and ABC chosen in the experiments have been widely used, and the parameters of DSA are decided after running some tests. All of them are shown in Table 1.

Table 1 Parameters used for different algorithms

Algorithm	Parameter(s)
DSA	$T_1=3, T_2=1000, \text{Speed}=1, A=5, M=3, e=4$
PSO	$V=10, C_1=2, C_2=2, W=0.8$
GA	$\text{SwiP}=0.6, \text{ChaP}=0.05$
ABC	$\text{MaxTimes}=10$

4.1 Comparison of convergence rate

We compare these four algorithms under the benchmark functions $f_1(x), f_5(x), f_6(x)$, and $f_7(x)$, respectively, with 10 dimensions (each dimension is in the range $[-100, 100]$), 10 individuals, and 10000 calls of benchmark functions. We plot the results of the former 50 loops that are typical and easy to observe and obtain four curves (Figs. 7–10).

Figs. 7–10 show that although $f_1(x), f_5(x), f_6(x)$, and $f_7(x)$ represent the unimodal function, step function, function with random numbers, and multimodal function, respectively, having completely different distributions of extreme values, the convergence of these four algorithms does not change much because of the different distributions of the extreme value. In the initial loops, ABC has the fastest convergence and DSA has the slowest convergence. With the increase of the number of loops, the other three algorithms' convergence slows down, but DSA retains a relatively high convergence rate. Furthermore, DSA becomes the fastest one when the number of loops reaches 25. In addition, we can see that the convergence curve of DSA does not decline linearly as the curve of PSO does, and that it has a periodic convergence, similar to GA.

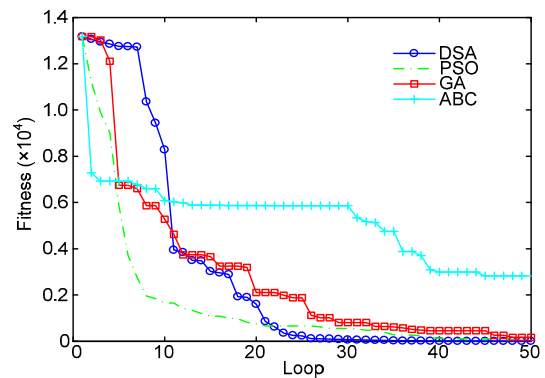


Fig. 7 The former 50 loops' convergence curves under a unimodal function $f_1(x)$

Through the above analysis, we can understand that DSA possesses some great features such as first-slow-then-fast convergence and periodic convergence. The reason that DSA possesses these features is that the information exchanges cause delay. We can learn from Eq. (13) that the delay is proportional to the distance between the dolphins. So, in the first few loops, the convergence is greatly influenced

by the delay, but the influence becomes smaller with the decrease in the distance between dolphins. The advantage is that each dolphin has more time for searching its own surrounding area, which enlarges the whole dolphin swarm's search area at the group level and avoids prematurity in DSA.

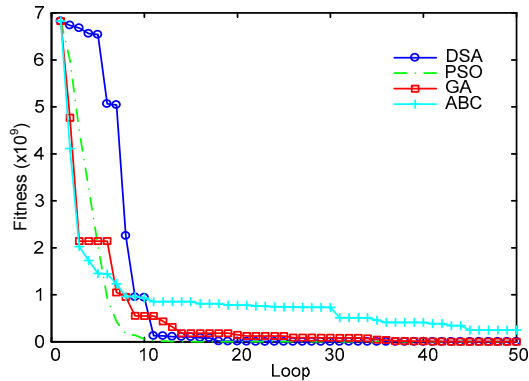


Fig. 8 The former 50 loops' convergence curves under a step function $f_5(x)$

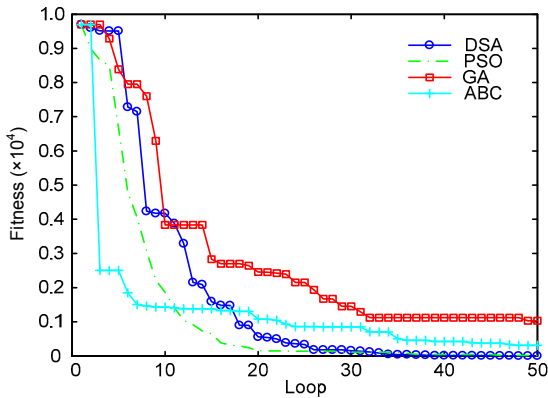


Fig. 9 The former 50 loops' convergence curves under a random number function $f_6(x)$

