Engineering Applications of Artificial Intelligence 64 (2017) 401–414

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Dynamic social behavior algorithm for real-parameter optimization problems and optimization of hyper beamforming of linear antenna arrays

K. Prajindra Sankar [a,*], Tiong Sieh Kiong [b], Johnny Koh Siaw Paw [a]

[a] Electronics and Communication Engineering Department, College of Engineering, Universiti Tenaga Nasional, Kajang, Selangor, Malaysia
[b] Institute of Sustainable Energy, Universiti Tenaga Nasional, Kajang, Selangor, Malaysia

ARTICLE INFO

Keywords:
Meta-heuristics
Optimization
Swarm algorithms
Global optimization
Bio-inspired algorithms
Beamforming

ABSTRACT

The ever evolving complexity of real-world problems had become an impetus for the development of many new and efficient optimization algorithms. Meta-heuristics based on evolutionary computation and swarm intelligence are successful examples of nature-inspired optimization techniques. In this work, a new Dynamic Social Behavior (DSB) algorithm is proposed to solve global optimization problems. The DSB algorithm is based on the simulation of cooperative behavior of animal groups. In the proposed algorithm, individuals emulate the interaction of individuals based on biological laws of cooperative colony. This algorithm partially adopts the foraging strategy of animal groups and utilizes recruitment signal as a means of information transfer among individuals. In order to illustrate the proficiency and robustness of the proposed algorithm, it is compared with other well-known evolutionary algorithms. The comparison examines several series of widely used benchmark functions and an engineering problem on hyper beamforming optimization. The results testifies the superior performance of DSB compared with other state-of-the-art meta-heuristics.

1. Introduction

Meta-heuristics optimization algorithms has attracted great interest in the last two decades. Application of meta-heuristic algorithms have permeated into almost all areas of sciences, engineering and industries, from computational intelligence to business planning, from data mining to optimization, and from bioinformatics to industrial applications.

Despite the popularity and success of meta-heuristics, there remains a big question of which meta-heuristic technique is best suited to solve all optimization problem. In this connection, the No Free Lunch (NFL) theorem (Wolpert and Macready, 1997) would be very much relevant to answer the question. According to this theorem, it is impossible to have a meta-heuristic that is best suited for all optimization problems. Simply put, a specific meta-heuristic could perform extremely well on a set of problem and may show a poor performance on another set of problems. In this regard, the findings of NFL gives motivation to develop new meta-heuristics which makes this field of study highly active over the years.

Meta-heuristics algorithms are generally based on mathematical programming or formal logic which makes it an effective solver for complex optimization problems compared to conventional Evolutionary Algorithm (EA) and Swarm Intelligence (SI) methods. In order to improve the solution quality in EA, the population have to determine whether to explore the unexplored search space or to exploit the previously evaluated positions. The ability of an EA to search for the global optimum very much depends on its ability to find the proper balance between the exploration of the search space and exploitation of existing elements. Pure exploration increases the potential to seek for new solutions but degrades the precision of the evolutionary process. Likewise, pure exploitation enhances existing solutions but adversely causes the evolutionary process to get stuck in local optima. Up to date, the issue of achieving an ideal exploration–exploitation balance is still an open ended subject matter within the framework of evolutionary algorithms.

Generally, EA exhibits the uniform behavioral pattern as the individuals are defined with the same characteristics. Therefore, the algorithm lacks the search operator to generate a scenario with different individual characteristics. By incorporating these type of operators, the algorithm characteristics such as population diversity or searching capabilities could be improved. In branch of SI, quite a number of algorithms have emerged in the past decades. However, several algorithms such as PSO, ABC and the more recently proposed GWO are widely employed and studied among researchers. Nevertheless, these algorithms exhibit

* Corresponding author.
E-mail address: sankar@uniten.edu.my (K. Prajindra Sankar).

http://dx.doi.org/10.1016/j.engappai.2017.06.027
Received 4 January 2017; Received in revised form 5 June 2017; Accepted 29 June 2017
Available online 4 August 2017
0952-1976/

several shortcomings such as low solving precision, inability to escape from local minima and premature convergence (Yu and Li, 2015). These deficiencies are caused by the search operators which are employed to manipulate the individual positions. In the case of PSO, the position of every individual is updated in the subsequent iteration cycle based upon the inclination to move towards the best individual in the entire population. In the case of ABC, a randomly chosen individual will be the center of attraction whereas in the case of GWO, the attraction is directed towards the position of the best three agents. Even though such operators encourages dynamic behavioral pattern, the operator tends to divert the entire population towards the best particles or causes the population to diverge without control as the algorithm evolves. This in turn damages the exploration–exploitation balance and leads to premature convergence.

In this paper, a new SI based algorithm inspired by social behavior of communal groups named Dynamic Social Behavior (DSB) is proposed. This work attempts to find the proper mechanism to balance the exploitation and exploration with the ability to track the best solution. The employment of community based social behavior as a metaphor introduces new concepts in the field of evolutionary computing. The concepts involve dividing the population into various search categories and apply collective knowledge search operators to each categories. This strategy allows the population to maintain its size and yet makes it possible to enhance the exploration–exploitation balance. The social behavior element in DSB introduces a new computational mechanism which has three distinctive descriptions. Firstly, every individual is evaluated separately according to their behavioral characteristics. Secondly, all the members of the population share the same communication mechanism to allow the dissemination of crucial information pertaining to the process of changing the search operators. Thirdly, the search operators utilize the global information (positions of all the individual types) to modify the position of a particular individual type.

The proposed algorithm has been tested by solving the CEC 2005 benchmark problems as well as a complex real world problem related to hyper beam antenna design. The optimization of hyper beamforming is considered as a complex problem as it has strong nonlinearities with many local minima. An efficient optimization algorithm is required to generate the optimal hyper beam radiation pattern. The DSB algorithm is benchmarked with the original PSO, ABC and GWO algorithms respectively. This approach of benchmarking with the original algorithms was suggested in Fong et al. (2016) to prove the novelty of any new meta-heuristic design (the inner designs are fundamentally different from existing algorithms) as the variants of the original algorithms have several similar and widely used core components from the original algorithm. The results display a high performance of DSB in searching for a global optimum and as well as in generating optimal hyper beams.

This paper is organized as follows. Section 2 presents a literature review on SI algorithms. Section 3 describes the proposed DSB algorithm in detail. The problem descriptions and evaluation methods are outlined in Section 4 whereas Section 5 presents the experimental results followed by discussion. Finally, Section 6 concludes the work and suggests directions for future studies.

2. Literature review

Meta-heuristic algorithms are often nature-inspired and can be divided into three main branches namely evolutionary (EA), physics-based and SI algorithms. The first branch, EAs are generally inspired by concepts of natural evolution. Generally, the optimization is done by generating an initial random population and evolving the population over a period of certain iteration values. During each iteration, a new set of population would be created by imposing certain sets of operators on the previous generation. These sets of operators will ensure that the best candidate will have higher probability to participate in the generation of the new population thus creating a better population compared to the previous generation(s). This is the general principles of how an initial random population is evolved over the course of generations. Some of the most prominent EAs are Genetic Algorithm (GA) (Goldberg, 1989), Genetic Programming (GP) (Koza, 1992), Evolutionary Programming (EP) (Yao et al., 1999), Evolution Strategy (ES) (Beyer and Schwefel, 2002), Differential Evolution (DE) (Storn and Price, 1997) and Biogeography-Based Optimizer (BBO) (Simon, 2008).

The second branch of meta-heuristics focuses on physics-based techniques that mimics certain physical laws. Physical rules such as electromagnetic force, gravitational force, weights and inertia force are applied to propel the movement of individuals in the search space. This mechanism is what differentiates EAs and physics-based techniques. Some of the most popular algorithms are Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), Curved Space Optimization (CSO) (Moghaddam et al., 2012), Gravitational Local Search (GLSA) (Hosseinabadi et al., 2015), Charged System Search (CSS) (Kaveh and Talatahari, 2011), Central Force Optimization (CFO) (Formato, 2009), Small-World Optimization (SWO) (Xiaohu et al., 2009) and Artificial Chemical Reaction Optimization (ACROA) (Alatas, 2011).

The third branch of meta-heuristics is the SI algorithms which will the prime focus of this work. The mechanism of SI algorithms are almost similar to physics-based algorithm but the search process is purely inspired by the social behavior of swarms, flocks, herds or schools of creatures in nature. The individual navigation is done by imposing certain operators based on the mathematical model of social behavior of communal groups and collective social knowledge. Some of the SI algorithms are as follows:

- Ant Colony Optimization (ACO) (Dorigo and Stützle, 2004).
- Cuckoo Search (CS) (Yang, 2013).
- Firefly Algorithm (FA) (Yang, 2013).
- Bat Algorithm (BA) (Yang and He, 2013).
- Dolphin Partner Optimization (DPO) (Shiqin et al., 2009).
- Monkey Search (MS) (Mucherino and Seref, 2007).

Some of the popular SI algorithms are Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), Artificial Bee Colony (ABC) (Karaboga and Basturk, 2007) and the recent Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014). PSO is represented by a swarm particles and their respective positions in the search space denotes the possible solution for the optimization problem. PSO utilizes the information of individual experience and socio-cognitive tendency to manipulate the movements of these particles. These two kinds of information correspond to cognitive learning and social learning which will eventually lead the population to perform better optimization (Yu and Li, 2015). ABC mimics the collective behavior of bees in finding food sources. The bees are divided into three groups namely the scout bees, the onlooker bees and the employee bees. The scout bees are responsible for exploring the search space, whereas the onlooker and the employee bees exploit the potential solutions found by scout bees (Mirjalili et al., 2014). The GWO is a recently proposed SI algorithm which mimics the social leadership and hunting behavior of grey wolves in their natural habitat. The population is divided into four groups: alpha, beta, delta and omega. The first three groups of wolves will guide the other wolves towards the promising areas of the search space.

Even though PSO, ABC and GWO are one of the most popular swarm algorithms for solving complex optimization problems, they display certain flaws such as premature convergence, inability to jump over local optima and prone to stagnation in local solutions (Wang et al., 2011; li Xiang and qing An, 2013; Mirjalili et al., 2014). Such problems could have been caused by the set of operators applied on each individual positions. In the case of PSO, every individual position is updated during every iteration based on the attraction towards the position of the best individual seen so far. In ABC, the individual position update is done based on attraction towards randomly chosen individuals whereas in GWO, the attraction is towards a fixed set of individuals. As the iteration evolves, these operators cause the entire population to revolve around the best individual or diverges without control. In either case, it damages the exploration–exploitation balance and leads to premature convergence (Banharnsakun et al., 2011; Wang et al., 2013).

The principle of swarm phenomenon or collective nature has been extensively studied in animal behavior ecology. Communal animals gather and live together to increase the chances of securing a food source (foraging) and reduce the energy cost in this process (Sumpter, 2010). In order to emulate the social foraging behavior, researchers have established two foraging models namely Producer–Scrounger (PS) model (Barnard and Sibly, 1981) and Information Sharing (IS) model (Clark and Mangel, 1984). In the PS model, the individuals are divided into leaders and followers whereas in IS model, individuals perform search operation while communicating with other individuals to look for better solution potentials. The PS model has already been introduced in GWO and have displayed remarkable outcomes (Mirjalili et al., 2014). This leaves room to venture into the IS model and also one of the motivation for this work to incorporate IS model to control the searching pattern of the proposed algorithm.

