Journal of Intelligent & Fuzzy Systems 37 (2019) 8063–8076
DOI:10.3233/JIFS-190495
IOS Press

Wildebeest herd optimization: A new global optimization algorithm inspired by wildebeest herding behaviour

D. Geraldine Bessie Amali[a,*] and M. Dinakaran[b]
[a]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India*
[b]*School of Information Technology, Vellore Institute of Technology, Vellore, India*

Abstract. This paper proposes a new metaheuristic global optimization algorithm inspired by Wildebeest herding behavior called Wildebeest Herd Optimization (WHO) algorithm. WHO algorithm mimics the way nomadic Wildebeest herds search vast areas of grasslands efficiently for regions of high food density. The WHO algorithm models five principal Wildebeest behaviors: firstly Wildebeests have limited eyesight and can only search for food locally, secondly Wildebeests stick to the herd to escape predators, thirdly Wildebeest herd as a whole migrates to regions of high food availability based on historical knowledge of annual grass growth rates and rainfall patterns, fourthly Wildebeests move out of crowded overgrazed regions and finally Wildebeests move to avoid starvation. The WHO algorithm is compared to Physics inspired, Swarm based, Biologically inspired and Evolution inspired global optimization algorithms on an extended test suite of benchmark optimization problems including rotated, shifted, noisy and high dimensional problems. Extensive simulation results indicate that the WHO algorithm proposed in this paper significantly outperforms state-of-the-art popular metaheuristic optimization algorithms like Particle Swarm Optimization Algorithm (PSO), Genetic Algorithm (GA), Gravitational Search Algorithm (GSA), Artificial Bee Colony Algorithm (ABC) and Simulated Annealing (SA) on shifted, high dimensional and large search range problems.

Keywords: Global optimization, Wildebeest Herd Optimization Algorithm, biologically inspired metaheuristic, heuristic optimization, heuristic search algorithm

1. Introduction

The need to find the global minimum of high-dimensional non-convex function arises in many real world applications involving engineering design, machine learning, optimal filter design, VLSI circuit optimization, bioinformatics and power system optimization [1]. In a majority of applications the cost function being minimized is highly multimodal and high dimensional involving a large number of independent variables. The exact solution to global optimization problems cannot be computed in polynomial time and involves an exhaustive search of all possible solutions. Thus the problem of finding the exact solution to global optimization problems is computationally intractable despite their ubiquitous appearance in applications. Thus heuristic based optimization algorithms that compute an approximate or acceptable solution to global optimization problems are of interest [16]. These heuristic algorithms do not always find the global minimum but usually succeed in locating a deep local minimum. In recent years heuristic global optimization algorithms inspired by Physics, Biology, Social, Swarm and Herd behavior have been proposed and applied successfully to a wide variety of economically important and challenging problems.

[*]Corresponding author. D. Geraldine Bessie Amali, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India. E-mail: geraldine.amali@vit.ac.in.

Heuristic global optimization algorithms attempt to perform a biased random search of the solution space with the search biased towards promising areas as identified by the heuristic used. The stochastic nature of heuristic based algorithms allow escape from local minima and enables exploration of large high dimensional spaces. Deterministic algorithms such as Nelder Mead Simplex [14], Gradient Descent and Quasi Newton are proven to get stuck in local minimum [16] and underperform on high dimensional problems [16]. In most deterministic algorithms, a sequence of points with decreasing function values is computed iteratively and hence these algorithms cannot escape from local minima. On the other hand stochastic algorithms such as Genetic Algorithm [3], Particle Swarm Optimization algorithm [19], Ant Colony Optimization algorithm [15] and Artificial Bee colony algorithm [4] perform a biased random search and hence have the ability to escape local minimum [12]. However even stochastic search algorithms are not guaranteed to escape from all local minima and exploration of new local minima must always be performed at the cost of exploitation of already known good solutions. If too much computational effort is expended in exploration of numerous local minima then the task of accurately determining the location of the local minima will be compromised [16]. Thus a fundamental and unavoidable tradeoff exists between exploration of new solutions and exploitation of already known good solutions. In most practical applications, accuracy of more than four decimal digits is seldom useful, the proposed WHO algorithm was tuned to limit exploitation of local minima to this accuracy to avoid wasting computational resources meant for exploration of new solutions.

A review of extant literature confirms that metaheuristics are mostly inspired by principles of physics [1], natural evolution [2] and collective animal behaviour [5]. A majority of metaheuristic optimization algorithms are derivative-free and hence can be applied to discontinuous and non-differentiable cost functions that frequently occur in practical applications [21]. Metaheuristics can be broadly classified into single agent based and multi-agent based depending upon the number of candidate solutions that are considered at a time [12]. In the case of a single-agent metaheuristic, a single solution is continuously improved until the solution stagnates. Single-agent metaheuristics are cheaper in terms of memory and computational resources as they do not keep track of multiple intermediate locations with high and low cost values. Since single agent metaheuristics do not consider multiple solutions simultaneously these approaches suffer from slow or premature convergence to local minima. Simulated annealing and Hill climbing are examples of single-agent metaheuristics which consider only one tentative solution at a time. Multi-agent based metaheuristics (also referred to as population based metaheuristics) on the other hand consider multiple solutions at the same time. Exchange of information between the search agents (members of the population) improves exploration and prevents premature convergence to the nearest local minimum without considering better solutions in distant regions on the search space.

Nature inspired multiagent metaheuristics can be broadly classified as evolutionary, swarm and herd behaviour based approaches. Evolutionary approaches are inspired by natural evolution processes which result in ever more complex and better adapted life forms [11]. Swarm and herd behavior based approaches are attempts to mimic collective intelligence, decentralized problem solving and emergent behaviour exhibited by large collections of animals. Large collections (insect swarms and animal herds) of interacting animals can exhibit surprisingly complex behaviour although composed of simple individuals with limited intelligence. In such collections, individuals act locally to secure their own selfish goals and often mindlessly copy the behaviour of their neighbors without any planning or global awareness [5]. However such collections due to the complexity of interactions between individuals often exhibit surprisingly complex problem solving abilities without any centralized coordination or planning. The terms swarm is usually applied to collections of simple organisms while the term herd is applied to collections of more complex animals which exhibit 'neighbor copy' behaviour [19]. Although numerous optimization metaheuristics inspired by swarm behaviour have been proposed only a small number of herd behaviour based metaheuristics have been proposed [8]. One such herd behavior based algorithm is krill herd optimization algorithm [9]. Models of herd behavior have demonstrated their usefulness in numerous domains from neuroscience to economics. Thus exploration of new herd behaviour based optimization metaheuristics with novel features is of interest. In this paper a new herd behaviour based metaheuristic based on Wildebeest herd foraging behaviour is proposed and shown to be competitive with swarm and evolutionary approaches.