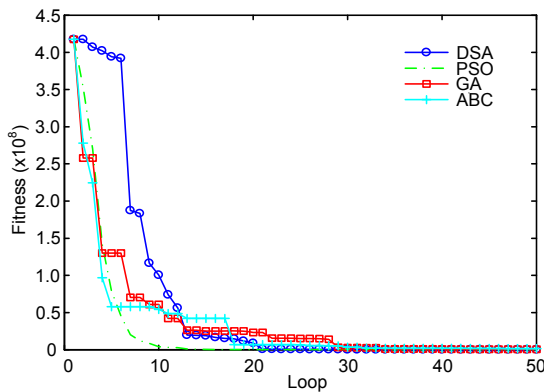


Fig. 10 The former 50 loops' convergence curves under a multimodal function $f_7(x)$

4.2 Comparison of results

DSA, PSO, GA, and ABC are compared with each other under four conditions: (1) 10 dimensions, 10 individuals, and 10000 calls of benchmark functions; (2) 30 dimensions, 10 individuals, and 10000 calls of benchmark functions; (3) 30 dimensions, 10 individuals, and 20000 calls of benchmark functions; (4) 30 dimensions, 20 individuals, and 20000 calls of benchmark functions (to be clear, each dimension ranges from -100 to 100). We use 10 benchmark functions with different properties to test these four algorithms and compare the results to find out the performances of these four algorithms in terms of different dimensions, different numbers of calls of benchmark functions, and different numbers of individuals. To obtain a better comparison and to reduce the influence of accidental factors, all the experimental results are obtained by carrying out each experiment 20 times. Furthermore, for each benchmark function, the algorithm that obtains the best result is tested with the other three algorithms using Wilcoxon's tests at the 5% significance level to assess whether it is significantly different from others.

Considering five significant figures, the results of 10 dimensions, 10 individuals, and 10000 calls of benchmark functions are shown in Table 2.

Considering five significant figures, the results of 30 dimensions, 10 individuals, and 10000 calls of benchmark functions are shown in Table 3.

Considering five significant figures, the results of 30 dimensions, 10 individuals, and 20000 calls of benchmark functions are shown in Table 4.

Considering five significant figures, the results of 30 dimensions, 20 individuals, and 20000 calls of benchmark functions are shown in Table 5.

The Wilcoxon test results of 10 dimensions, 10 individuals, and 10000 calls of benchmark functions are shown in Table 6.

The Wilcoxon test results of 30 dimensions, 10 individuals, and 10000 calls of benchmark functions are shown in Table 7.

The Wilcoxon test results of 30 dimensions, 10 individuals, and 20000 calls of benchmark functions are shown in Table 8.

The Wilcoxon test results of 30 dimensions, 20 individuals, and 20000 calls of benchmark functions are shown in Table 9.

Table 2 Results of 10 dimensions, 10 individuals, and 10 000 calls of benchmark functions*