In recent times (late 2015 to early 2017), there has been various variants of PSO, ABC and GWO such as efficient player selection strategy based diversified particle swarm optimization (Agarwalla and Mukhopadhyay, 2017), composite particle algorithm (De et al., 2016), cooperative learning PSO (Alexandridis et al., 2016), comprehensive learning ABC (Su et al., 2017), adaptive ABC (Song et al., 2017), multi-objective GWO (Mirjalili et al., 2016) and modified GWO (Mittal et al., 2016). These development implies that each meta-heuristic method possess certain advantages and works well when applied in certain engineering domain. In (Fong et al., 2016), the authors managed to demonstrate that any incremental modification on any meta-heuristics can potentially enhance its performance. It is a matter of which components are chosen to assemble into a new hybrid or what parameter settings are chosen in applying the hybrid to a particular problem (Fong et al., 2016). This work aims to develop inner designs which are fundamentally different from the existing algorithms as will be explained in the subsequent sections.

3. Dynamic social behavior

Dynamic Social Behavior (DSB) algorithm is a new paradigm for designing evolutionary optimization algorithms inspired by concepts adopted from evolution of social behavior of communal groups and collective social knowledge. In a nutshell, social behavior relates how a particular social member interact with each other within its natural habitat. As such, the proposed algorithm utilizes certain interaction rules with information transfer strategies (recruitment signal) which resembles collective social behavior as optimization operators on a population of individuals. Another new feature included in the optimization operator is to dynamically control the search process based on the average performance of the population to ensure that the search process is guided towards the median value. Hence, the name Dynamic Social Behavior was adopted for this algorithm.

Following the approach of Couzin et al. (2002), Cuevas et al. (2012) and Sumpter (2010), DSB emulates the behavior of individuals based upon the relative position and orientation of individuals with respect to one another. This is achieved by applying local attraction or repulsion operators according to the signals emitted by the individuals. In this approach, every individual position from the population represents a unique solution within the search space. Depending on the individual supremacy, every individual is assigned with a respective fitness score indicating its supremacy from highest to the lowest with respect to the whole population. The overall optimization process exhibits the collective social behavior in communal groups. The social members living in a communal habitat utilizes acoustic based recruitment signal strategy as a means for information transfer and communication. Each member in the communal habitat holds a position and the quality (or fitness) of the solution which is evaluated via the objective function which represents the potential of finding a food source at the position. The members have the freedom to move freely within the communal habitat. However, they cannot leave the communal habitat as the positions off the communal habitat represents unfeasible solutions to the optimization problem. Every social member will emanate an acoustic signal which acts as an information exchange about the location of food from an informed individual to an uninformed individual and also their respective locations as well.

3.1. Social member structure

The individuals from the population is referred as the social members which act as the agents of DSB to perform optimization. During initialization, a predefined number of social members, m are evenly distributed in the communal habitat. Each member holds the following information:

- The positions of m in the communal habitat.
- The fitness values of the present position of m.

The DSB is an iterative algorithm similar to many other evolutionary algorithms whereby the initial phase is to randomly initialize the entire population and distribute it evenly across the search space. The initialization phase begins by defining N social member positions in the search space S. Each member position, m is an n-dimensional vector containing the parameter values to be optimized. The predefined lower initial parameter bound B_j^{low} and upper initial parameter bound B_j^{high} ensures that the social members are uniformly distributed, as described by the following equation:

$$m_{i,j}^0 = B_j^{low} + rand(0,1)\cdot(B_j^{high} - B_j^{low}) \quad i = 1,2,\ldots,N;\ j = 1,2,\ldots,n \tag{1}$$

where j and i are the parameter and individual indexes respectively. Hence, $m_{i,j}$ is the jth parameter of the ith member position. The superscript zero indicates the initial population whereas the $rand(0,1)$ function generates a random number within the range of 0–1.

3.2. Fitness allocation

In communal habitat, the size of the member plays a vital role in determining the individual capacity to perform better over the assigned tasks. Member with a larger size implies that the individual has a higher fitness score and naturally tends to perform better. In order to emulate this approach, every individual is assigned a weight value, w_i which represents the quality of the solution which is affiliated to the member i of the population M. The fitness allocation is done by calculating the weight of every member based on the following equation:

$$w_i = \frac{F(m_i) - worst_m}{best_m - worst_m} \tag{2}$$

where $F(m_i)$ represents the fitness score obtained by evaluating the objective function $F(\cdot)$ with respect to the member position m_i. This work focuses on minimizing optimization problem hence the values $worst_m$ and $best_m$ are defined as:

$$best_m = \min_{i\in\{1,2,\ldots,N\}} (F(m_i)) \quad \text{and} \quad worst_m = \max_{i\in\{1,2,\ldots,N\}} (F(m_i)). \tag{3}$$

3.3. Modeling of the acoustic based recruitment signal

Information transfer is one of most effective means of collective coordination of all the members in the population. This information is encoded in an acoustic based recruitment signal and transmitted among the communal members. The signal strength and traversing distance depends on the weight and distance of the member which has generated them. Since the received signal strength is relative to the signal source, it is only natural that members located in a distant position will detect weaker signals whereas the members with a close proximity with the member which has generated the signals will detect a stronger signals.

The information transmitted by member j will be perceived by member i based on the following model:

$$Sig_{i,j} = w_j \cdot e^{-d_{i,j}^2} \tag{4}$$

where $d_{i,j} = \|m_i - m_j\|$, which is the Euclidean distance between the member i and j.

By considering all the possible pair of individuals within the entire population, it is possible to compute all the perceived signals. Nevertheless, the DSB takes a different approach towards the signal transfer to make the information transfer more effective rather than just a random information transfer. The two special signal transfer has been identified as:

1. The signal transmitted by member $c\,(m_c)$ and perceived by member $i\,(m_i)$. Member $c\,(m_c)$ has two distinctive characteristics: it has a higher weightage compared to member $i(w_c > w_i)$ and has the shortest distance to member i. The signal perceived by member i, $Sigc_i$ is expressed as:

$$Sigc_i = w_c \cdot e^{-d_{i,c}^2}. \tag{5}$$

2. The signal $Sigb_i$ perceived by the member i as a result of the information transmitted by the member $i\,(m_b)$, with b being the individual holding the best weight (best fitness value) of the entire population M, such that $w_b = max_{i\in\{1,2,\ldots,N\}}(w_i)$. The signal transmitted by member $i\,(m_b)$ which holds the best fitness score (best weight) in the entire population and perceived by member i. The signal perceived by member i, $Sigb_i$ is expressed as:

$$Sigb_i = w_b \cdot e^{-d_{i,b}^2} \tag{6}$$

where $w_b = max_{i\in\{1,2,\ldots,N\}}(w_i)$.

Fig. 1 illustrates the difference between these two signal transmissions: (a) $Sigc_i$ and (b) $Sigb_i$.

3.4. Cooperative search operators

The efficacy of any meta-heuristics is more or less governed by exploration and exploitation strategies which are implemented as search operators iteratively (Fong et al., 2016). These dual steps are categorized as global exploration (exploration) and local intensification (exploitation). Exploration is the strategy undertaken to diversify its search agents from its current position over the search space by means of sporadically migrations. This function enhances the chances of finding a better solution and avoids the problem of getting stuck at a local optima. However, the process of exploitation is a means of steering the search agents to a given neighborhood strategically (Fong et al., 2016). DSB algorithm incorporates certain interaction rules over other communal members as a measure to implement these dual step process. The first process (exploration) resembles collective social behavior that manipulates the orientation and relative position of members with respect to their neighbors. This is achieved by applying local attraction and repulsion operators in accordance with the signal strength perceived over the communal habitat. As explained in the previous section, since the signal strength and traversing distance depends on the weight and distance of the member which has generated them, either a large member (higher weight) would have generate the signal or the member which is perceiving the strong signal is situated in close proximity with the member which had produced them. The attraction and repulsion operation over the selected individuals will be decided based upon internal factors such as random phenomena and curiosity. The second process (exploitation) is achieved by dividing the population into sub populations and steering the weak individuals towards the weighted mean of the population.

3.4.1. Exploration operators

To implement these attraction and repulsion operations over the communal members, a new operator is defined. In a nutshell, the attraction and repulsion operations involves the positional shift of the individual i during every iteration cycle. These positional shifts are driven by a combination of three elements. The first element involves the positional shift towards the individual which produces the signal $Sigc_i$ and holds a higher weight. The second one involves the positional shift with regards to the best individual in the entire population M and emanates the signal $Sigb_i$. Lastly, the third element utilizes random positional movement.

The selection criteria to implement either the attraction or the repulsion operator is modeled as a stochastic decision since the implementation depends on several random scenario. In order to emulate these random scenarios, a uniform random number r_m is generated within the range $[0, 1]$ and evaluated against a threshold value. An attraction movement will be generated if the r_m value is less than the threshold value, PF otherwise a repulsion movement will be implemented. The attraction and repulsion operator can be expressed as:

$$\mathbf{m}_i^{k+1} = \begin{cases} \mathbf{m}_i^k + \alpha \cdot Sigc_i \cdot (\mathbf{m}_c - \mathbf{m}_i^k) + \beta \cdot Sigb_i \cdot (\mathbf{m}_b - \mathbf{m}_i^k) \\ \quad + \delta \cdot \left(rand - \frac{1}{2}\right) \quad \text{with probability } PF \\ \mathbf{m}_i^k + \alpha \cdot Sigc_i \cdot (\mathbf{m}_c - \mathbf{m}_i^k) - \beta \cdot Sigb_i \cdot (\mathbf{m}_b - \mathbf{m}_i^k) \\ \quad + \delta \cdot \left(rand - \frac{1}{2}\right) \quad \text{with probability } 1 - PF \end{cases} \tag{7}$$

where k denotes the iteration number whereas α, β, δ and rand are random numbers between [0,1]. The nearest individual to i that holds a higher weight and the best individual of the entire population M are denoted as member m_c and m_b respectively.

This type of interaction operator avoids the quick concentration of members at a particular point and encourages local interaction within its neighborhood. That is because the new positional shift represents a movement that is a combination of previous position vector over the global best individual m_b and local best individual m_c during that particular iteration. The random operator also allows the members to explore unexplored search space. The implementation of this scheme brings about two advantages. Firstly, the algorithm is not prone to premature convergence since the members are not solely driven towards the global best position. Secondly, the scheme encourages the members to explore their own neighborhood in advance before converging towards the global best position. Such implementation enhances the exploration nature of the algorithm hence increasing its capability to perform global search.