Regardless of their origin, the performance of all global optimization algorithms is primarily determined by the trade-off made by the particular algorithm between exploration of new solutions and exploitation of known regions of the search space. Many popular algorithms suffer from poor exploration and exhibit premature convergence to local minima. Still other algorithms have an inbuilt predetermined bias for the origin ($x = (0, 0, ..., 0)$) and converge to a location near the origin even when the true global minimum is not located at the origin. Thus when testing the performance of different global optimization algorithm, care must be exercised to use test functions with global minima significantly shifted from$x = (0, 0, ..., 0)$. In many test suites only small random shifts in a small interval like [–2, 2] are considered. Since the average of these random shifts is zero in each coordinate, these small random shifts do not really differentiate between an algorithm that blindly converges to zero and an algorithm that actually finds the global minimum. The WHO algorithm proposed in this paper is shown to perform significantly better that state-of-the-art algorithms on high dimensional test functions with large shifts.

The remainder of this paper is organized as follows: A review of the literature on global optimization metaheuristics is presented in section 2, the Wildebeest herd behaviour and relationship to global optimization is discussed in section 3, a mathematical model of Wildebeest herd behaviour and a new global optimization behaviour based on Wildebeest behaviour is presented in 4 and finally the performance of the proposed WHO algorithm is compared to a wide variety of heuristic global optimization algorithms on challenging high dimensional, rotated and shifted benchmark problems.

2. Literature review

Metaheuristic algorithms offer several advantages over existing classical gradient based optimization approaches that converge to the nearest local minimum. If the initial position lies in the neighborhood of a local minimum then gradient based approaches cannot escape from that neighborhood. Some approaches also require the calculation of matrices such as Jacobians and Hessians which are computationally very expensive for high dimensions [21]. Metaheuristic approaches on the other hand are stochastic and perform a random search. This inherent nature of the metaheuristic algorithms help escape from local minimum and assist in exploring the multidimensional solution spaces. Popular metaheuristics algorithms developed are swarm inspired, evolution inspired or physics inspired.

Algorithms such as PSO, ACO, ABC, GSO and GWO models swarm behavior. The PSO algorithm copies the behavior of migratory flock of birds and school of fishes [19, 26]. The movement of the particles in the search space is controlled by the best position found by each particle and the best position found by the entire the swarm. These positions get updated every time better positions are identified thereby guiding the swarm to better solutions. Updating each particle's best solution contributes to the local search whereas the initial random distribution of the particles contributes to the exploration of the search space. PSO has very simple update formulae and does not require the cost problem to be differentiable. Since the PSO algorithm moves every particle towards a single global best solution, it can potentially converge to a local minimum [16].

A recently proposed biologically inspired optimizer is based on the hunting behaviour of grey wolves [20]. Grey wolf optimizer divides its population among four types of wolves namely alpha, beta, delta and omega. The best solution is termed as the alpha which tops the hierarchy. The second and third best become the beta and delta respectively. Searching for prey by the omegas are guided by the positions of the alpha, beta and gamma. They then converge towards the location of the alpha to carry out the attack. Diversification is provided during searching the prey and convergence happens during attack. Since the search of the prey is heavily influenced by the position of the alpha, the algorithm might converge to locally optimal solution.

Another popular group of metaheuristics draws inspiration from the Darwin's theory of evolution. The most widely used among them is the Genetic Algorithm (GA) [12]. An initial population of candidate solutions is randomly generated. Candidate solutions are improved by applying operators such as selection, crossover and mutation which model evolutionary processes in nature. Since fit individuals have a higher chance of being selected for the next generation the solutions get better with each generation. The selected individuals are then recombined and mutated to generate the next population. The crossover operation allows the resultant offspring to inherit the favorable traits of both the parents. Mutation and crossover operations ensure that the offsprings are not identical to their parents resulting in exploration of unkown regions of the search space. Many variants of GAs have been proposed and applied to a variety of engineering problems. Some of them are Nelder Mead GA [2], Gradient Descent GA and Real valued GA [3]. A feature of GA is the loss of valuable information stored by less fit individuals during the selection process. This loss of information reduces the performance of GAs on highly multimodal cost functions. Other evolution based metaheuristics include Differential evolution, Evolutionary algorithm and Evolutionary Programming [2].

The third main category of metaheuristics draws inspiration from the laws of physics. These include Gravitational Search algorithm [7], Electromagnetic field optimization [10], Electromagnetism metaheuristic [22] and Ray optimization [1]. Gravitational search algorithm simulates the interaction between collection of masses in the universe based on Newton's laws of gravity and motion. The gravitational force which acts upon the masses is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. Good solutions are assumed to have higher mass and therefore attract other masses towards them. This helps in convergence to the global minimum. Masses also have velocity and acceleration governed by the Newton's laws of motion [7]. Velocity of a particle depends upon the previous velocity and the acceleration at the moment.

Since comparison to all known metaheuristic optimization algorithms is not possible, representatives from each category are taken for comparison in this paper.

3. Wildebeest herd behaviour

The biological inspiration behind WHO algorithm is discussed in this section. The mathematical formulation is provided in Section 4. The movement of individual wildebeests and entire wildebeest herds in search of regions with high grass density is very similar to the operation of population based stochastic global search algorithms. The search by wildebeests to identify regions of high grass density is analogous to the search for points of minimum cost by global optimization algorithms. Although the exact behaviour of wildebeests in a herd is quite complicated and difficult to model, the following simplified wildebeest behaviours are proposed as the basis of a new global optimization algorithm:

I **Local movement (Milling behaviour):** Individual wildebeests are short-sighted relying mostly on sound and smell. So individual wildebeests can only perform a local search for regions of high grass density [23].

II **Herd instinct:** Wildebeests also have a herd instinct and move towards distant random individuals in the herd provided they are located in regions of high grass density. This behaviour happens with a small probability since crowding and poor eyesight limit vision range [23].

III **Starvation avoidance:** Wildebeest move away from barren or overexploited regions to avoid starvation [23].

IV **Population pressure:** Wildebeest movement towards regions of high grass density is balanced by tendency of wildebeests to move away from crowded regions to avoid territorial conflicts [24].

V **Herd social memory:** Finally the entire wildebeest herd as a whole retains memory of fertile regions of high grass density from past experiences and migrations [25]. Thus a search around regions of previously identified best locations is also performed.

4. Mathematical formulation of the WHO algorithm

A simplified mathematical model of wildebeest behaviour is presented in this section as the basis of a new heuristic global optimization algorithm. The complete WHO algorithm is presented in Table 1. The distribution of a population of wildebeest in a

Table 1
Nomenclature

N	No of iterations
D	Dimension of the search space
f	Cost function
N	Size of wildebeest population
x^P	Position of the P^{th} wildebeest
η	Learning rate parameter
U	Standard uniform random variable
α_1, β_1	Parameters controlling local movement of wildebeest
α_2, β_2	Parameters controlling global movement of wildebeest
δ_w	Distance threshold for nearness to regions with least grass
δ_c	Distance threshold to avoid crowding in regions with most grass
n_s	No of exploration steps taken by each wildebeest
n_e	No of exploitative steps around best solution
P_H	Probability of following herd instinct