f	DSA		PSO		GA		ABC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	4.0952E-02	1.9310E-02	1.0381E-01	6.7210E-02	2.5144E+00	1.4113E+00	8.7918E+00	5.8950E+00
2	3.6584E+02	3.7017E+01	5.2116E+01	4.8908E+01	3.3395E+00	7.5360E+00	3.1984E+01	6.3306E+00
3	1.8570E-01	8.8500E-02	1.6225E+00	6.2251E-01	3.0973E+02	5.7932E+02	2.9351E+01	1.1479E+01
4	2.2849E-01	5.5701E-02	1.1778E+00	3.1622E-01	3.4555E+00	1.1239E+00	2.3269E+00	8.5382E-01
5	5.5000E-01	5.0133E+02	9.5000E-01	7.5239E+02	3.1000E+00	1.9903E+03	2.6550E+01	4.8312E+03
6	1.2387E-01	4.2163E-01	2.9170E-01	1.6633E+00	4.8454E+00	1.3540E+00	1.0538E+01	2.0277E+01
7	1.2126E+01	4.3159E-02	1.4582E+02	1.4755E-01	2.3748E+02	1.0488E+01	4.2692E+02	1.1533E+02
8	4.5203E+01	1.5839E+01	3.9601E+01	1.2466E+01	3.1065E+01	1.0484E+01	1.1078E+02	1.5082E+01
9	3.0474E-01	2.8693E-02	1.8998E-01	7.7729E-02	1.5882E-01	4.3140E-02	3.6338E-01	1.7930E-01
10	1.9193E-02	1.2286E-02	1.1459E+00	2.2834E-01	6.4376E-01	2.1027E-01	5.9111E+00	2.8407E+00

* 10 dimensions, 10 individuals, and 10000 calls of benchmark functions. DSA: dolphin swarm algorithm; PSO: particle swarm optimization; GA: genetic algorithm; ABC: artificial bee colony algorithm. The best means and standard deviations are shown in bold font

Table 3 Results of 30 dimensions, 10 individuals, and 10 000 calls of benchmark functions*

f	DSA		PSO		GA		ABC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	1.5366E+00	4.1730E-01	2.4810E+01	8.4769E+00	1.1025E+02	1.9096E+01	2.3064E+02	1.1955E+02
2	3.5572E+07	1.8406E+08	3.0532E+02	6.3785E+06	2.8050E+01	4.7533E+00	1.1133E+06	4.4385E+11
3	4.8027E+03	3.0112E+03	1.8541E+03	2.2175E+03	8.3556E+03	5.5925E+03	8.9346E+02	2.7768E+02
4	4.7295E+01	3.2349E+00	1.6671E+01	5.6519E+00	1.7271E+01	2.5167E+00	8.9888E+00	1.6007E+00
5	1.6650E+01	1.6492E+04	6.9450E+01	4.3500E+04	1.0045E+02	1.1690E+05	4.6680E+02	8.9046E+05
6	7.7853E+01	7.6855E+00	7.2054E+02	3.1462E+01	1.0932E+04	2.4360E+01	1.3858E+05	1.6839E+02
7	2.6139E+03	2.8632E+02	6.5296E+04	3.8438E+03	2.0330E+05	1.9170E+04	8.2267E+05	7.1539E+04
8	5.7357E+02	1.2665E+01	4.6590E+02	8.5866E+01	3.2698E+02	5.5421E+01	7.6486E+02	1.3112E+02
9	1.3722E-01	5.4700E-02	7.1891E-01	1.8520E-01	1.0199E+00	2.7714E-02	1.0621E+00	2.4348E-02
10	3.6159E+01	1.3720E+01	6.8454E+02	3.0980E+03	2.4708E+03	1.7005E+02	3.2510E+04	6.5621E+04

* 30 dimensions, 10 individuals, and 10000 calls of benchmark functions. DSA: dolphin swarm algorithm; PSO: particle swarm optimization; GA: genetic algorithm; ABC: artificial bee colony algorithm. The best means and standard deviations are shown in bold font

Table 4 Results of 30 dimensions, 10 individuals, and 20 000 calls of benchmark functions*