3.4.2. Exploitation operators

It is certainly undesirable to have the search process under the strong influence of either very good members or extremely bad members. Therefore, a filtering mechanism is introduced to partially control the search process based upon the average performance of a sub-group of the entire population. In order to incorporate the cooperative search operator, the population is divided into two groups namely the dominant members D and the non-dominant members ND. Comparatively, D members have better fitness characteristics than the ND members. The segregation is done by evaluating their respective weights with regards to the median value. The dominant individuals D are members with a weight value above the median value of the population whereas the remaining individuals are categorized as non-dominant ND members. The computation is done by arranging the member population M ($M = \{m_1, m_{2,} \ldots, m_N\}$) in decreasing order according to their weight value. The member whose weight is w_{med} and located in the middle is considered as the median member. To prompt the new position of the communal member, the search operator is modeled as:

$$\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha \cdot \left(\frac{\sum_{i=1}^{N} \mathbf{m}_i^k \cdot w_i}{\sum_{i=1}^{N} w_i} - \mathbf{m}_i^k \right) \tag{8}$$

$$\text{if } w_i \leqslant w_{med}$$

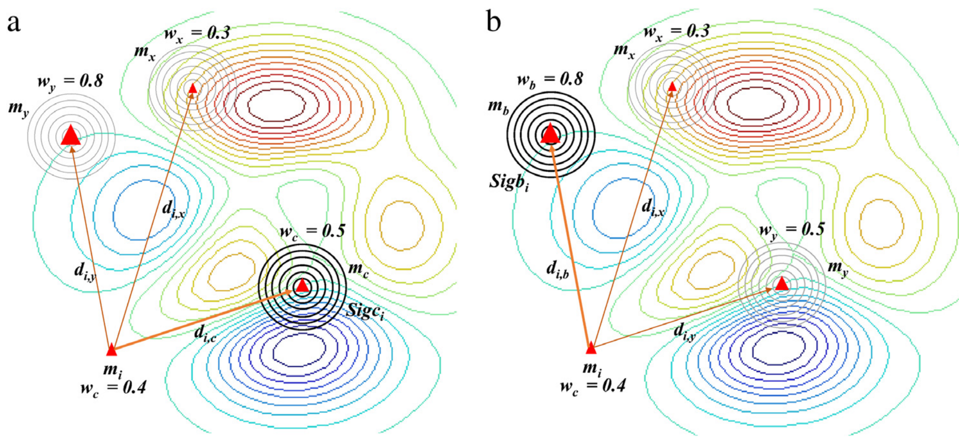

Fig. 1. Signal transmissions: (a) $Sigc_i$ and (b) $Sigb_i$.

where $(\sum_{i=1}^{N}\mathbf{m}_i^k \cdot w_i / \sum_{i=1}^{N} w_i)$ corresponds to the weighted mean of the population M.

The benefit of this operator is that it allows the ND members to be steered towards the weighted mean of the entire population. This search process is being partially controlled by the average performance of the population hence protecting the search process from being influenced by either very good members or extremely bad members.

3.5. Computational procedure

The computational procedure for the proposed algorithm can be outlined as follows:

Step 1: Define the population size in the entire population M. N is considered as the total number of n-dimensional communal members.

$$m_{i,j}^0 = B_j^{low} + rand(0,1)\cdot(B_j^{high} - B_j^{low})$$
$$i = 1,2,\dots,N;\ \ j = 1,2,\dots,n$$

where rand generates a random number between 0-1 whereas floor(.) maps a real number to an integer number.

Step 2: Calculate the weight of every individual of M (Section 3.2).

Algorithm 1 Weight Assignment

for $(i = 1, i < N+1; i++)$ **do**
  $w_i = \frac{F(m_i) - worst_m}{best_m - worst_m}$
end for

where $best_m = \min_{i\in\{1,2,\dots,N\}} (F(m_i))$ and $worst_m = \max_{i\in\{1,2,\dots,N\}} (F(m_i))$

Step 3: Initiate individual move based on the cooperative search operators (Section 3.4).

Algorithm 2 Cooperative Operator

for $(i = 1, i < N+1; i++)$ **do**
  Calculate $Sigc_i$ and $Sigb_i$ (Section C)
  if $(r_m < PF)$; where $r_m \in rand(0,1)$ **then**
    $\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha\cdot Sigc_i\cdot(\mathbf{m}_c - \mathbf{m}_i^k) + \beta\cdot Sigb_i\cdot(\mathbf{m}_b - \mathbf{m}_i^k) + \delta\cdot(rand - \frac{1}{2})$
  else
    $\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha\cdot Sigc_i\cdot(\mathbf{m}_c - \mathbf{m}_i^k) - \beta\cdot Sigb_i\cdot(\mathbf{m}_b - \mathbf{m}_i^k) + \delta\cdot(rand - \frac{1}{2})$
  end if
end for

Step 4: Perform the median search to improve individuals search pattern (Section 3.4).

Step 5: The process is completed if the stopping criteria is met else repeat Step 2.

Algorithm 3 Median Search

Find the median individual w_{med} from M
for $(i = 1, i < N+1; i++)$ **do**
  if $w_i \leqslant w_{med}$ **then**
    $\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha\cdot\left(\frac{\sum_{i=1}^{N}\mathbf{m}_i^k\cdot w_i}{\sum_{i=1}^{N}} - \mathbf{m}_i^k\right)$
  end if
end for

4. Problem descriptions and evaluation method

This section will describe the 25 well-known real-parameter optimization benchmark problems and an engineering optimization problem on hyper beamforming to produce narrow First Null Beam Width (FNBW) with reduced Side Lobe Level (SLL) to judge the performance of the proposed algorithm.

4.1. Benchmark functions

A set of 25 benchmark functions are utilized to evaluate the performance of DSB. Table 1 lists the benchmark functions. The CEC 2005 special session on real-parameter optimization defines the problem definitions and evaluation criteria for each one of these base functions. These benchmark functions are classified into four categories:

- Group I: $f_1 - f_5$ are unimodal functions.
- Group II: $f_6 - f_{12}$ are multimodal functions.
- Group III: $f_{13} - f_{14}$ are expanded multimodal functions.
- Group IV: $f_{15} - f_{25}$ are hybrid multimodal functions.

The list of all the benchmark functions, its implementation and evaluation methods can be found in Suganthan et al. (2005). In order to increase the difficulty level, all the benchmark functions are either shifted or rotated minimization problems where D denotes the dimension of the problem. Shifted or rotated functions indicate that their global minimum is not located at zero and possess a challenge for the algorithm to locate the global minimum easily.

The unimodal functions in Group I are utilized to test the converging performance of DSB which is suitable for benchmarking its exploitation capabilities. Functions in Group II are employed to evaluate the ability of DSB to escape from getting trapped in local optima and avoid premature convergence. Group II functions have a large number of local minima points which serves the purpose of benchmarking the exploration ability of DSB. To push the searching capability of DSB to another level, more complex functions are introduced in Group III. Finally, Group IV functions consist of various sub-components with different properties which will be able to test the performance of DSB in handling both

Table 1
Benchmark functions.

Function	Search space	f_{bias}	Name
$f_1(x)=\sum_{n=1}^{N} z_i^2 + f_{bias}$	$x\in[-100,100]^D$	−450	Shifted Sphere Function
$f_2(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 + f_{bias}$	$x\in[-100,100]^D$	−450	Shifted Schwefel's Problem 1.2
$f_3(x)=\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2 + f_{bias}$	$x\in[-100,100]^D$	−450	Shifted Rotated High Conditioned Elliptic Function
$f_4(x)=(\sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2) * (1+0.4\lvert N(0,1)\rvert) + f_{bias}$	$x\in[-100,100]^D$	−450	Shifted Schwefel's Problem 1.2 with Noise in Fitness
$f_5(x)=\max\{\lvert A_i x - B_i\rvert\} + f_{bias}$	$x\in[-100,100]^D$	−310	Schwefel's Problem 2.6 with Global Optimum on Bounds
$f_6(x)=(\sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i-1)^2)) + f_{bias}$	$x\in[-100,100]^D$	390	Shifted Rosenbrock's Function
$f_7(x)=(\sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D}\cos(\frac{z_i}{\sqrt{i}}) + 1 + f_{bias})$	$x\in[0,600]^D$	−180	Shifted Rotated Griewank's Function without Bounds
$f_8(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2})$ $-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)) + 20 + e + f_{bias}$	$x\in[-32,32]^D$	−140	Shifted Rotated Ackley's Function with Global Optimum on Bounds
$f_9(x)=\sum_{i=1}^{D}(z_i^2 - 10\cos(2\pi z_i) + 10) + f_{bias}$	$x\in[-5,5]^D$	−330	Shifted Rastrigin's Function
$f_{10}(x)=\sum_{i=1}^{D}(z_i^2 - 10\cos(2\pi z_i) + 10) + f_{bias}$	$x\in[-5,5]^D$	−330	Shifted Rotated Rastrigin's Function
$f_{11}(x)=(\sum_{i=1}^{D}(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(z_i+0.5))]))$ $-D\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k\cdot 0.5)] + f_{bias}$	$x\in[-0.5,0.5]^D$	90	Shifted Rotated Weierstrass Function
$f_{12}(x)=\sum_{i=1}^{D}(A_i - B_i(x))^2 + f_{bias}$ $A_i=\sum_{i=1}^{D}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j)$, $B_i(x)=\sum_{i=1}^{D}(a_{ij}\sin x_j + b_{ij}\cos x_j)$, $for\ i=1,\dots,D$	$x\in[-\pi,\pi]^D$	−460	Schwefel's Problem 2.13
$f_{13}(x)=F8(F2(z_1,z_2)) + F8(F2(z_2,z_3)) + \dots$ $+F8(F2(z_{D-1},z_D)) + f_{bias}$ $F8(x)=\sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D}\cos(\frac{z_i}{\sqrt{i}}) + 1$ $F2(x)=\sum_{i=1}^{D-1}(100(x_i^2 - x_{i+1})^2 + (x_i-1)^2)$	$x\in[-3,1]^D$	−130	Shifted Expanded Griewank's plus Rosenbrock's Function (F8F2)
$f_{14}(x)=F(z_1,z_2) + F(z_2,z_3) + \dots + F(z_{D-1},z_D)$ $+F(z_D,z_1) + f_{bias}$ $F(x,y)=0.5+\frac{(sin^2(\sqrt{x^2+y^2})-0.5)}{(1+0.001(x^2+y^2))^2}$	$x\in[-100,100]^D$	−300	Shifted Rotated Expanded Scaffer's F6 Function
$f_{15}(x)=\sum_{i=1}^{n}\{w_i * [f_i'((x-o_i)/\lambda_i * M_i) + bias_i]\} + f_{bias}$	$x\in[-5,5]^D$	120	Hybrid Composition Function 1
$f_{16}(x)=f_{15}(M_i)$	$x\in[-5,5]^D$	120	Rotated Version of Hybrid Composition Function f_{15}
$f_{17}(x)=G(x) * (1+0.2\lvert N(0,1)\rvert) + f_{bias}$ $G(x)=f_{16} - f_{bias16}$	$x\in[-5,5]^D$	120	f_{16} with Noise in Fitness
$f_{18}(x)=f_{15}(M_i)$	$x\in[-5,5]^D$	10	Rotated Hybrid Composition Function 1
$f_{19}(x)=f_{18}(M_i)$	$x\in[-5,5]^D$	10	Rotated Hybrid Composition Function 1 with narrow basin global optimum
$f_{20}(x)=f_{18}(M_i)$	$x\in[-5,5]^D$	10	Rotated Hybrid Composition Function 1 with global optimum on the bounds
$f_{21}(x)=\sum_{i=1}^{n}\{w_i * [f_i'((x-o_i)/\lambda_i * M_i) + bias_i]\} + f_{bias}$	$x\in[-5,5]^D$	360	Rotated Hybrid Composition Function 2
$f_{22}(x)=f_{21}(M_i)$	$x\in[-5,5]^D$	360	Rotated Hybrid Composition Function 2 with high condition number matrix
$f_{23}(x)=f_{21}(M_i)$	$x\in[-5,5]^D$	360	Non-Continuous Rotated Hybrid Composition Function 2
$f_{24}(x)=\sum_{i=1}^{n}\{w_i * [f_i'((x-o_i)/\lambda_i * M_i) + bias_i]\} + f_{bias}$	$x\in[-5,5]^D$	260	Rotated Hybrid Composition Function 3
$f_{25}(x)=f_{24}(M_i)$	$x\in[2,5]^D$	260	Rotated Hybrid Composition Function 3 without bounds

exploration and exploitation capabilities. Due to the massive number of local optima present in such functions, the ability of DSB to avoid local optima can be examined.