Table 2
Wildebeest herd optimization algorithm

```
Randomly initialize $X^P \in [X_{\max}, X_{\min}]^D$ for $P = 1$ to $N$ from a uniform probability distribution.
Let $C_P := f(X^P)$ for $P = 1$ to $N$
for $i = 1$ to $I_{\max}$
    for $P = 1$ to $N$
        for $s = 1$ to $n_s$
            $y^s = x^P + \eta u v$
            $c'_s = f(y^s)$
        $s^* = \arg\min_s c_s$
        $x^P = \alpha_1 y^{s^*} + \beta_1 (x^P - y^{s^*})$
        $c_P = f(x^P)$
        if $(c_P < c^*)$ then update $x^*$ & $c^*$
    for $P = 1$ to $N$
        $h :=$ random_integer$(N)$
        if $(c^h < c^P$ AND $u < P_H)$
            $x^P = \alpha_2 x^P + \beta_2 x^h$
            $c_P = f(x^P)$
        if $(c_P < c^*)$ then update $x^*$ & $c^*$
    $x^\omega = \arg\max_{x^P} f(x^P)$
    for $P = 1$ to $N$
        if $\|x^P - x^\omega\| < \delta_\omega$
            $x^P := x^P + u\chi_{(x_{\max} - x_{\min})}\hat{v}$
            $c_P := f(x^P)$
            if $c_P < c^*$ then update $x^*$ & $c^*$
        if $(\|x^* - x^P\| < \delta_c$ AND $\|x^* - x^P\| > 1)$
            $x^P = x^* + \eta\,\hat{n}$
            $c_P = f(x^P)$
            update $x^*$ & $c^*$
        for $n = 1$ to $n_e$
            $x = x^* + 0.1\hat{v}$
            $c = f(x)$
            if $c < c^*$ then update $x^*$ & $c^*$
```

grassland is modelled as random points within a search space (Step 1 in Table 1).

4.1. Local movement

First the 'local movement' or Milling Behaviour can be modelled as taking a fixed number $\mathbf{n_s}$ of small random steps around each tentative solution or wildebeest individual location and movement towards the best local location identified. A random exploratory step y^s taken by a wildebeest located at x^p must equally explore all directions, so small steps are taken in the direction of a random unit vector. Each wildebeest step length is also random so to allow the steps to have variable length the random unit vector is multiplied with ηu, where η is a small constant and u is a uniform random number in [0,1]. Thus the local exploratory step y^s taken by each wildebeest is modelled as (1).

$$y^s = x^p + \eta u v \tag{1}$$

Next a fixed number $\mathbf{n_s}$ of small random exploratory are taken in accordance with (1) and the wildebeest is moved toward the best location identified stochastically. This movement towards the best location identified is achieved by linearly combining x^p and y^{s^*} as in (2).

$$x^p = \alpha_1 y^{s^*} + \beta_1(x^p - y^{s^*}) \tag{2}$$

4.2. Herd instinct

The 'herd instinct' is modelled by movement of each wildebeest towards other wildebeests. This can be achieved by choosing a random individual in the herd and moving towards that individual with a small

Fig. 1. A herd of wildebeest grazing the grasslands [13].

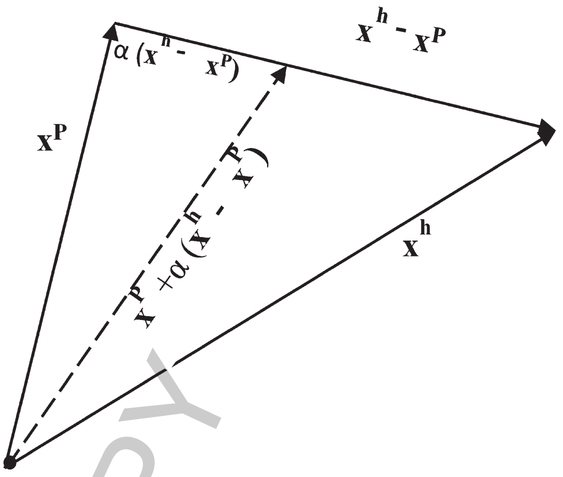

Fig. 2. Illustration of the combination of two wildebeest's position.

probability if that random individual is in a more fertile region. The herd instinct driving a wildebeest to move towards other wildebeest is only actualized if the other wildebeest is located in a region with more food. This movement is implemented by linearly combining the position of each wildebeest x^p with that of a randomly chosen fitter wildebeest x^h as in (3). Figure 2 illustrates the combination of x^p and x^h.

$$x^P = \alpha_2 x^P + \beta_2 x^h \tag{3}$$

4.3. Starvation avoidance

Movement of wildebeest away from regions without grass to avoid starvation is modelled by movement away from regions with least grass. This behaviour is analogous to movement away from the worst solution and can be modelled as taking a random step away from the worst solution (4)

$$x^P := x^P + u\chi(x_{\max} - x_{\min})\hat{v} \tag{4}$$

In Equation (4), $\hat{v}$ is a random unit vector which is multiplied by a step size parameter proportional to the search range. Further to account for variable wildebeest step lengths the search range is multiplied by a uniform random number u in [0, 1].

4.4. Population pressure

The population pressure in regions of highest grass density lead to territorial conflicts between wildebeest and movement of losing individuals away from regions of highest grass density. The strongest wildebeest dominate the regions of high grass density and drive away competitors. This is modelled by movement of wildebeest near but not too close to the best solution away from it in accordance with (5).

$$IF(\|x^* - x^P\| < \delta_c AND \|x^* - x^P\| > 1)$$
$$THEN \quad x^P = x^* + \eta\hat{n} \tag{5}$$

4.5. Herd social memory

The social memory of the entire herd of locations of regions of high grass density and growth rates leads the herd to explore previously known fertile regions. However the social memory is not perfect resulting in different individuals in the herd remembering a slightly different location. This social memory of best locations and imperfect recall of the best location can be modelled as random search around the best location. This local search around the best solution is implemented by taking random exploratory steps around the best solution to refine it as in (6).