f	DSA		PSO		GA		ABC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	4.5687E-01	1.3113E-01	6.0207E+00	2.6094E+00	2.4860E+01	4.3055E+00	1.4821E+02	7.3404E+01
2	1.4030E+06	1.9511E+07	9.6458E+01	1.4728E+03	1.3919E+01	1.7036E+00	1.7950E+04	8.8785E+08
3	7.2629E+02	1.3672E+02	9.8590E+02	1.7856E+02	8.1918E+03	1.2433E+03	7.4988E+02	1.9090E+024
4	3.2425E+01	5.1714E+00	1.3214E+01	3.6323E+00	1.1264E+01	1.3484E+00	7.5195E+00	1.0374E+00
5	9.6500E+00	4.7348E+03	4.8950E+01	5.3335E+03	2.3900E+01	1.3369E+04	4.3845E+02	1.7307E+05
6	1.0345E+00	3.5901E+00	3.5383E+02	2.4369E+01	1.1233E+03	8.2030E+00	8.2524E+04	1.0039E+02
7	5.7867E+02	2.8560E-01	5.7293E+03	1.2105E+02	1.5000E+04	1.0281E+03	2.8673E+05	3.7745E+04
8	3.8178E+02	1.9239E+02	4.0938E+02	1.2907E+02	1.4909E+02	2.8508E+01	6.6761E+02	1.5228E+02
9	4.9872E-02	9.0204E-03	3.2722E-01	9.9407E-02	6.0646E-01	1.0127E-01	1.0172E+00	3.5509E-02
10	2.5223E-01	2.6650E+02	4.7023E+01	3.6029E+02	4.1694E+00	1.2468E+02	4.3353E+03	8.4872E+03

* 30 dimensions, 10 individuals, and 20000 calls of benchmark functions. DSA: dolphin swarm algorithm; PSO: particle swarm optimization; GA: genetic algorithm; ABC: artificial bee colony algorithm. The best means and standard deviations are shown in bold font

Table 5 Results of 30 dimensions, 20 individuals, and 20 000 calls of benchmark functions*

<i>f</i>	DSA		PSO		GA		ABC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	2.5703E-01	1.1975E-01	2.7147E+00	1.4511E+00	4.4210E+01	1.3022E+01	6.5128E+01	2.5017E+01
2	1.0323E+06	3.3844E+04	4.9666E+03	3.4153E+01	1.5371E+01	1.9777E+00	2.2891E+03	2.7662E+01
3	1.4992E+03	5.1954E+02	2.8314E+02	6.8338E+01	8.5693E+03	2.7337E+03	2.4964E+02	7.5860E+01
4	3.8435E+01	1.1171E+01	9.1320E+00	2.4495E+00	1.3460E+01	1.5461E+00	4.6010E+00	9.7510E-01
5	8.8000E+00	1.2198E+03	2.2050E+01	1.4610E+03	3.6100E+01	2.0406E+04	1.5335E+02	5.7125E+04
6	3.1105E+00	4.2111E+00	1.6347E+02	6.3736E+00	4.5588E+03	7.8571E+00	7.9554E+03	4.3773E+01
7	6.7230E+02	2.0771E+00	2.4310E+03	8.0269E+01	5.0200E+04	6.0619E+03	6.8598E+04	7.7401E+03
8	3.3455E+02	7.5910E+01	2.9593E+02	9.5788E+01	1.9154E+02	2.7627E+01	4.5543E+02	4.7401E+01
9	4.9769E-02	1.0960E-02	2.0259E-01	7.0340E-02	8.8745E-01	9.5604E-02	9.4599E-01	5.0370E-02
10	6.2604E+00	3.1977E+00	3.0539E+01	1.9782E+01	2.5368E+01	3.7161E+02	3.3203E+01	7.7212E+00

* 30 dimensions, 20 individuals, and 20000 calls of benchmark functions. DSA: dolphin swarm algorithm; PSO: particle swarm optimization; GA: genetic algorithm; ABC: artificial bee colony algorithm. The best means and standard deviations are shown in bold font

Table 6 Results of Wilcoxon’s tests of the four algorithms*

<i>f</i>	DSA	PSO	GA	ABC
1	–	0	1	1
2	1	0	–	1
3	–	1	1	1
4	–	1	1	1
5	–	1	1	1
6	–	0	1	1
7	–	1	1	1
8	0	0	–	1
9	0	0	–	1
10	–	1	1	1

* 10 dimensions, 10 individuals, and 10000 calls of benchmark functions

Table 8 Results of Wilcoxon’s tests of the four algorithms*

<i>f</i>	DSA	PSO	GA	ABC
1	–	1	1	1
2	1	1	–	1
3	–	1	1	1
4	1	1	1	–
5	–	1	1	1
6	–	1	1	1
7	–	1	1	1
8	0	0	–	1
9	–	1	1	1
10	–	1	0	1