The evaluation methods are set as follows. The dimension for each benchmark function, D is set as 10. Each function is evaluated for 25 runs and in each run, the termination criteria is set to a maximum number of $10^4 \times n$ function evaluations. According the requirements set by (Suganthan et al., 2005), only one fixed parameter setting is allowed to be used for evaluating the functions in all the groups. To make a fair comparison of the performance of DSB, similar natured algorithms such as PSO, ABC and GWO are selected to be benchmarked as well. All the source codes are adopted from the original author with some minor changes to adapt to the current benchmark functions. Nevertheless, the main function and the logic of the algorithms are untouched.

The benchmark test functions utilized in this work contains continuous variables with explicit constraints and limited to static environments. To adapt DSB to optimize mixed integer linear programming (MILP) problems, certain modifications are required. Mixed integer linear programming problems have both implicit constraints and discrete variables so two methods can be used to deal with these situation. First is to implement penalty functions to eliminate the implicit constraints. Secondly, a sigmoid function (piece-wise linear interpolation) can be introduced to deal with the discrete variables. The discrete variables are always 0 or 1 variables, so the sigmoid functions can be used to constrain the variables within [0, 1]. The sigmoid value is compared with a random value within [0, 1] to determine its discrete value. The MILP problems optimization is beyond the scope of this work but future attempts will be made to employ DSB for MILP problems or dynamic environments.

4.2. Hyperbeam design equations

Compared to conventional beam forming techniques, hyperbeam technique generates a narrow beam with an improved FNBW and reduced SLL which are dependent on the exponent parameter value (μ) being selected. The inter element spacing is varied from $\lambda/2$ to λ uniformly in order to achieve the above mentioned improvements. Hyperbeam is obtained by subtraction of sum and difference beams, each raised to the power of exponent μ (Ram et al., 2013). The sum beam pattern is generated by summing the absolute complex values of both left and right half beams as depicted in Fig. 2. On the contrary, the difference beam pattern is generated by taking the absolute magnitude of the difference of complex left half beam and right half beam signals as shown in Fig. 3. From the figure, it can be observed that the difference beam produces a minimum in the direction of the sum beam at zero degree which indicates that it is the difference beam.

A broadside linear array of N equally spaced isotropic elements is considered which is symmetric in both geometry and excitation with respect to the array center (Anitha et al., 2012). For broadside beams, the array factor is given in Balanis (2012) but a slight modification is done to the formula by adding the beam-width control coefficient to control the beam-width:

$$AF(\theta) = \sum_{n=1}^{N} I_n e^{j(n-1)Kd[a.\sin\theta\cos\phi - \sin\theta_0\cos\phi_0]} \quad (9)$$

where θ = angle of radiation of electromagnetic plane wave; d = inter-element spacing; K = propagation constant; N = total number of elements in the array; I_n = excitation amplitude of nth element; a = beam-width control coefficient. The equations for the creation of sum,

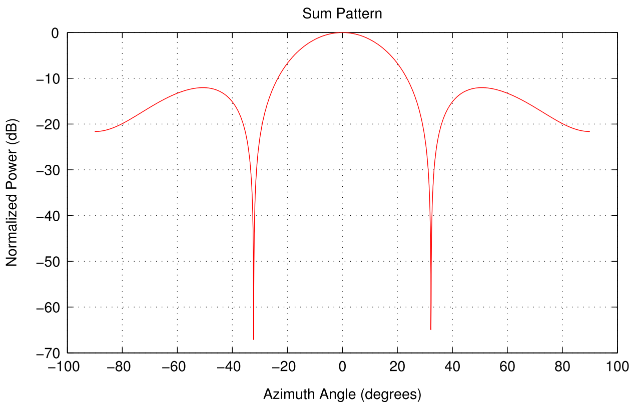

Fig. 2. Sum beam pattern for 10-element linear array with $\mu = 0.5$.

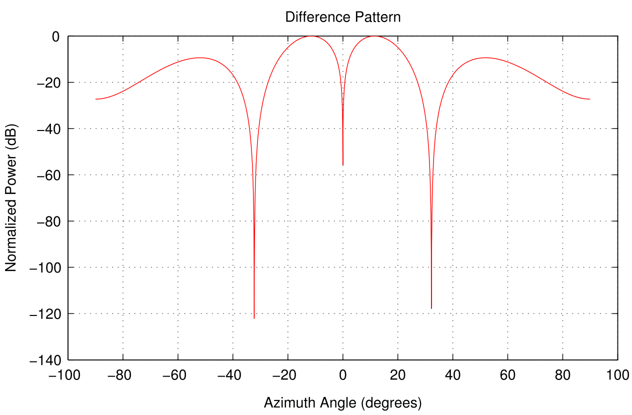

Fig. 3. Difference beam pattern for 10-element linear array with $\mu = 0.5$.

difference, and simple hyperbeam pattern in terms of two half beams are as follows (Ram et al., 2013):

Sum pattern:

$$Sum\,(\theta) = |R_L| + |R_R|. \tag{10}$$

Difference pattern:

$$Diff\,(\theta) = |R_L - R_R| \tag{11}$$

where

$$\begin{aligned} R_L &= \sum_{n=1}^{N/2} I_n e^{j(n-1)Kd[a.\sin\theta\cos\phi - \sin\theta_0\cos\phi_0]} \\ R_R &= \sum_{n=N/2+1}^{N} I_n e^{j(n-1)Kd[a.\sin\theta\cos\phi - \sin\theta_0\cos\phi_0]}. \end{aligned} \tag{12}$$

Hyperbeam is obtained by subtraction of sum and difference beams, each raised to the power of the exponent μ; the general equation of hyperbeam is a function of hyperbeam exponent μ as given in

$$AF_{Hyper}(\theta) = \left\{(|R_L| + |R_R|)^{\mu} - (|R_L - R_R|)^{\mu}\right\}^{1/\mu} \tag{13}$$

where μ ranges from 0.2 to 1. If μ lies below 0.2, hyperbeam pattern will contain a large depth spike at the peak of the main beam without changing in the hyperbeam pattern. If μ increases more than 1, side lobes of hyperbeam will be more as compared to conventional radiation pattern (Ram et al., 2013).

All the antenna elements are assumed isotropic. Only amplitude excitations, inter-element spacing and beam-width control coefficient are used to change the antenna radiation pattern. The fitness function (FF) for improving the SLL of radiation pattern of hyperbeam linear antenna arrays is given in Ram et al. (2013):

$$FF = Max\frac{|AF_{Hyper}(\theta_{msl\_left}, I_n)|}{|AF_{Hyper}(\theta_0, I_n)|} + Max\frac{|AF_{Hyper}(\theta_{msl\_right}, I_n)|}{|AF_{Hyper}(\theta_0, I_n)|} \tag{14}$$

where θ_0 is the angle where the highest maximum of central angle is attained in $\theta \in [-\pi/2, \pi/2]$. θ_{msl_left} is the angle where maximum side lobe $AF_{Hyper}(\theta_{msl\_left}, I_n)$ is attained in the lower band of hyperbeam pattern. θ_{msl_right} is the angle where the maximum side lobe $AF_{Hyper}(\theta_{msl\_left}, I_n)$ is attained in the upper side band of hyperbeam pattern. In FF, both numerator and denominator are in absolute magnitude. Minimization of FF means maximum reduction of SLL. GWO, PSO, ABC and DSB are employed individually for minimization of FF by optimizing current excitation weights of elements (I_n), inter-element spacing (d) and beam-width control coefficient (a). Results of the minimization of FF and SLL while maintaining a narrow beam-width are described in the subsequent section.

5. Experimental results and analysis

This section discusses the numerical experimental results and analyzes the performance of the proposed algorithm on an engineering problem.

5.1. CEC 2005 problems

The results from the 25 benchmark functions tested on DSB are compared with the results produced by GWO, PSO and ABC evaluated on the same 25 functions. The population size has been set to 30 individuals in all algorithms. The stopping criterion is set to $10^4 \times n$ function evaluations to conform with the requirements mentioned in the CEC 2005 technical report (Suganthan et al., 2005).

For each algorithms, the parameter settings are configured as follows:

1. GWO: The variables a, A and C are adaptive values which will be updated automatically during the optimization process. In order to strike the balance between exploration and exploitation, variable a is decreased from 2 to 0 whereas A is decreased linearly throughout the iteration process to emphasize exploitation (Mirjalili et al., 2014). On the other hand, the variable C which emphasizes the exploration/exploitation at any stage is randomly generated throughout optimization.
2. PSO: The parameters are set to $c_1 = 2$ and $c_2 = 2$; the weight factor decreases linearly from 0.9 to 0.2 (Kennedy and Eberhart, 1995).
3. ABC: The parameter $limit$ is set to 100 whereas all the other settings have been implemented as it is (Karaboga and Basturk, 2007).
4. DSB: The parameter PF is set to 0.7 upon experimental trial and maintained for all function evaluations.

The results of 25 runs for GWO, PSO, ABC and DSB are tabulated in Tables 2–5. The tabulated data displays the comparative performance indexes such as the mean, median, the best and the standard deviation of all the benchmarked algorithms. The best outcome from each function is highlighted in bold. In addition to that, to evaluate the performance similarity of DSB with other algorithms, a series of Wilcoxon rank sum tests on null hypothesis is performed. The null hypothesis is used to express the relationship between two quantities. The results from the null hypothesis (all p values are less than 5% significance level) indicate that DSB algorithm has similar relationship with the other benchmarked algorithms and the results are statistically significant (not occurred by coincidence due to common noise contained in the process). This ensures fair comparison of performance indexes with the other benchmarked algorithms. According to the tabulated results, DSB outperformed GWO, PSO and ABC in all the benchmark functions. The