$$x = x^* + 0.1\hat{v} \tag{6}$$

5. Results and discussion

The WHO algorithm proposed in this paper was compared to representative algorithms from swarm inspired, evolution inspired and physics inspired metaheuristics. The benchmark functions taken for comparative analysis is presented in Table 3. To challenge the explorative ability of the WHO algorithm, the search range was set as large as possible and multimodal high-dimensional test functions were used. Testing on high dimensional search spaces is done to check if the performance degrades gracefully with increasing dimension [12]. The choice of the test suite of benchmark functions was guided by [17, 18] for large scale optimization problems. A brief description of the challenged posed by different benchmark functions is first provided before discussion of results. The Sphere function is a unimodal convex function. It is also differentiable and separable. The Sphere function tests the ability of an algorithm to quickly and accurately explore a single global minimum. Simple local search algorithms algorithms like Gradient Descent and Nelder-Mead Simplex method perform well on the Sphere function. Stochastic search algorithms are designed for test functions with many local minima and hence perform poorly on unimodal functions because computational effort is wasted by exhaustive random search of the solution space. If a function is known to be convex very efficient convex optimization techniques exist and so global optimization methods should not be used. The Sphere function tests the local search or exploitative ability of a global optimization algorithm. Zakharov is also a unimodal function but with a flat region near the minimum. Flat regions are quite challenging to optimization algorithms since local movement around any point in flat regions do not lead to a reduction in the function value. Since all search directions are equally feasible inside flat regions, a random search must be performed to explore flat regions. The Rastrigin, Ackley and Griewangk functions are highly multi modal functions which test the ability of the algorithm to escape local minima and locate the true global minimum. The minimum of the Noisy Sphere function, is offset by a Gaussian random number vector. This tests the robustness of the proposed algorithm to incorrect or noisy function values. Rosenbrock function has a flat narrow ridge that makes finding good search directions extremely challenging. Rotated version of the Ackley and Griewank test functions were obtained by applying Salomon's random coordinate rotation technique. The global minimum of Shifted-Sphere, Shifted-Rosenbrock and Shifted-Griewangk is randomly located in the hypercube $[-10, 10]^D$. Shifted and rotated test functions are used to weed out algorithms that have an inherent systemic bias. Algorithms which are biased to a predefined point like [0, 0,... 0] will perform badly on shifted test functions. Algorithms that have a systemic bias in the search direction will perform poorly on rotated test functions. Thus rotated and shifted test functions provide a measure of the true performance of an optimization algorithm because they provide an unbiased performance metric.

The parameters of the WHO algorithm were set to perform uniformly well on all test functions chosen. A population size of 20 was found to be adequate and provide good performance for all dimensions up to 100. The population size and the number of iterations is usually progressively increased with increasing dimension [6]. However WHO algorithm demonstrated good performance on the entire test suite with the population size fixed at 20. Parameters α_1, β_1control the local movement of the wildebeest towards region of higher grass density and were empirically observed to provide good local

Table 3
Benchmark Test functions used for comparison

Function (Type)	Definition	Search Range	Position of x^*
Sphere(Unimodal)	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	$[0_1, 0_2, ...0_D)$
Sumsquares(Unimodal)	$f_2(x) = \sum_{i=1}^{D} ix_i^2$	$[-10, 10]^D$	$[0_1, 0_2, ...0_D)$
Step(Unimodal)	$f_3(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]^D$	$[0_1, 0_2, ...0_D)$
Noisy sphere(Unimodal)	$f_4(x) = \sum_{i=1}^{D} y_i^2; \quad y = x + N(0, 1)$	$[-100, 100]^D$	$N(0, 1)$
Zakharov(Unimodal)	$f_5(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} \frac{i}{2} x_i)^2 + (\sum_{i=1}^{D} \frac{i}{2} x_i)^4$	$[-100, 100]^D$	$[0_1, 0_2, ...0_D)$
Schwefel2.21(Unimodal)	$f_6(x) = \max_i \{\lVert x_i \rVert, 1 \leqslant i \leqslant D\}$	$[-100, 100]^D$	$[0_1, 0_2, ...0_D)$
Rosenbrock (Multimodal)	$f_7(x) = \sum_{i=1}^{D-1} [100(x_{i+1}^2 - x_i^2) + (1 - x_i)^2]$	$[-30, 30]^D$	$[1_1, 1_2, ...1_D)$
Rastrigin (Multimodal)	$f_8(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\Pi x_i)]$	$[-5.12, 5.12]^D$	$[0_1, 0_2, ...0_D)$
Griewangk(Multimodal)	$f_9(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$	$[0_1, 0_2, ...0_D)$
Ackley (Multimodal)	$f_{10}(y) = 20 + e - \exp(\frac{1}{D}\sum_{i=1}^{D} \cos(2\Pi y_i)) - 20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2})$ $y = M * x$	$[-40, 40]^D$	$[0_1, 0_2, ...0_D)$
Rotated Ackley (Multimodal)	$f_{11}(y) = 20 + e - \exp(\frac{1}{D}\sum_{i=1}^{D} \cos(2\Pi y_i)) - 20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2})$ $y = M * x$	$[-40, 40]^D$	$[0_1, 0_2, ...0_D)$
Rotated Griewangk (Multimodal)	$f_{12}(y) = \sum_{i=1}^{D} \frac{y_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{y_i}{\sqrt{i}}) + 1$ $y = M * x$	$[-00, 600]^D$	$[0_1, 0_2, ...0_D)$
Shifted Sphere (Unimodal)	$f_{13}(y) = \sum_{i=1}^{D} y_i^2; y = (x - o)$	$[-100, 100]^D$	$[o_1, o_2, ...o_D)$
Shifted Rosenbrock (Multimodal)	$f_{14}(y) = \sum_{i=1}^{D-1} [100(y_{i+1}^2 - y_i^2) + (1 - y_i)^2]; y = (x - o)$	$[-30, 30]^D$	$[o_1, o_2, ...o_D)$
Shifted Griewangk (Multimodal)	$f_{15}(y) = \sum_{i=1}^{D} \frac{y_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{y_i}{\sqrt{i}}) + 1; y = (x - o)$	$[-600, 600]^D$	$[o_1, o_2, ...o_D)$

search ability when set to 0.9 and 0.3 respectively. Parameters α_2, β_2 control the global movement of the wildebeest and were observed to provide good exploration ability when set to 0.2 and 0.8 respectively. These parameter choices were empirically observed to provide a good tradeoff between exploration and exploitation. Improper parameter settings can result in premature convergence to local minima or divergence to infinity.

Figure 2 illustrates the linear combination of two wildebeest's positions x^p and x^h. The choice of α determines the movement of the wildebeest. If α is 0.5 then it is midway between the two members. Since good solutions need to be explored near x^h the value was set to be 0.9. If the value of α is more than 1, it will overshoot the position of the wildebeest.

The performance of the proposed WHO algorithm was compared with GA, PSO, ABC, SA and GSA. The algorithms were tested on unimodal, multimodal, rotated and shifted benchmark functions for 30, 50 and 100 dimensions. The average performance over 50 independent trials was considered to reduce the effects of lucky or unlucky random initial positions. The mean, median and standard deviation for the 50 independent runs for each test function is presented in Tables 4 to 6.

The mean and standard deviation are significantly influenced by outliers therefore the median is also