* 30 dimensions, 10 individuals, and 20000 calls of benchmark functions

Table 7 Results of Wilcoxon’s tests of the four algorithms*

<i>f</i>	DSA	PSO	GA	ABC
1	–	1	1	1
2	1	1	–	1
3	1	1	1	–
4	1	1	1	–
5	–	1	1	1
6	–	1	1	1
7	–	1	1	1
8	0	0	–	1
9	–	1	1	1
10	–	1	1	1

* 30 dimensions, 10 individuals, and 10000 calls of benchmark functions

Table 9 Results of Wilcoxon’s tests of the four algorithms*

<i>f</i>	DSA	PSO	GA	ABC
1	–	1	1	1
2	1	1	–	1
3	1	1	1	–
4	1	1	1	–
5	–	1	1	1
6	–	1	1	1
7	–	1	1	1
8	0	0	–	1
9	–	1	1	1
10	–	1	1	1

* 30 dimensions, 20 individuals, and 20000 calls of benchmark functions

In Tables 2–5, the best means and standard deviations of the four algorithms are shown in bold font. In Tables 6–9, the algorithm that obtains the best result is represented as ‘–’, and ‘1’ means that the algorithm is significantly different from the best one.

On the contrary, ‘0’ means that the algorithm is not significantly different from the best one.

Tables 2–9 show that DSA performs better in most cases and has advantages in terms of magnitude over the other three algorithms in more than half of

the cases. In the benchmark function $f_8(x)$, DSA (with a smaller standard deviation) and PSO are not significantly different from GA, although GA leads to the best results in the benchmark function $f_8(x)$. In some benchmark functions, taking benchmark function $f_2(x)$ as an example, DSA obtains unsatisfactory results, while GA obtains the best results. We can learn from Eq. (26) that there is a continuous multiplication that influences the benchmark function $f_2(x)$ the most. It means that the shape of the benchmark function $f_2(x)$ is too precipitous, which is unsuitable for DSA, because dolphins move around the neighborhood optimal solution in DSA. If the shape of the fitness function around the neighborhood optimal solution is too precipitous, then the predation phase is more likely to move to a random position. On the contrary, GA just exchanges the genes and selects the best one, so the shape of the fitness function has little influence on GA's performance. In other words, the performance of DSA varies when the shape of the fitness function is different, but in most cases, DSA performs well.

Comparing Tables 2 and 3, we can figure out that without changing other parameters but the dimension, DSA performs better in low-dimensional unimodal functions and high-dimensional multimodal functions, and that the performance with low dimension is generally better than that with high dimension. Comparison of Tables 3 and 4 shows that DSA leads to better results with more calls of the benchmark function, which corresponds to the conclusions shown in Section 4.1. Because of the delay caused by information exchanges, DSA converges faster than the other three algorithms due to the increase in the number of calls of the benchmark function. Comparison of Tables 4 and 5 shows that DSA performs worse than before when the number of individuals increases, which also conforms to the conclusions shown in Section 4.1. Because the increase in the individual number means a decrease of the total loop number, there are not enough loops for DSA to converge faster than the other three algorithms since DSA has first-slow-then-fast convergence. Based on the above analysis, we can see that better results can be achieved by increasing the number of calls of the benchmark function appropriately when the solutions to the optimization problem become complex, while increasing the number of individuals does not work.

5 Conclusions

In this paper we first introduce some of the noteworthy biological characteristics and living habits of a dolphin and use the dolphin swarm's predatory process to explain how these biological characteristics and living habits work. Then, we propose a brand new algorithm, the dolphin swarm algorithm (DSA), based on the idea of swarm intelligence by simulating these biological characteristics and living habits and give specific descriptions of the four pivotal phases in the algorithm, which are search phase, call phase, reception phase, and predation phase.

In the experiment section, DSA is compared with PSO, GA, and ABC in terms of the convergence rate and the benchmark function results. After comparing these four algorithms' convergence curves under four benchmark functions with different extreme value distributions, we conclude that DSA possesses some great features such as first-slow-then-fast convergence and periodic convergence, and that DSA is more suitable for the optimization problems where fitness functions are called more often. After comparing these four algorithms' results under 10 benchmark functions with different properties, we conclude that in more than half of the cases, DSA performs better, especially in low-dimensional unimodal functions, high-dimensional multimodal functions, step functions, and functions with random numbers. The experimental results indicate that DSA has better performance with fewer individuals, that DSA is sensitive to increase in dimensionality, and that the performances in low-dimensional multimodal functions and high-dimensional unimodal functions may not be good enough, and can be further optimized.