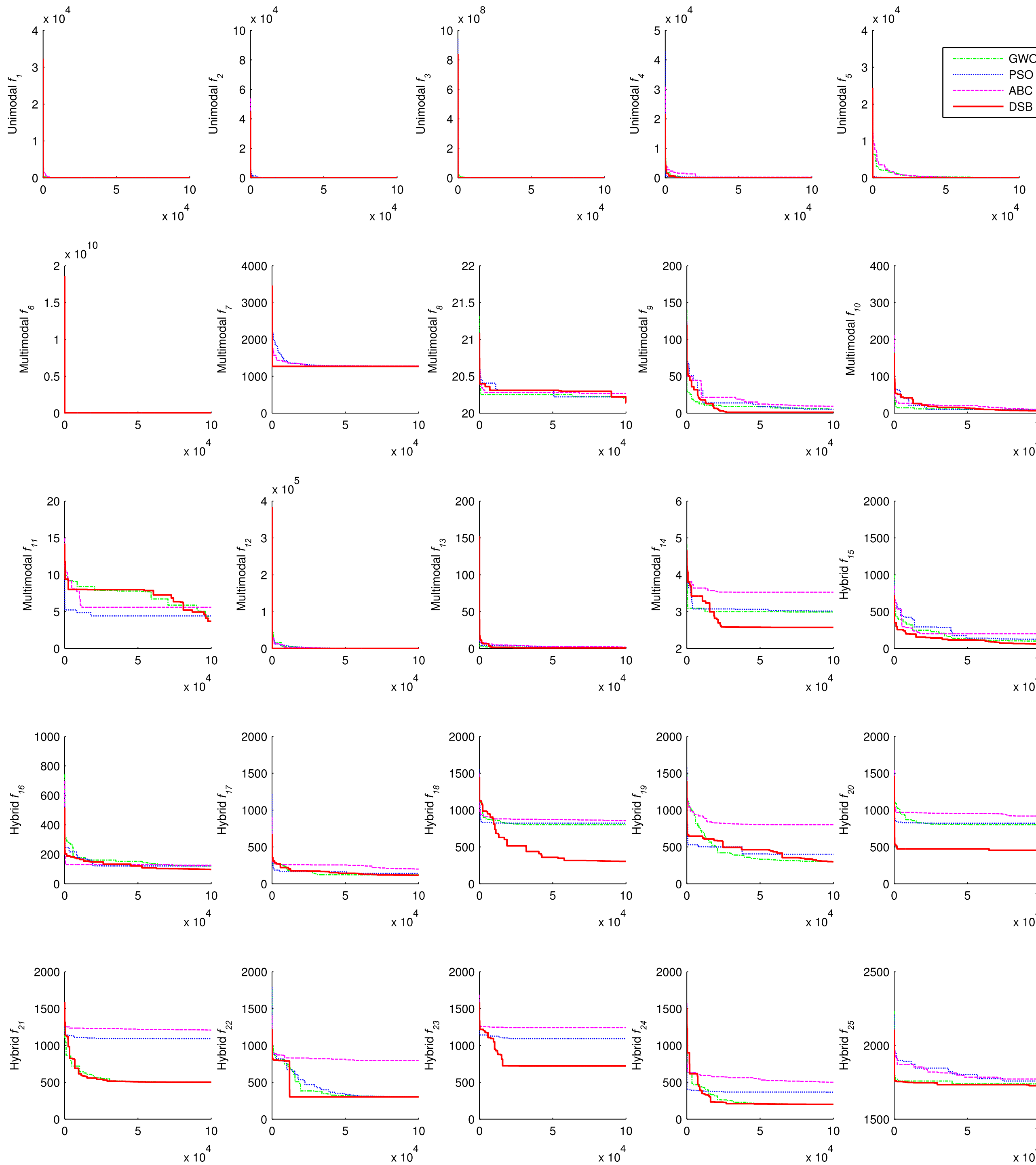

Fig. 4. Comparative convergence plot.

result testifies the better trade-off between exploration and exploitation in the DSB algorithm.

In order to evaluate the efficiency of the DSB algorithm over GWO, PSO and ABC algorithms, the method stated in Suganthan et al. (2005) is employed to analyze the computational complexity of the compared algorithms. Function f_3 is used as the benchmark evaluation function for the testing methodology suggested by Suganthan et al. (2005). The 10-dimension complexity analysis results for DSB, GWO, PSO and ABC are 85.06, 95.11, 83.8 and 84.88 respectively, whereas the 30-dimension complexity analysis results for DSB, GWO, PSO and ABC are 317.56, 320.35, 316.55 and 317.13. A smaller complexity value means that the algorithm is less complex and leads to a relatively faster execution speed under the same condition. From the results, GWO seems to be the most complicated among all the benchmarked algorithms. Although DSB is slightly more complicated than PSO and ABC, their complexities are not far apart. Nevertheless, DSB managed to display better results in large size problems compared to GWO, PSO and ABC.

From Fig. 4, Tables 2–5, several important observations can be noted:

- In terms of statistical test, DSB had outperformed all the compared algorithms. Among all the 25 functions, DSB generated the best simulations (largest difference) results in functions 14, 18, 20 and 23 compared with GWO, PSO and ABC respectively. Even

Table 2
Unimodal simulation results for 10-D problems.

Func	Algorithm	Mean	Median	Best	Std	p	h	stats.ranksum
1	GWO	3.58E+01	2.77E+01	3.00E−06	**1.04E+02**	0	1	1.50E+10
	PSO	7.63E+00	1.92E−02	1.36E−03	1.78E+02	0	1	1.50E+10
	ABC	5.28E+01	4.79E−02	1.93E−03	4.21E+02	0	1	1.50E+10
	DSB	**2.47E+00**	**0.00E+00**	**0.00E+00**	2.21E+02	0	0	0.00E+00
2	GWO	1.36E+02	1.15E+02	4.98E+01	2.51E+02	0	1	1.49E+10
	PSO	3.53E+01	3.46E−02	1.34E−03	**1.99E+02**	0	1	1.43E+10
	ABC	9.66E+01	4.48E−02	2.24E−03	5.45E+02	0	1	1.43E+10
	DSB	**1.82E+01**	**0.00E+00**	**0.00E+00**	4.28E+02	0	0	0.00E+00
3	GWO	6.53E+05	1.24E+05	3.71E+04	3.25E+06	0	1	1.49E+10
	PSO	1.50E+05	1.08E+05	8.23E+04	3.34E+06	0	1	1.49E+10
	ABC	5.15E+05	3.16E+05	3.16E+05	**1.52E+06**	0	1	1.50E+10
	DSB	**1.01E+05**	**1.37E+04**	**1.04E+04**	5.94E+06	0	0	0.00E+00
4	GWO	1.43E+02	7.15E−02	3.61E−03	7.34E+02	0	1	1.06E+10
	PSO	1.52E+02	1.24E+02	5.61E+01	**2.78E+02**	0	1	1.42E+10
	ABC	5.59E+02	2.04E+02	2.04E+02	8.33E+02	0	1	1.44E+10
	DSB	**9.70E+01**	**3.23E−02**	**2.17E−03**	6.23E+02	0	0	0.00E+00
5	GWO	6.20E+02	2.62E+02	1.62E+01	1.11E+03	0	1	1.48E+10
	PSO	8.66E+01	6.05E+01	6.05E+01	3.48E+02	0	1	1.48E+10
	ABC	8.02E+02	1.78E+02	6.22E+01	1.68E+03	0	1	1.49E+10
	DSB	**2.13E+01**	**7.50E+00**	**7.38E+00**	**3.11E+02**	0	0	0.00E+00

Table 3
Multimodal simulation results for 10-D problems.

Func	Algorithm	Mean	Median	Best	Std	p	h	stats.ranksum
6	GWO	4.64E+05	7.56E+00	1.55E+00	2.75E+07	0	1	1.50E+10
	PSO	**6.81E+04**	1.37E+02	9.74E+01	**3.55E+05**	0	1	1.50E+10
	ABC	1.46E+05	2.27E+04	5.59E+02	2.29E+07	0	1	1.50E+10
	DSB	7.10E+05	**1.35E−01**	**1.16E−01**	9.07E+07	0	0	0.00E+00
7	GWO	**1.27E+03**	1.27E+03	1.27E+03	**9.17E+00**	0	1	1.50E+10
	PSO	1.33E+03	1.28E+03	1.27E+03	1.51E+02	0	1	1.50E+10
	ABC	1.31E+03	1.28E+03	1.28E+03	7.56E+01	0	1	1.50E+10
	DSB	1.27E+03	**1.27E+03**	**1.27E+03**	1.64E+01	0	0	0.00E+00
8	GWO	**2.02E+01**	**2.03E+01**	2.02E+01	2.94E−02	0	1	6.05E+09
	PSO	2.03E+01	2.03E+01	2.02E+01	6.81E−02	0	1	7.99E+09
	ABC	2.03E+01	2.03E+01	2.03E+01	**2.76E−02**	0	1	6.30E+09
	DSB	2.03E+01	2.03E+01	**2.01E+01**	3.80E−02	0	0	0.00E+00
9	GWO	9.20E+00	9.00E+00	5.13E+00	**4.35E+00**	0	1	1.32E+10
	PSO	1.36E+01	8.71E+00	5.21E+00	1.25E+01	0	1	1.35E+10
	ABC	1.76E+01	1.22E+01	9.19E+00	1.09E+01	0	1	1.37E+10
	DSB	**5.89E+00**	**9.95E−01**	**9.95E−01**	1.15E+01	0	0	0.00E+00
10	GWO	**1.09E+01**	**9.14E+00**	9.14E+00	**4.65E+00**	0	1	8.89E+09
	PSO	1.53E+01	9.95E+00	9.95E+00	1.29E+01	3.47E−21	1	9.88E+09
	ABC	1.90E+01	2.02E+01	9.99E+00	6.23E+00	0	1	1.18E+10
	DSB	1.72E+01	1.35E+01	**6.40E+00**	1.33E+01	0	0	0.00E+00
11	GWO	7.20E+00	7.79E+00	3.69E+00	1.32E+00	0	1	9.38E+09
	PSO	**4.54E+00**	**4.42E+00**	4.42E+00	**2.83E−01**	0	1	5.52E+09
	ABC	5.91E+00	5.58E+00	5.58E+00	1.04E+00	0	1	7.60E+09
	DSB	7.20E+00	7.98E+00	**3.69E+00**	1.32E+00	0	0	0.00E+00
12	GWO	2.08E+03	1.88E+01	1.88E+01	6.72E+03	0	1	1.19E+10
	PSO	2.23E+03	2.73E+02	2.09E+01	5.03E+03	0	1	1.46E+10
	ABC	1.91E+03	3.05E+02	3.34E+01	4.77E+03	0	1	1.47E+10
	DSB	**6.21E+01**	**1.27E+01**	**1.27E+01**	**2.51E+03**	0	0	0.00E+00
13	GWO	**1.30E+00**	9.28E−01	7.43E−01	**6.33E−01**	0	1	1.06E+10
	PSO	2.81E+00	2.23E+00	1.12E+00	1.17E+00	0	1	1.42E+10
	ABC	3.50E+00	2.82E+00	1.25E+00	1.44E+00	0	1	1.43E+10
	DSB	1.41E+00	**7.43E−01**	**5.95E−01**	2.34E+00	0	0	0.00E+00
14	GWO	3.01E+00	3.00E+00	2.99E+00	7.49E−02	0	1	1.33E+10
	PSO	3.07E+00	3.06E+00	3.02E+00	1.17E−01	0	1	1.35E+10
	ABC	3.55E+00	3.53E+00	3.53E+00	**7.31E−02**	0	1	1.47E+10
	DSB	**2.73E+00**	**2.58E+00**	**2.57E+00**	3.34E−01	0	0	0.00E+00
15	GWO	1.77E+02	1.31E+02	1.01E+02	9.72E+01	0	1	1.16E+10
	PSO	2.24E+02	1.41E+02	1.24E+02	1.24E+02	0	1	1.31E+10
	ABC	2.27E+02	2.00E+02	2.00E+02	8.38E+01	0	1	1.43E+10
	DSB	**1.23E+02**	**1.14E+02**	**6.08E+01**	**5.92E+01**	0	0	0.00E+00

though certain functions may seem to produce similar results, the advantage is quite apparent: DSB still performs better than GWO, PSO and ABC in functions 1, 2, 3, 4, 5, 16, 17, 21, 24 and 25 respectively. In functions 8, 10, 11 and 19 even though DSB convergence rate is slower, nevertheless DSB algorithm still managed to produce the best results before hitting the termination criteria.

- In the first group of benchmark functions $f_1 - f_5$, DSB has proved to the best performing algorithm in comparison to GWO, PSO and ABC respectively. Even though from Fig. 4, it may seem

Table 4
Rotated Multimodal Simulation Results for 10-D problems.