Table 4
Test results for 30 dimensions

Algorithm	Sphere			SumSquares			Step		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**2.02E-07**	**2.85E-08**	**2.32E-07**	**9.57E-06**	**8.96E-06**	**4.28E-06**	**0**	**0**	**0**
PSO	2.74E+03	2.63E+03	989.0208	3.15E+02	279.4504	155.2565	3.54E+03	3.51E+03	1.46E+03
GA	1.48E+01	13.669	5.1733	1.89E+02	1.83E+02	78.2781	24.7333	24.5	8.9786
GSA	8.68E-02	2.66E-01	2.01E-01	8.56E-06	1.8585e-11	4.54E-05	294.5667	207	174.7086
ABC	10.7737	6.5475	13.9403	32.4749	18.1887	39.3715	11.1667	9	7.0373
SA	7.85E+04	7.72E+04	1.48E+04	1.14E+06	1.07E+06	2.74E+05	9.79E+04	102297	1.70E+04
Algorithm	Noisy Sphere			Zakharov			Schwefel2.21		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.0141**	**1.38E-02**	**0.0025**	**0.0026**	**9.59E-04**	**0.0026**	**0.0025**	**0.0022**	**9.74E-04**
PSO	3.07E+03	2.58E+03	1.63E+03	67.3767	1.41E+02	126.2502	2.5993	2.5395	0.5641
GA	20.28	20.815	5.5386	2.39E+03	1.03E+03	2.85E+03	1.9285	1.8515	0.4425
GSA	1.10E+03	804.0117	461.0469	3.61E+03	3.33E+03	513.6956	9.412	8.9184	0.9574
ABC	50.3589	39.9688	34.074	1.48E+05	1.37E+05	5.28E+04	64.4395	65.495	3.7946
SA	9.24E+04	9.49E+04	1.51E+04	6.76E+13	4.06E+09	2.02E+14	81.5564	82.6719	5.0144
Algorithm	Rosenbrock			Rastrigin			Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**28.3593**	**1.1313**	**24.2867**	**0.0023**	**0.0021**	**0.001**	**9.09E-08**	**5.84E-08**	**7.68E-08**
PSO	1.04E+06	8.46E+05	4.76E+05	3.95E+03	3.63E+03	3.71E+03	31.2244	24.012	23.7046
GA	2.39E+03	2.34E+03	1.16E+03	1.23E+02	123.4266	12.8823	0.5275	0.5532	0.1443
GSA	2.59E+01	2.58E+01	0.3245	4.25E+01	4.22E+01	3.9618	183.3737	1.80E+02	8.3641
ABC	6.69E+06	5.97E+06	2.82E+06	1.09E+02	30	1.63E+02	1.03	1.0326	3.84E-02
SA	5.25E+07	4.77E+07	2.65E+07	3.11E+02	3.19E+02	37.3797	884.4891	891.0254	134.3867
Algorithm	Ackley			Rotated Ackley			Rotated Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.0835**	**3.43E-04**	**0.3174**	**7.40E-04**	**3.29E-04**	**0.0022**	**1.92E-04**	**1.85E-04**	**7.15E-05**
PSO	20.6062	20.5885	0.1582	20.7967	20.8636	0.1625	2.92E+01	2.72E+01	12.5812
GA	3.9955	3.9590	0.4431	3.9685	3.9337	0.5545	5.76E-01	5.64E-01	0.1739
GSA	20.0094	20.0000	0.0516	20.8589	20.9841	0.1548	1.81E+02	180.5911	11.5681
ABC	20.1789	2.02E+01	0.3614	20.1155	2.02E+01	0.6335	1.03	1.03	0.0791
SA	20.1941	2.02E+01	0.1072	20.1725	20.1779	0.1402	8.67E+02	888.289	159.5704
Algorithm	Shifted Sphere			Shifted Rosenbrock			Shifted Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.0056**	**0.0058**	**5.54E-04**	**29.3699**	**29.5705**	**0.5546**	**0.0069**	**0.0077**	**0.0076**
PSO	3.96E+03	3.28E+03	1.70E+03	4.54E+06	2.95E+06	3.10E+06	28.6214	26.7513	9.9825
GA	3.12E+03	1.82E+03	2.86E+03	963.5006	1.98E+03	332.4126	2.05E+05	1.98E+05	8.73E+04
GSA	37.3499	9.1276	111.8367	1.61E+06	1.10E+06	1.77E+06	153.5281	148.9120	17.6426
ABC	4.0499	2.3769	4.3353	4.45E+06	4.66E+06	2.42E+06	1.0241	1.0353	0.0841
SA	8.54E+04	8.47E+04	1.24E+04	1.56E+08	1.61E+08	7.13E+07	872.2425	851.1212	143.4604

provided for comparison. The test results presented in Tables 4 to 6 shows that the proposed WHO algorithm is competitive with other popular algorithms. Table 4 compares the WHO algorithm with other algorithms for 30 dimensions. WHO algorithm returns better results than the other algorithms for all the functions. WHO algorithm returns zero within machine precision for the step function. GSA provides the second best solution for Schwefel 2.21 and the shifted versions of the Sphere, Rosenbrock and Griewangk. GA SA,ABC and PSO underperform in almost all the test functions

Table 5 presents the results for 50 dimensions. It is observed that the proposed WHO algorithm performs significantly better for 50 dimensions. GSA algorithm which performed competitively well with the WHO algorithm in 30 dimensional search space, underperforms in 50 dimensions. As in the case of 30 dimensions, GA, SA, ABC and PSO underperform in almost all the test functions.

The results for 100 dimensions is presented in Table 6. It is observed that the WHO algorithm returns the best result for all the test functions. The significantly better performance of the WHO algorithms for 100 dimensions clearly demonstrates its superior explorative ability. The WHO algorithm is also observed to display graceful degradation of performance with increasing dimension in contrast to other