With reference to follow-up work, the application of DSA to other models, such as applying DSA in an extreme learning machine, is a further goal.

References

- Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. *Swarm Intelligence: from Natural to Artificial Systems*. Oxford University Press.
- Cura, T., 2012. A particle swarm optimization approach to clustering. *Expert Syst. Appl.*, **39**(1):1582-1588. <http://dx.doi.org/10.1016/j.eswa.2011.07.123>

- Dorigo, M., Birattari, M., 2010. Ant colony optimization. In: Sammut, C., Webb, G.I. (Eds.), *Encyclopedia of Machine Learning*. Springer, p.36-39.
http://dx.doi.org/10.1007/978-0-387-30164-8_22
- Dorigo, M., Maniezzo, V., Colomi, A., 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B*, **26**(1):29-41.
<http://dx.doi.org/10.1109/3477.484436>
- Ducatelle, F., di Caro, G.A., Gambardella, L.M., 2010. Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intell.*, **4**(3):173-198.
<http://dx.doi.org/10.1007/s11721-010-0040-x>
- Eberhart, R.C., Kennedy, J., 1995. A new optimizer using particle swarm theory. Proc. 6th Int. Symp. on Micro Machine and Human Science, p.39-43.
<http://dx.doi.org/10.1109/mhs.1995.494215>
- Eberhart, R.C., Shi, Y.H., 2001. Particle swarm optimization: developments, applications and resources. Proc. Congress on Evolutionary Computation, p.81-86.
<http://dx.doi.org/10.1109/CEC.2001.934374>
- Garnier, S., Gautrais, J., Theraulaz, G., 2007. The biological principles of swarm intelligence. *Swarm Intell.*, **1**(1):3-31. <http://dx.doi.org/10.1007/s11721-007-0004-y>
- Karaboga, D., 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Turkey.
- Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Glob. Optim.*, **39**(3):459-471. <http://dx.doi.org/10.1007/s10898-007-9149-x>
- Karaboga, D., Gorkemli, B., Ozturk, C., et al., 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.*, **42**(1):21-57.
<http://dx.doi.org/10.1007/s10462-012-9328-0>
- Kennedy, J., 2010. Particle swarm optimization. In: Sammut, C., Webb, G.I. (Eds.), *Encyclopedia of Machine Learning*. Springer, p.760-766.
http://dx.doi.org/10.1007/978-0-387-30164-8_630
- Mitchell, M., 1998. *An Introduction to Genetic Algorithms*. MIT Press.
- Mohan, B.C., Baskaran, R., 2012. A survey: ant colony optimization based recent research and implementation on several engineering domains. *Expert Syst. Appl.*, **39**(4):4618-4627.
<http://dx.doi.org/10.1016/j.eswa.2011.09.076>
- Parpinelli, R.S., Lopes, H.S., 2011. New inspirations in swarm intelligence: a survey. *Int. J. Bio-inspired Comput.*, **3**(1):1-16.
<http://dx.doi.org/10.1504/IJBIC.2011.038700>
- Poli, R., Kennedy, J., Blackwell, T., 2007. Particle swarm optimization. *Swarm Intell.*, **1**(1):33-57.
<http://dx.doi.org/10.1007/s11721-007-0002-0>
- Saleem, M., di Caro, G.A., Farooq, M., 2011. Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions. *Inform. Sci.*, **181**(20):4597-4624.
<http://dx.doi.org/10.1016/j.ins.2010.07.005>
- Whitley, D., 1994. A genetic algorithm tutorial. *Stat. Comput.*, **4**(2):65-85. <http://dx.doi.org/10.1007/BF00175354>
- Yao, X., Liu, Y., Lin, G.M., 1999. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.*, **3**(2):82-102.
<http://dx.doi.org/10.1109/4235.771163>