Func	Algorithm	Mean	Median	Best	Std	*p*	*h*	stats.ranksum
16	GWO	1.52E+02	1.52E+02	1.21E+02	3.75E+01	0	1	1.24E+10
	PSO	1.36E+02	1.21E+02	1.21E+02	3.25E+01	0	1	1.12E+10
	ABC	1.30E+02	1.31E+02	1.26E+02	**1.19E+01**	0	1	1.05E+10
	DSB	**1.26E+02**	**1.21E+02**	**9.71E+01**	2.77E+01	0	0	0.00E+00
17	GWO	**1.46E+02**	**1.22E+02**	1.22E+02	4.46E+01	2.12E−273	1	9.54E+09
	PSO	1.54E+02	1.62E+02	1.40E+02	**2.04E+01**	5.59E−99	1	1.03E+10
	ABC	2.41E+02	2.55E+02	2.00E+02	2.40E+01	0	1	1.43E+10
	DSB	1.56E+02	1.42E+02	**1.16E+02**	4.42E+01	0	0	0.00E+00
18	GWO	8.19E+02	8.03E+02	8.01E+02	4.86E+01	0	1	1.40E+10
	PSO	8.26E+02	8.23E+02	8.23E+02	1.76E+01	0	1	1.40E+10
	ABC	8.75E+02	8.73E+02	8.56E+02	**1.75E+01**	0	1	1.41E+10
	DSB	**4.55E+02**	**3.59E+02**	**3.03E+02**	2.07E+02	0	0	0.00E+00
19	GWO	**4.25E+02**	**3.32E+02**	3.01E+02	1.93E+02	0	1	8.29E+09
	PSO	4.44E+02	4.05E+02	4.01E+02	6.17E+01	0.016713429	1	1.00E+10
	ABC	8.28E+02	8.03E+02	8.01E+02	**5.90E+01**	0	1	1.50E+10
	DSB	4.58E+02	4.63E+02	**3.00E+02**	1.14E+02	0	0	0.00E+00
20	GWO	8.25E+02	8.04E+02	8.01E+02	5.77E+01	0	1	1.50E+10
	PSO	8.25E+02	8.22E+02	8.22E+02	**1.22E+01**	0	1	1.50E+10
	ABC	9.49E+02	9.53E+02	9.18E+02	1.94E+01	0	1	1.50E+10
	DSB	**4.69E+02**	**4.73E+02**	**4.54E+02**	2.96E+01	0	0	0.00E+00

Table 5
Hybrid Multimodal Simulation Results for 10-D problems

Func	Algorithm	Mean	Median	Best	Std	*p*	*h*	stats.ranksum
21	GWO	5.54E+02	5.11E+02	5.01E+02	1.01E+02	0	1	1.05E+10
	PSO	1.10E+03	1.09E+03	1.09E+03	1.19E+01	0	1	1.48E+10
	ABC	1.22E+03	1.22E+03	1.21E+03	**1.06E+01**	0	1	1.49E+10
	DSB	**5.52E+02**	**5.07E+02**	**5.00E+02**	1.26E+02	0	0	0.00E+00
22	GWO	4.04E+02	3.23E+02	3.02E+02	1.71E+02	0	1	1.39E+10
	PSO	4.26E+02	3.45E+02	3.03E+02	1.70E+02	0	1	1.39E+10
	ABC	8.14E+02	8.13E+02	7.94E+02	**2.55E+01**	0	1	1.46E+10
	DSB	**3.61E+02**	**3.02E+02**	**3.02E+02**	1.62E+02	0	0	0.00E+00
23	GWO	7.76E+02	7.21E+02	7.21E+02	1.36E+02	0	1	1.44E+10
	PSO	1.10E+03	1.09E+03	1.09E+03	1.51E+01	0	1	1.49E+10
	ABC	1.24E+03	1.24E+03	1.24E+03	**4.40E+00**	0	1	1.49E+10
	DSB	**5.96E+02**	**5.59E+02**	**5.59E+02**	1.09E+02	0	0	0.00E+00
24	GWO	2.66E+02	**2.15E+02**	2.00E+02	1.23E+02	0	1	1.06E+10
	PSO	3.72E+02	3.68E+02	3.66E+02	**1.02E+01**	0	1	1.40E+10
	ABC	5.49E+02	5.45E+02	5.00E+02	4.23E+01	0	1	1.43E+10
	DSB	**2.58E+02**	2.06E+02	**2.00E+02**	1.43E+02	0	0	0.00E+00
25	GWO	1.75E+03	1.74E+03	1.74E+03	1.16E+01	0	1	1.32E+10
	PSO	1.81E+03	1.80E+03	1.76E+03	4.90E+01	0	1	1.50E+10
	ABC	1.81E+03	1.78E+03	1.77E+03	3.82E+01	0	1	1.50E+10
	DSB	**1.74E+03**	**1.73E+03**	**1.73E+03**	**8.27E+00**	0	0	0.00E+00

that all the compared algorithms have the achieved the same performance in $f_1 - f_3$, a close inspection in Table 2 would clearly indicate that DSB has outperformed the other algorithms. The results indicate that DSB has a slightly better convergence speed in solving unimodal optimization problem in comparison with GWO, PSO and ABC. As mentioned before, unimodal functions are suitable for benchmarking exploitation. The slightly better results indicate that the exploitation operator discussed previously managed to steer the population towards the optimum faster guided by the weighted mean of the population.

- Even when it comes to solving Group II multimodal optimization problems, DSB has exhibited superior performance in comparison with other algorithms. Multimodal functions have many local optima and makes them suitable for benchmarking the exploration ability of an algorithm. Even though the convergence speed of DSB in $f_7 - f_{11}$ and f_{14} is not as fast as the other algorithms, DSB still managed to obtain the best minimum result. It should be noted that DSB operators perform both the exploration and exploitation simultaneously during the optimization process. Exploitation leads to faster convergence but the exploration obstructs the movements of individual into smaller region of the search space hence affecting the convergence speed. This is because exploration operators are applied to every individuals whereas exploitation operators are applied to weaker individuals. In other words, the DSB algorithm is slightly tweaked to exhibit the exploration nature more compared to the exploitation nature. Nevertheless, the result indicates that DSB managed to strike the required balance between exploitation and exploration to tackle multimodal functions. In f_6, f_{12} *and* f_{13}, the performance of DSB is equally comparable with the other algorithms in terms of exploration capabilities.
- DSB is very competitive in hybrid rotated multimodal functions. These hybrid functions are suitable to benchmark both exploration and exploitation functions respectively, especially to evaluate the ability of an algorithm to avoid local minima. The convergence plot clearly displays the benefit of having a searching process with combined exploration and exploitation operator. Take f_{18}, f_{20} *and* f_{23} for an example. DSB managed to escape from the local optima and achieved superior result by the end of searching whereas the other algorithms are trapped in the local optima within the searching process. The exploration operators which employs attraction, repulsion and random movements allows a better individual distribution in the search space which increases the ability of DSB to find

global optima. The coordination of the movements are based on incorporated recruitment signals which holds the attraction and repulsion vector over the local best individual and the global best individual seen so far. This scheme had certainly allowed the individuals to explore their own neighborhood before converging towards the global optimum.

Interestingly, the No-Free Lunch (NFL) Theorem (Wolpert and Macready, 1997) mentions that it is theoretically impossible to have a single best universal optimization solver as all the meta-heuristic algorithms perform exactly the same when all possible evaluated benchmark functions are averaged. However, this creates endless possibility to develop new algorithms as the total number of possible problems is too huge. Hence, it is always worthwhile to venture into new methodologies with superior performance. This is the motivation to propose DSB as a global numerical optimization problem solver.

5.2. Optimization of hyper beamforming radiation pattern

This section discusses the experimental results for various optimized hyperbeam antenna array designs obtained by GWO, PSO, ABC and DSB optimization algorithms. Three sets of linear antenna array configurations are chosen for each one of the algorithm to optimize the nonuniform current excitation weights, optimal uniform inter-element spacing and beam-width control coefficient. Fig. 5 displays the optimized nonuniform excitations and optimized fixed inter-element spacing radiation patterns of linear antenna arrays for sets of 10, 14, and 20 number of elements with the exponent values, $\mu = 0.5$ and $\mu = 1$ respectively, as obtained by the algorithms. From Fig. 5, it is obvious that optimized hyper beam produces an enhanced SLL and FNBW compared to conventional hyper beam. Tables 6 and 7 lists the optimal current excitation weights, beam-width control coefficient, optimized uniform inter-element spacing ($\lambda \in [\lambda/2, \lambda]$), SLL and FNBW with hyperbeam exponent value, $\mu = 0.5$ and $\mu = 1$ respectively, using GWO, PSO, ABC and DSB.

The following observations can be made from Fig. 5, Tables 6 and 7 in which the exponent values are $\mu = 0.5$ and $\mu = 1$ respectively. Overall, DSB managed to yield the lowest SLL which is obviously noticeable and produce a considerably narrow beam width compared to PSO, ABC and GWO. The performance of PSO, ABC and GWO are almost on par with GWO topping the list but DSB had out performed the benchmarked algorithms with a remarkable feat. This testifies that the exploration and exploitation operators employed by DSB not only managed to solve linear problems but non-linear problems such as hyper beam optimization. While DSB produced the lowest SLL in all scenarios, the FNBW for 10-element antenna array for both exponent values, $\mu = 0.5$ and $\mu = 1$ were not the best. A slight trade off to achieve lower SLL. Nevertheless, the FNBW were at optimal level for 14 and 20-element antenna arrays. Another observation from the tabulated data is that the radiation patterns with $\mu = 0.5$ produces lower SLL and narrower FNBW compared to radiation patterns with $\mu = 1$. Hence, the exponent value, $\mu = 0.5$ would be ideal for achieving a narrower FNBW with low SLL for any future research reference.

5.3. Discussion about the DSB algorithm

Even though DSB algorithm falls under the category of swarm intelligence algorithm, there are several differences that distinguishes DSB from PSO, ABC and GWO. Similar to DSB, all those algorithms were also inspired by collective social behavior of animals and were initially developed to solve continuous optimization problems. The first difference is the individual movements in the search space. In PSO, the movement of each particle is driven towards the global best position and also towards their own personal best positions. In ABC, there is no mechanism to use the global information in the search space, so the individual bee's movements easily results in a waste of computing power and gets trapped in local optima as the mechanism only encourages local interaction. Similarly, GWO also updates the positions of its search agents based on the locations of local neighboring wolves which highly encourages exploration and encounters the issue as ABC algorithm. Whereas in DSB, the members perform positional shift based on their own historical positions (local memory), their neighboring members' current positions and the global best position. These patterns lead to a better searching behaviors due to the fact that the individual current positions and the global best position differs greatly during most of the iteration process. Therefore, this type of search patterns can be more efficient in solving multi-modal optimization problems with a large number of local minima. Another point to note is that DSB is not sensitive to parameter tuning as all the members are simultaneously governed by the exploration and exploitation operators. This means that there is no need to define the parameter values to control the extend of exploration and exploitation affects the optimization process. On the contrary, PSO and GWO requires careful parameters tuning to ensure a smooth transition between exploration and exploitation otherwise the search results may not lead to the optimal result. The intricate behavior of the agents in PSO and GWO is found to be dependent on the settings of the different algorithm parameters. Thus, the interdependence of the different parameters makes these algorithms sensitive to proper parameter tuning as one set of parameters may not be the best fit for all problems to guarantee optimum results.

Subsequently, the next difference lies in their optimization design metaphor. The PSO was inspired by the principles of how animal groups such as flocks of birds or schools of fishes coordinate their motions. ABC adopts the food foraging model of bees in a colony whereas GWO utilizes Producer–Scrounger (PS) model where the individuals are divided into leaders and followers. On the other hand, DSB utilizes communal social behavior as its design metaphor which means that individuals perform search operation while communicating with other individuals to look for better solution potentials. One of the most prominent difference between DSB and other proposed algorithms is that DSB employs Information Sharing (IS) model into all its search agents to enhance the general social animal searching behavior. IS model is also present in ABC and GWO but limited to certain groups of individuals only which means that crucial piece of information could be lost leading to sub-optimal search performance.