Table 5
Test results for 50 dimensions

Algorithm	Sphere			SumSquares			Step		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**8.63E-07**	1.29E-07	**6.69E-07**	**3.78E-04**	**2.87E-04**	**3.02E-04**	**0**	**0**	**0**
PSO	9.29E+03	8.77E+03	2.49E+03	1.98E+03	1.90E+03	575.9947	9.89E+03	9.43E+03	2.70E+03
GA	1.48E+01	13.669	5.1733	7.16E+02	703.3009	212.4076	4.73E+01	4.55E+01	9.0108
GSA	248.2691	2.20E+02	115.3174	0.0163	8.36E-05	0.0399	916	916	103.7437
ABC	1.03E+03	**0**	4.4E+03	2.53E+05	2.26E+05	1.06E+05	1.55E+04	1.38E+04	6.20E+03
SA	1.36E+05	1.36E+05	1.91E+04	3.27E+06	3.37E+06	6.07E+05	1.63E+05	161417	1.89E+04
Algorithm	Noisy Sphere			Zakharov			Schwefel2.21		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.045**	**0.0447**	**0.0029**	**0.0145**	**0.0142**	**0.0022**	**0.0061**	**0.0056**	**0.0092**
PSO	4.07E+03	3.18E+04	1.92E+03	236.0479	377.477	339.1979	3.2297	3.2173	0.4618
GA	1.02E+04	9.99E+03	2.98E+03	2.25E+03	1.70E+03	1.75E+03	2.2671	2.2211	0.4383
GSA	2.70E+03	2.52E+03	559.8626	6.95E+03	6.54E+03	896.5153	0.0062	4.10E-09	0.0342
ABC	1.65E+04	1.88E+04	6.20E+03	4.71E+05	4.25E+05	2.73E+05	87.6695	88.3025	3.723
SA	1.57E+05	1.54E+05	2.06E+04	6.90E+14	3.82E+12	1.89E+15	86.774	86.8178	2.622
Algorithm	Rosenbrock			Rastrigin			Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**48.2349**	**9.8039**	**48.2036**	**0.0065**	**0.0062**	**0.002**	**1.99E-04**	**8.28E-05**	**2.28E-04**
PSO	6.64E+06	3.53E+06	3.23E+06	1.05E+04	1.19E+04	1.12E+04	1.21E+02	92.2615	28.1263
GA	6.82E+03	5.70E+03	3.39E+03	267.782	267.1457	40.8137	0.6327	0.6255	0.1395
GSA	4.80E+01	46.9983	1.2251	1.24E+02	119.3178	15.4827	354.9249	347.8052	13.6617
ABC	2.41E+08	2.45E+08	4.91E+07	5.91E+02	589.9039	29.2372	1.25E+02	1.16E+02	43.4619
SA	1.56E+08	1.44E+08	5.75E+07	5.67E+02	565.5729	58.5504	1.49E+03	1.50E+03	145.9811
Algorithm	Ackley			Rotated Ackley			Rotated Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**5.40E-04**	**5.41E-04**	**3.33E-05**	**0.0394**	**5.27E-04**	**0.2125**	**1.10E-07**	**8.59E-08**	**7.77E-08**
PSO	20.8514	20.8448	0.1352	21.0043	21.0396	0.1365	9.13E+01	90.2282	14.9412
GA	4.3960	4.4121	0.3591	4.4492	4.5362	0.3836	0.6526	0.6568	0.1200
GSA	20.0979	20.0603	0.1924	21.1147	21.108	0.0319	293.687	288.601	27.857
ABC	21.0567	21.0941	0.102	21.0542	21.1073	0.1543	130.296	1.25E+02	4.08E+01
SA	2.05E+01	2.05E+01	0.0913	2.05E+01	20.5166	0.088	1.43E+03	1.36E+03	203.9101
Algorithm	Shifted Sphere			Shifted Rosenbrock			Shifted Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.0092**	**0.0092**	**7.78E-04**	**49.5788**	**49.5864**	**0.4395**	**0.0058**	**0.0077**	**0.0049**
PSO	1.58E+04	1.59E+04	3.49E+03	2.94E+07	2.50E+07	1.29E+07	94.9663	94.4131	21.3237
GA	1.62E+04	4.72E+03	6.37E+04	2.16E+07	1.81E+03	8.36E+07	1.59E+05	1.67E+05	1.10E+05
GSA	7.11E+03	6.33E+03	2.05E+03	1.14E+07	1.18E+07	7.58E+05	363.8755	363.5631	1.7112
ABC	1.25E+04	1.17E+04	4.80E+03	2.38E+08	2.26E+08	6.44E+07	137.0022	119.4885	42.5872
SA	1.61E+05	1.59E+05	2.48E+04	4.23E+08	3.93E+08	1.51E+08	1.47E+03	1.51E+03	201.7547

algorithms. GSA, ABC, SA, GA and PSO return considerably poor results for 100 dimensions clearly indicating the steep degradation in the performance with increase in the dimension.

Convergence plots are shown in Figs. 3 to 10 for the 100 dimension test cases. Figures 3 to 10 illustrate the convergence characteristics of the algorithms for Griewangk, Schwefel2.21, Rotated Ackley, Noisy sphere, Step, Ackley, Rastrigin and Sphere respectively. Figures 3 to 10 indicate that the WHO algorithm approaches close to the global optimum in less than 200 iterations for a majority of test functions.

Many optimization algorithms perform poorly when the global minimum is significantly shifted from the origin. These algorithms have a systematic bias and converge to points near the origin even when the global minimum is not located at the origin. Thus to obtain a realistic measure of performance on real world problems test functions with global minimum significantly shifted from $x = (0, 0, ..., 0)$ must be considered. Results for large shifts is presented in Tables 4, 5 & 6. The results clearly show that WHO algorithm performs significantly better than all other algorithms taken for comparison on test functions with large shifts.

Table 6
Test results for 100 dimensions

Algorithm	Sphere			SumSquares			Step		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.005**	0.0029	**0.003**	**0.4465**	**0.4562**	**0.1805**	**0**	**0**	**0**
PSO	2.85E+04	2.83E+04	5.73E+03	1.41E+04	1.36E+04	3.56E+03	3.37E+04	3.38E+04	6.67E+03
GA	76.1632	74.0793	13.6791	3.72E+03	3.70E+03	717.0759	9.49E+01	96.5	15.5757
GSA	2.66E+03	2.69E+03	663.409	2.7641	0.9763	2.0331	4.76E+03	4141	1.40E+03
ABC	2.98E+04	**0.00E+00**	8.94E+04	1.12E+07	1.13E+07	8.92E+05	2.54E+05	254958	1.57E+04
SA	2.83E+05	2.86E+05	2.84E+04	1.43E+07	1.46E+07	1.77E+06	3.26E+05	3.24E+05	2.42E+04
Algorithm	Noisy Sphere			Zakharov			Schwefel2.21		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.5689**	**0.5902**	**0.1284**	**0.0816**	**0.0088**	**0.0879**	**0.094**	**0.0832**	**0.0465**
PSO	3.14E+04	3.23E+04	5.61E+03	1.61E+05	1.43E+05	8.55E+04	3.9369	3.935	0.5158
GA	110.8273	105.9547	20.6769	5.80E+03	4.97E+03	3.74E+03	2.5077	2.5095	0.4612
GSA	9.07E+03	8.74E+03	1.63E+03	1.55E+04	1.41E+04	2.08E+03	21.3567	18.5681	14.383
ABC	2.58E+05	2.61E+05	1.73E+04	1.66E+06	1.49E+06	9.20E+05	95.8895	96.0463	0.9231
SA	3.25E+05	3.26E+05	2.72E+04	2.18E+16	4.41E+08	5.12E+16	91.1831	90.6008	1.5359
Algorithm	Rosenbrock			Rastrigin			Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**98.6786**	**0.4391**	**97.7565**	**0.1021**	**0.0899**	**0.0509**	**1.86E-04**	**1.88E-04**	**6.2497E-05**
PSO	2.29E+07	1.92E+07	1.17E+07	3.56E+04	3.53E+04	7.46E+03	279.0246	263.6163	65.2342
GA	2.19E+04	2.03E+04	9.29E+03	690.0805	677.127	73.0644	0.8203	0.8156	0.1149
GSA	152.2187	103.6792	71.1242	498.3507	467.3594	64.5403	8.06E+02	807.5075	3.1139
ABC	1.17E+09	1.18E+09	9.18E+07	1.61E+03	1.62E+03	56.8469	2.32E+03	2.31E+03	150.6027
SA	5.73E+08	5.69E+08	1.00E+08	1.31E+03	1.31E+03	98.6534	2.93E+03	2.90E+03	264.0962
Algorithm	Ackley			Rotated Ackley			Rotated Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.0371**	**0.0352**	**0.0109**	**0.1347**	**0.0358**	**0.3061**	**1.76E-04**	**1.590E-04**	**5.8126E-05**
PSO	21.1672	21.1650	0.0771	21.2741	21.2772	0.0653	0.7838	0.7764	0.1130
GA	4.8179	4.8297	0.2661	4.7995	4.8435	0.2923	795.3487	788.3802	21.3046
GSA	20.0421	20.0000	0.2285	21.3345	21.3542	0.0249	822.2099	840.9590	38.1392
ABC	21.3021	21.3153	0.043	21.3175	21.3204	0.0292	2.33E+03	2.31E+03	126.4207
SA	20.8157	20.8057	0.0745	20.8167	20.8118	0.0512	3.01E+03	3.00E+03	270.9423
Algorithm	Shifted Sphere			Shifted Rosenbrock			Shifted Griewangk		
	Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
WHO	**0.0257**	**0.0258**	**0.0023**	**100.2898**	**100.4097**	**0.4424**	**0.0024**	**5.30E-04**	**0.0035**
PSO	5.26E+04	5.18E+04	8.10E+03	1.88E+08	1.79E+08	5.45E+07	291.0571	287.8777	74.4098
GA	3.28E+13	1.86E+05	1.58E+14	1.10E+13	1.46E+08	3.91E+13	2.52E+12	3.84E+05	1.08E+13
GSA	8.96E+03	8.07E+03	2.24E+03	1.79E+08	1.87E+08	4.50E+07	816.2332	817.9467	28.2019
ABC	2.78E+05	2.82E+05	2.08E+04	2.23E+09	2.20E+09	2.98E+08	2.35E+03	2.32E+03	156.5436
SA	3.25E+05	3.26E+05	3.37E+04	1.41E+09	1.41E+09	3.29E+08	2.88E+03	2.85E+03	236.8834

5.1. Time complexity analysis

The time complexity of stochastic optimization algorithms can be measured by computing the average time required to achieve a predefined accuracy. This average must be take over a large number of independent runs to reduce the effects of lucky or unlucky random number sequences generated during the run of the algorithm. In this paper the average over 30 independent trials is taken since the average of 30 independent random variables is approximately Gaussian according to the Central Limit Theorem. Table 7 below shows the average time required to compute the global minimum to an accuracy of

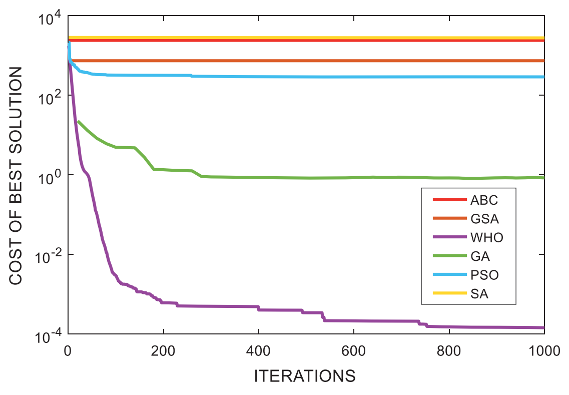

Fig. 3. Convergence characteristics of Griewangk function.

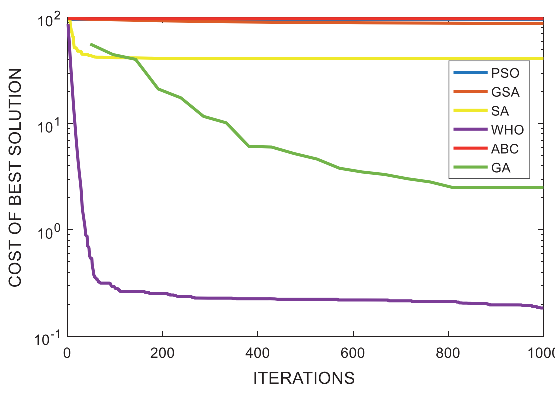

Fig. 4. Convergence characteristics of Schwefel 2.21 function.

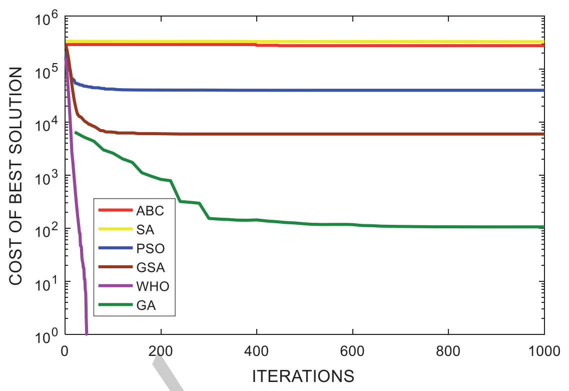

Fig. 7. Convergence characteristics of Step function.

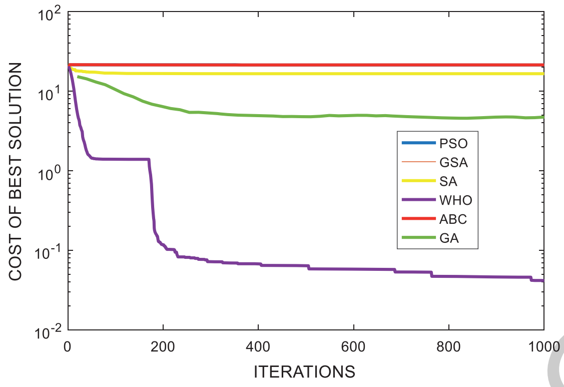

Fig. 5. Convergence characteristics of Rotated Ackley function.

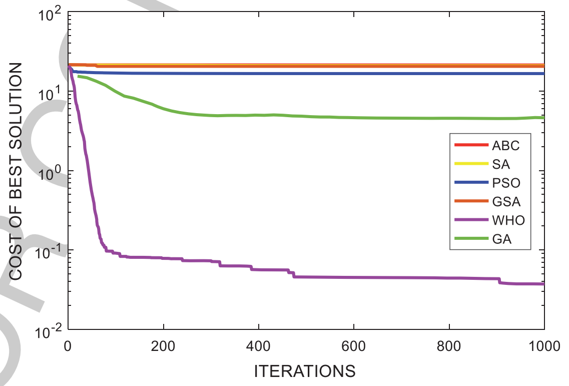

Fig. 8. Convergence characteristics of Ackley function.

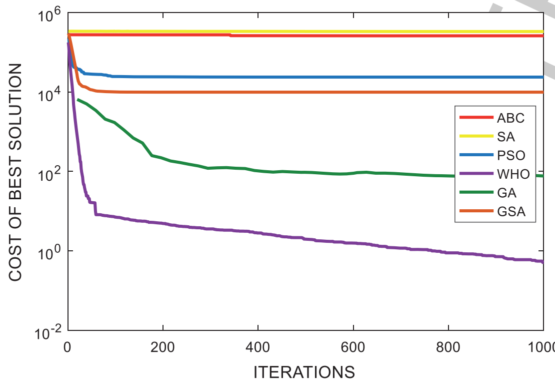

Fig. 6. Convergence characteristics of Noisy Sphere function.

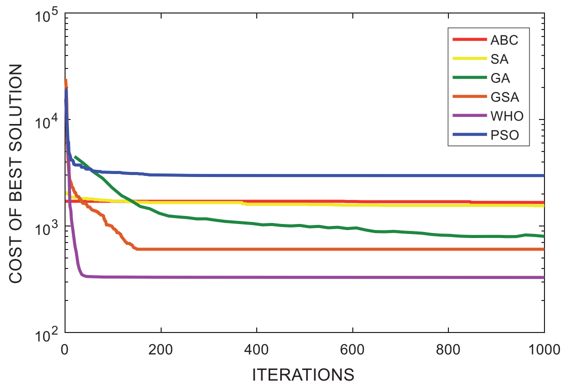

Fig. 9. Convergence characteristics of Rastrigin function.