Another obvious difference between the conventional PSO, ABC and GWO algorithms and DSB is in the method of information transfer. In the conventional PSO algorithm, the method of information transfer is not applied and it is assumed that all the particles are aware of the system information without loss. Nevertheless, in recent times several variant of PSO, conventional ABC and GWO have considered the validity of information but not on the information loss characteristics. The unique feature of DSB is that the information transfer method is modeled through the acoustic based recruitment signal which considers a generic knowledge system with information loss. Up to date, there is still no research on the impact of information loss with regards to optimization technique based on social behavior strategy. Hence, this opens up new opportunity for exploration in the near future.

Knowledge sharing is another feature which differentiates PSO, ABC, GWO and DSB. In DSB, every member generates new information and transmits the information to the whole population whereas in PSO, ABC and GWO, the algorithm does not contain the shared information of the entire population. The common information is focused on the best particle in the system in the case of PSO and local best individuals in the case of ABC and GWO. In DSB, the information transfer is derived from the current position of individuals, neighboring individuals and the best individual position instead of purely relying on the best historical positions unlike PSO or best neighboring positions unlike ABC and GWO, and this varies the searching process of DSB.

Finally, even though DSB, ABC and GWO are all inspired by the social animal foraging strategy, there are some obvious differences. In ABC and GWO, the populations are divided into several groups that

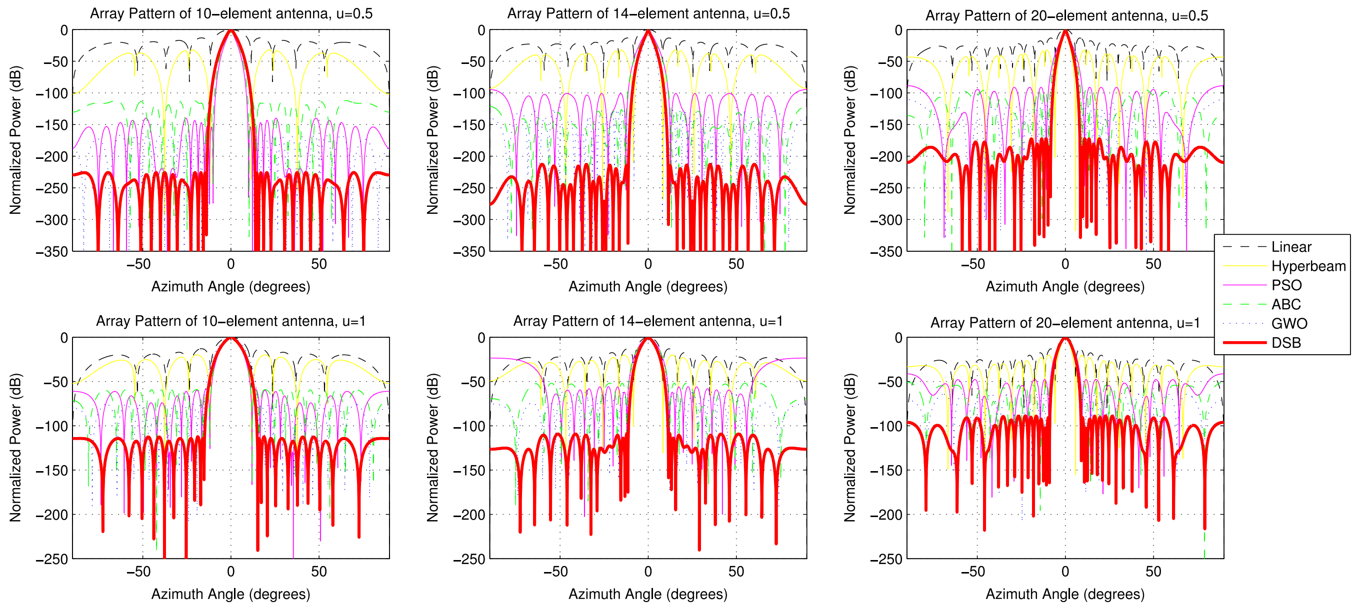

Fig. 5. Comparison of optimized array pattern for 10, 14 and 20-element array with $u = 0.5$ and 1 respectively with improved SLL.

Table 6
SLL, FNBW, optimal current excitation weights and optimal inter-element spacing for hyper beam pattern of linear array with hyper beam exponent ($u = 0.5$), obtained by PSO, ABC, GWO and DSB for different sets of arrays.

N	Algorithms	Optimized current excitation weights $[I_1, I_2, I_3, \ldots, I_N]$					Beam width control coefficient	Optimal inter-element spacing (λ)	SLL of hyper beam with optimization (dB)	FNBW of hyper beam with optimization (deg)
10	PSO	0.2219	0.5435	0.9516	0.9032	0.6932	0.8690	0.9768	−140	20.12
		1.5105	0.7287	0.2529	0.7488	0.3500				
	ABC	0.1773	0.5184	1.0292	1.0589	1.0654	0.9600	0.8259	−112.3	20.8
		1.0576	0.7828	0.4202	0.5760	0.3943				
	GWO	0.1028	0.3814	0.6264	0.9474	0.8024	0.9700	0.8455	−166.7	26.08
		1.0425	0.5369	0.5555	0.2123	0.2135				
	DSB	0.1790	0.3917	0.3183	0.6822	0.9713	0.8425	0.9230	−225.7	27.76
		0.9858	0.9874	0.9030	0.3707	0.1317				
14	PSO	0.3658	0.2561	0.4931	0.0043	0.6128	0.7134	0.5947	−100.8	25.04
		0.5837	0.8801	0.5165	0.5880	0.6064				
		0.7424	0.5612	0.2889	0.3275					
	ABC	0.2236	0.1858	0.6027	0.5261	0.8026	0.6800	0.6161	−128.9	26.54
		0.4233	0.6275	0.7401	0.9499	0.2960				
		0.2525	0.2800	0.5037	0.3656					
	GWO	0.2297	0.3706	0.3107	0.2272	0.6682	1.0006	0.7953	−127.4	15.88
		0.9515	0.6941	0.8598	0.4166	0.7559				
		0.7312	0.2396	0.3754	0.0982					
	DSB	0.1301	0.3198	0.4095	0.2993	0.5739	0.7003	0.9892	−212.8	22.78
		0.9710	0.9424	0.9167	0.7404	0.8678				
		0.7414	0.3445	0.2248	0.0591					
20	PSO	0.3201	0.4118	0.4894	0.4673	0.3027	0.6383	0.5381	−91.24	21.48
		0.6716	0.3586	0.9886	0.6950	0.9040				
		0.5547	0.9970	0.4059	0.7208	1.0449				
		0.1579	0.8253	0.3337	0.0020	0.5574				
	ABC	0.1675	0.2460	0.2113	0.5307	0.6119	0.7000	0.5353	−95.8	18.92
		0.5740	0.7962	0.2236	0.8399	0.2515				
		1.0186	0.3767	0.8215	0.1836	0.2281				
		0.1792	0.4317	0.6579	0.2191	0.3467				
	GWO	0.1585	0.1384	0.5700	0.4827	0.9529	0.7056	0.5989	−115.6	18.16
		0.7749	0.9684	0.7941	0.6995	0.6093				
		1.0250	0.9416	0.4721	0.4252	0.2891				
		0.3321	0.4208	0.4571	0.4279	0.3235				
	DSB	0.0006	0.0001	0.0062	0.2111	0.3791	0.7001	0.9549	−172.6	16.82
		0.5922	0.7999	0.8638	0.8769	0.9311				
		0.9476	0.9212	0.7571	0.5878	0.5321				
		0.4865	0.3985	0.3611	0.1815	0.0609				

perform specific tasks while collectively searching the search space. However in DSB, all members equally perform all the tasks that would be executed by multiple types of populations in the other two algorithms. Therefore, the uniform structure of the population in DSB with multiple execution tasks may enhance the searching process in many multimodal optimization problems. In a nutshell, the underlying social behavior strategy coupled with IS foraging model leads to the unique searching pattern and enhanced performance of DSB over other algorithms.

Table 7
SLL, FNBW, optimal current excitation weights and optimal inter-element spacing for hyper beam pattern of linear array with hyper beam exponent ($u = 1$), obtained by PSO, ABC, GWO and DSB for different sets of arrays.

N	Algorithms	Optimized current excitation weights $[I_1, I_2, I_3, \ldots, I_N]$					Beam width control coefficient	Optimal inter-element spacing (λ)	SLL of hyper beam with optimization (dB)	FNBW of hyper beam with optimization (deg)
10	PSO	0.1460	0.1010	0.4741	0.4133	0.6267	1.0558	0.7059	−58.66	25.04
		0.6457	0.7009	0.4038	0.3935	0.2140				
	ABC	0.3811	0.4379	0.2107	0.7478	0.8039	0.9865	0.6749	−59.11	24.12
		0.4424	0.3736	0.5041	0.1774	0.1603				
	GWO	0.2606	0.5162	1.0342	0.9984	1.0340	0.7301	1.0454	−76.15	31.24
		0.9947	0.6257	0.0164	0.0724	0.0343				
	DSB	0.1090	0.3320	0.8501	0.9670	0.9875	0.9817	0.7364	−112.6	30.22
		0.9790	0.7005	0.3502	0.3917	0.1729				
14	PSO	0.0415	0.4310	0.6027	0.8172	0.8079	0.7264	0.6426	−55.81	28.08
		0.8317	0.4347	1.1156	0.2403	0.3684				
		0.4689	0.0313	0.6110	0.1934					
	ABC	0.1003	0.2575	0.3051	0.5360	0.6398	0.7008	0.6728	−52.75	24.92
		0.7006	0.5262	0.8292	0.3861	0.4833				
		0.3820	0.4812	0.1342	0.3375					
	GWO	0.1808	0.4093	0.3065	0.3611	0.5661	0.7008	0.7419	−62.18	24.06
		0.8375	0.6791	0.7061	0.4670	0.6386				
		0.5091	0.3592	0.0831	0.1959					
	DSB	0.1091	0.3043	0.4322	0.3514	0.5521	0.7000	0.9919	−109.3	22.78
		0.9394	0.9737	0.9810	0.8525	0.9474				
		0.7552	0.3591	0.1935	0.0481					
20	PSO	0.2739	0.0772	0.4660	0.3213	0.4341	0.7215	0.5624	−47.81	18.62
		0.6219	0.6238	0.8594	0.4543	0.7166				
		0.6867	0.9221	0.3948	0.5011	0.4219				
		0.6935	0.3145	0.3454	0.1700	0.4868				
	ABC	0.5928	0.0927	0.4036	0.0208	0.6520	0.7029	0.6014	−54.1	17.98
		0.1704	0.7801	0.0025	0.8693	0.8679				
		0.6607	0.7677	0.9863	1.0091	0.7817				
		0.9002	0.5228	0.7505	0.2867	0.2434				
	GWO	0.0947	0.1492	0.2643	0.5640	0.6044	0.7000	0.6599	−63.6	17.8
		0.8503	0.8923	0.7102	0.6176	0.7435				
		0.8187	0.9004	0.7045	0.3612	0.2097				
		0.3706	0.3647	0.5066	0.3815	0.1691				
	DSB	0.1272	0.2162	0.4029	0.4432	0.5117	0.4501	0.9636	−89.02	18.08
		0.5840	0.6999	0.9549	0.8148	0.9630				
		0.9240	0.9804	0.6897	0.6667	0.5921				
		0.3478	0.1447	0.0000	0.0004	0.0001				

6. Conclusions

This work proposes a novel Dynamic Social Behavior based on social communal interaction rules to solve global optimization problems. The social communal interaction rules are integrated with information transfer strategy (recruitment signal) which resembles collective social behavior. The DSB algorithm has a relatively simple structure and does not require more than one parameter setting which makes it easier to implement.

A set of 25 benchmark functions were employed to evaluate the performance of the proposed algorithm in terms of convergence rate, local optima avoidance, exploration and exploitation. The simulation results and the tabulated data showed that DSB was able to provide highly competitive results compared to popular algorithms such as GWO, PSO and ABC in all the four different groups of functions. Furthermore, the proposed algorithm was evaluated by solving a real engineering problem on hyper beamforming optimization. The experimental results reveal that DSB produces optimal designs, providing the best reduction in SLL and improved FNBW as compared to other benchmark algorithms.

The remarkable performance of DSB is contributed by several factors such as the search operator factor and the population division factor. The search operator ensures that the population gets a better distribution in the search space, thus increasing the probability to locate the global optima or in other words, increase its exploration capability. On the other hand, the population division divides the population into two individual types and employs the median search approach to enhance the exploitation ability during the optimization process.

For future work, DSB will be evaluated for CEC 2006 real world test problems to verify its efficiency in solving other challenging search spaces.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Agarwalla, Prativa, Mukhopadhyay, Sumitra, 2017. Efficient player selection strategy based diversified particle swarm optimization algorithm for global optimization. Inform. Sci. 397, 69–90.

Alatas, Bilal, 2011. ACROA: Artificial chemical reaction optimization algorithm for global optimization. Expert Syst. Appl. (ISSN: 0957-4174) 38 (10), 13170–13180, URL http://dx.doi.org/10.1016/j.eswa.2011.04.126.

Alexandridis, Alex, Chondrodima, Eva, Sarimveis, Haralambos, 2016. Cooperative learning for radial basis function networks using particle swarm optimization. Appl. Soft Comput. 49, 485–497.

Anitha, V., Sri Jaya Lakshmi, S., Sreedevi, I., Khan, Habibulla, Sarat Kumar, K., Ramakrishna, P., 2012. An adaptive processing of linear array for target detection improvement. Int. J. Comput. Appl. (0975–8887) 42 (4), 33–36.

Balanis, Constantine A., 2012. Antenna Theory: Analysis and Design. John Wiley & Sons.

Banharnsakun, Anan, Achalakul, Tiranee, Sirinaovakul, Booncharoen, 2011. The best-so-far selection in artificial bee colony algorithm. Appl. Soft Comput. (ISSN: 1568-4946) 11 (2), 2888–2901, URL http://dx.doi.org/10.1016/j.asoc.2010.11.025.

Barnard, C.J., Sibly, R.M., 1981. Producers and scroungers: A general model and its application to captive flocks of house sparrows. Anim. Behav. 29 (2), 543–550, URL http://dx.doi.org/10.1016/s0003-3472(81)80117-0.

Beyer, Hans-Georg, Schwefel, Hans-Paul, 2002. Evolution strategies –A comprehensive introduction. Nat. Comput. (ISSN: 1572-9796) 1 (1), 3–52, URL http://dx.doi.org/10.1023/A:1015059928466.

Clark, Colin W., Mangel, Marc, 1984. Foraging and flocking strategies: information in an uncertain environment. Am. Nat. 123 (5), 626–641, URL http://www.jstor.org/stable/2461242.

Couzin, Iain D., Krause, Jens, James, Richard, Ruxton, Graeme D., Franks, Nigel R., 2002. Collective memory and spatial sorting in animal groups. J. Theoret. Biol. 218 (1), 1–11.

Cuevas, Erik, Gonzalez, Mauricio, Zaldivar, Daniel, Perez-Cisneros, Marco, García, Guillermo, 2012. An algorithm for global optimization inspired by collective animal behavior. Discrete Dyn. Nat. Soc. 2012.

De, Arijit, Mamanduru, Vamsee Krishna Reddy, Gunasekaran, Angappa, Subramanian, Nachiappan, Tiwari, Manoj Kumar, 2016. Composite particle algorithm for sustainable integrated dynamic ship routing and scheduling optimization. Comput. Ind. Eng. 96, 201–215.

Dorigo, Marco, Stützle, Thomas, 2004. Ant Colony Optimization. Bradford Company, Scituate, MA, USA, ISBN: 0262042193.

Fong, Simon, Wang, Xi, Xu, Qiwen, Wong, Raymond, Fiaidhi, Jinan, Mohammed, Sabah, 2016. Recent advances in metaheuristic algorithms: Does the Makara dragon exist? J. Supercomput. 72 (10), 3764–3786.

Formato, Richard A., 2009. Central force optimization: A new deterministic gradient-like optimization metaheuristic. OPSEARCH (ISSN: 0975-0320) 46 (1), 25–51, URL http://dx.doi.org/10.1007/s12597-009-0003-4.

Goldberg, David E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, first ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, ISBN: 0201157675.

Hosseinabadi, Ali, Siar, Hajar, Shamshirband, Shahaboddin, Shojafar, Mohammad, Nasir, Mohd, 2015. Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in Small and Medium Enterprises. Ann. Oper. Res. 229 (1), 451–474, URL http://EconPapers.repec.org/RePEc:spr:annopr:v:229:y:2015:i:1:p:451-474:10.1007/s10479-014-1770-8.

Karaboga, Dervis, Basturk, Bahriye, 2007. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization. In: Problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing. IFSA (2007), Springer-Verlag, pp. 789–798.

Kaveh, Ali, Talatahari, Siamak, 2011. An enhanced charged system search for configuration optimization using the concept of fields of forces. Struct. Multidiscip. Optim. (ISSN: 1615-1488) 43 (3), 339–351, URL http://dx.doi.org/10.1007/s00158-010-0571-1.

Kennedy, James, Eberhart, Russell, 1995. Particle swarm optimization.

Koza, John R., 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA, ISBN: 0-262-11170-5.

Mirjalili, Seyedali, Mirjalili, Seyed Mohammad, Lewis, Andrew, 2014. Grey wolf optimizer. Adv. Eng. Softw. (ISSN: 0965-9978) 69, 46–61, URL http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

Mirjalili, Seyedali, Saremi, Shahrzad, Mirjalili, Seyed Mohammad, Coelho, Leandro dos S., 2016. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst. Appl. 47, 106–119.

Mittal, Nitin, Singh, Urvinder, Sohi, Balwinder Singh, 2016. Modified grey wolf optimizer for global engineering optimization. Appl. Comput. Intell. Soft Comput. 2016, 8.

Moghaddam, Fereydoun Farrahi, Moghaddam, Reza Farrahi, Cheriet, Mohamed, 2012. Curved space optimization: a random search based on general relativity theory, CoRR, abs/1208.2214, URL http://arxiv.org/abs/1208.2214.

Mucherino, Antonio, Seref, Onur, 2007. Monkey search: a novel metaheuristic search for global optimization. AIP Conf. Proc. 953 (1), 162–173. http://dx.doi.org/10.1063/1.2817338. URL http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.2817338.

Ram, Gopi, Mandal, Durbadal, Kar, Rajib, Ghoshal, Sakti Prasad, 2013. Optimized hyper beamforming of linear antenna arrays using collective animal behaviour. Sci. World J. 2013.

Rashedi, Esmat, Nezamabadi-pour, Hossein, Saryazdi, Saeid, 2009. GSA: a gravitational search algorithm. Inform. Sci. (ISSN: 0020-0255) 179 (13), 2232–2248, URL http://dx.doi.org/10.1016/j.ins.2009.03.004.

Shiqin, Y., Jianjun, J., Guangxing, Y., A dolphin partner optimization, In: 2009 WRI Global Congress on Intelligent Systems, volume 1, May 2009, pp. 124–128, URL http://dx.doi.org/10.1109/GCIS.2009.464.

Simon, D., 2008. Biogeography-based optimization. IEEE Trans. Evol. Comput. (ISSN: 1089-778X) 12 (6), 702–713. http://dx.doi.org/10.1109/TEVC.2008.919004.

Song, Xiaoyu, Yan, Qifeng, Zhao, Ming, 2017. An adaptive artificial bee colony algorithm based on objective function value information. Appl. Soft Comput. 55, 384–401.

Storn, Rainer, Price, Kenneth, 1997. Differential evolution –a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. (ISSN: 1573-2916) 11 (4), 341–359, URL http://dx.doi.org/10.1023/A:1008202821328.

Su, Weixing, Chen, Hanning, Liu, Fang, Lin, Na, Jing, Shikai, Liang, Xiaodan, Liu, Wei, 2017. A novel comprehensive learning artificial bee colony optimizer for dynamic optimization biological problems. Saudi J. Biol. Sci..

Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S., 2005. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical report, Nanyang Technological University, Singapore.

Sumpter, David J.T., 2010. Collective Animal Behavior. Princeton University Press.

Wang, Yu, Li, Bin, Weise, Thomas, Wang, Jianyu, Yuan, Bo, Tian, Qiongjie, 2011. Self-adaptive learning based particle swarm optimization. Inform. Sci. (ISSN: 0020-0255) 181 (20), 4515–4538, URL http://dx.doi.org/10.1016/j.ins.2010.07.013.

Wang, Hui, Sun, Hui, Li, Changhe, Rahnamayan, Shahryar, Pan, Jeng-Shyang, 2013. Diversity enhanced particle swarm optimization with neighborhood search. Inform. Sci. (ISSN: 0020-0255) 223, 119–135, URL http://dx.doi.org/10.1016/j.ins.2012.10.012.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. Trans. Evol. Compt. (ISSN: 1089-778X) 1 (1), 67–82, URL http://dx.doi.org/10.1109/4235.585893.

li Xiang, Wan, qing An, Mei, 2013. An efficient and robust artificial bee colony algorithm for numerical optimization. Comput. Oper. Res. (ISSN: 0305-0548) 40 (5), 1256–1265. http://dx.doi.org/10.1016/j.cor.2012.12.006. URL http://www.sciencedirect.com/science/article/pii/S0305054812002754.

Xiaohu, L., Jinhua, Z., Sunan, W., Maolin, L., Kunpeng, L., A small world algorithm for high-dimensional function optimization, In: Computational Intelligence in Robotics and Automation, CIRA, 2009, IEEE International Symposium on, Dec 2009, pp. 55–59, URL http://dx.doi.org/10.1109/CIRA.2009.5423233.

Yang, Xin-She, 2013. Cuckoo Search and Firefly Algorithm: Theory and Applications. Springer Publishing Company, Incorporated, ISBN: 9783319021409.

Yang, Xin-She, He, Xingshi, 2013. Bat algorithm: literature review and applications. Int. J. Bio-Inspired Comput. (ISSN: 1758-0366) 5 (3), 141–149, URL http://dx.doi.org/10.1504/IJBIC.2013.055093.

Yao, Xin, Liu, Yong, Lin, Guangming, 1999. Evolutionary programming made faster. IEEE Trans. Evol. Comput. (ISSN: 1089-778X) 3 (2), 82–102. http://dx.doi.org/10.1109/4235.771163.

Yu, James J.Q., Li, Victor O.K., 2015. A social spider algorithm for global optimization. Appl. Soft Comput. (ISSN: 1568-4946) 30 (C), 614–627, URL http://dx.doi.org/10.1016/j.asoc.2015.02.014.