0.0001 for 50 dimensions. This accuracy is chosen since it is sufficient for most engineering applications.

The functions that were taken for comparison represent the unimodal, multimodal and shifted benchmark test functions. The functions f_1 to $f_6$are Griewangk, Step, Rotated Griewangk, Sumsquares, Noisy Sphere and Shifted Sphere respectively. The best mean running time is highlighted in bold. Table 7 clearly indicates that the proposed algorithm converged to the pre-set accuracy with the smallest mean running time in most of the test functions.

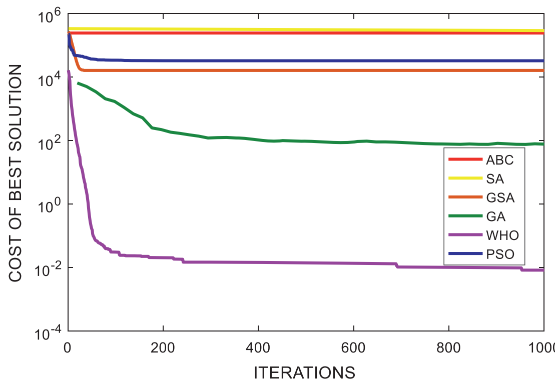

Fig. 10. Convergence characteristics of Sphere function.

Table 7
Test results on mean running time for 50 dimensions

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6
WHO	**0.5319**	**0.0738**	1.0807	1.2158	**1.2027**	**1.3303**
GA	1.0793	0.8944	0.8167	1.8037	1.8907	1.9127
PSO	1.99921	2.00084	2.50431	1.84016	1.92294	1.8877
GSA	1.9679	1.8837	2.3362	1.8287	1.8643	1.8924
ABC	1.7027	1.4806	2.6113	1.3688	1.4688	1.3650
SA	0.6174	0.3340	**1.0439**	**1.2106**	1.3404	1.3354

6. Conclusion

In this paper a new meta-heuristic global optimization algorithm inspired by Wildebeest herd behaviour was proposed and shown to perform significantly better on high-dimensional and shifted test functions. The proposed WHO algorithm was based on a simplified mathematical model of Wildebeest milling behaviour, herd instinct, starvation avoidance instinct, population pressure and herd social memory. Extensive simulation results on 15 benchmark test functions indicated that the WHO algorithm is highly competitive with well-known global optimization algorithms like GA, GSA, PSO, ABC and SA especially on high dimensional and shifted test functions. In particular the WHO algorithm exhibited superior explorative ability and performed significantly better on high dimensional test functions with global minimum significantly shifted from $x = (0, 0, ..., 0)$. The WHO algorithm also demonstrated graceful degradation in performance with increasing dimension compared to other algorithms. Future work can explore parallel implementation and variations of the WHO algorithm for discrete (combinatorial) optimization.

References

[1] A. Kaveh and M. Khayatazad, A new meta-heuristic method: Ray optimization, *Computers & Structures* **112** (2012), 283–294.

[2] A.K. Qin, V.L. Huang and P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE transactions on Evolutionary Computation* **13** (2009), 398–417.

[3] C.H. Wu, G.H. Tzeng, Y.J. Goo and W.C. Fang, A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy, *Expert Systems with Applications* **32** (2007), 397–408.

[4] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization* **39** (2007), 459–471.

[5] E. Cuevas and M. González, An optimization algorithm for multimodal functions inspired by collective animal behaviour, *Soft Computing* **17** (2013), 489–502.

[6] E. Kaya, S.A. Uymaz and B. Kocer, Boosting galactic swarm optimization with ABC, *International Journal of Machine Learning and Cybernetics* (2018), 1–19.

[7] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, GSA: A gravitational search algorithm, *Information Sciences* **179** (2009), 2232–2248.

[8] F. Fausto, E. Cuevas, A. Valdivia and A. González, A global optimization algorithm inspired in the behavior of selfish herds, *Biosystems* **160** (2017), 39–55.

[9] A.H. Gandomi and A. Hossein Alavi, Krill herd: A new bio-inspired optimization algorithm, *Communications in Nonlinear Science and Numerical Simulation* **12** (2012), 4831–4845.

[10] H. Abedinpourshotorban, S.M. Shamsuddin, Z. Beheshti and D.N. Jawawi, Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm, *Swarm and Evolutionary Computation* **26** (2016), 8–22.

[11] K. Deb, A. Anand and D. Joshi, A computationally efficient evolutionary algorithm for real-parameter optimization, *Evolutionary Computation* **10** (2002), 371–395.

[12] K. Hussain, M.N. Salleh, S. Cheng and Y. Shi, Metaheuristic research: A comprehensive survey, *Artificial Intelligence Review* (2018), 1–43.

[13] https://www.agefotostock.com/age/en/Stock-Images/plain-wildebeest-grazing.html

[14] M.A. Luersen and R. Le Riche, Globalized Nelder–Mead method for engineering optimization, *Computers & Structures* **82** (2004), 2251–2260.

[15] M. Dorigo and M. Birattari, Ant colony optimization, Springer US. 36–39.

[16] M.M. Noel, A new gradient based particle swarm optimization algorithm for accurate computation of global minimum, *Applied Soft Computing* **12** (2012), 353–359.

[17] M. Molga and C. Smutnicki, Test functions for optimization needs, *Test Functions for Optimization Needs* **101** (2005).

[18] P. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P Chen, A. Auger, and S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, *KanGAL report* (2005).

[19] R. Hassan, B. Cohanim, O. De Weck and G. Venter, A comparison of particle swarm optimization and the genetic algorithm, In *46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference* (2005), 1897.

[20] S. Mirjalili, S.M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software* **69** (2014), 46–61.

[21] V. Muthiah-Nakarajan and M.M. Noel, Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion, *Applied Soft Computing* **38** (2016), 771–787.

[22] Z. Naji-Azimi, P. Toth and L. Galli, An electromagnetism metaheuristic for the unicost set covering problem, *European Journal of Operational Research* **205** (2010), 290–300.

[23] R. Ben Shahar and M.J. Coe, The relationships between soil factors, grass nutrients and foraging behaviour of wildebeest and zebra, *Oecologia* **90** (1992), 422–428.

[24] M. Thaker, A.T. Vanak, C.R. Owen, M.B. Ogden and R. Slotow, Gropu dynamics of zebra and wildebeest in woodland Savanna: Effects of predation risk and habitat density, *PLoS one* **5** (2010), e12758.

[25] W. Corne, A. Reynolds and E. Bonabeau, Swarm intelligence, *Handbook of Natural Computing* (2012), 1599–1622.

[26] M. Bagheri, M. Miri and N. Shabakhty, Fuzzy reliability analysis using a new alpha level set optimization approach based on particle swarm optimization, *Journal of Intelligent & Fuzzy Systems* **30**(1) (2016), 235–244.