12

Elephant Herding Optimization

Nand K. Meena
School of Engineering and Applied Science
Aston University, Birmingham, United Kingdom

Jin Yang
School of Engineering and Applied Science
Aston University, Birmingham, United Kingdom

Adam Slowik
Department of Electronics and Computer Science
Koszalin University of Technology, Koszalin, Poland

CONTENTS

12.1 Introduction

Various researchers across the globe are impressed by the complex social and emotional family structure of elephants, as compared to other animals on the earth; elephant females form and lead the family. The elephants constitute profound clan bonding and like to stay in tight family groups headed by females called a herd. Generally, a herd comprises 8-100 elephants, depending on territory as well as on family size [1]. A herd is led by an old female, known as the matriarch. Mostly, a herd consist of females such as mother, her sisters and their calves. Sometimes, herd aggregation is also observed with 500–1000 elephants near a water and food source. It has been also researched that elephants makes lifelong emotional bonding with their family and friends, and even mourn the death of stillborn babies and their loved ones.

The walking style of elephants is also unique and found to be very similar to humans. In walking, babies generally use trunks to hold the tails of their respective mothers and other females are surrounding them in order to protect them from hungry predators [2]. A new born calf is brought up and protected by the whole matriarchal herd. The male elephants (bulls) like to live a solitary life therefore, leave the family group between the age of 12-15 years to hang alone or with other males.

Sometimes, the herd is also separated even though they are closely related. The separation can be influenced by ecology, and social factors. Therefore, it may also possible that different herds found in a large territory can be from the same family. They always keep in touch with their blood groups by using different types of calls. The elephant has a great sense of hearing and can produce different sounds like roars, snorts, cries etc. to communicate with other groups but is specialised in subsonic rumbling.

O'Connell-Rodwell carried out experiments on captive elephants in the United States, Zimbabwe and India, over a period of 5 years. The outcome of these experimentations showed that elephants respond to the low-frequency sound waves that travel through and just above the ground [3]. A later research also confirmed that the elephants can respond to seismic waves even in the absence of low frequency oscillations. The mammals can detect the stress of a distant herd and incoming water-storms when thunder sounds a hundred miles away, as Sri Lankan and Thai elephants reportedly run away from their locations before the destructive tsunami in the year 2004 [4].

> "The elephants can communicate through their feet, toenails, and trunks up to a distance of 20 miles. They have the ability to hear low frequency sound and seismic waves through their feet"
>
> Dr. Caitlin O'Connell
> Stanford University School of Medicine

The super intelligence and great memory of elephants have inspired a new nature inspired optimization technique. In 2015, Gai-Ge Wang et al. developed a swarm-intelligence based meta-heuristic optimization technique called elephant herding optimization (EHO). The method is inspired by the herding behaviour of elephants. As discussed, the elephant is considered to be a social animal and the herding consists several clans of elephants and their calves. Each clan moves under the influence of a leader, usually a matriarchal female. In the proposed algorithm, the leader elephant is representing the best solution of that clan whereas the male elephant is representing the worst solution. According to elephants' herding behaviour, the female elephants are used to living with their family groups while a male elephant separates when they grow up but lives in contact with their family groups using low frequency vibrations. For simplification, the number of elephants in each clan remains the same. For example, when the worst or male elephant leaves the clan, a new elephant can be produced to keep the number of elephants constant. The basic EHO is discussed in the following sections.

12.2 Elephant herding optimization

The complex herding behaviour of the elephants is modelled into mathematical equations. In order to search the global and local solutions, some set of rules is defined in basic EHO [5], as discussed below.

1. The number of elephants in each clan should be fixed. For example, if any elephant leaves the clan then a new elephant or a baby elephant can replace its position.
2. In each generation/iteration, a fixed number of male elephants will leave their family groups to live in isolation.
3. The elephants of each clan will live under the leadership of their respective matriarch.

An organization of an elephant herd consisting of multiple clans and elephants is presented in Fig. 12.1. In order to update the location of elephants in each generation, two position updating operators have been suggested in EHO namely, clan updating and separating operators. Based on these operators, the EHO method can be divided into two parts as discussed in the following sections.

12.2.1 Position update of elephants in a clan

In this stage, the positions of elephants remaining in the clan are updated. As discussed, the position of a elephant in a clan is affected by the matriarch of the respective clan. In EHO, the position of the jth elephant in clan c is updated as

$$p_{jc}^{t+1} = p_{jc}^{t} + \alpha \times (p_{best} - p_{jc}^{t}) \times r \tag{12.1}$$

where, p_{jc}^{t+1} and p_{jc}^{t} are the updated and previous positions, calculated in $t+1$ and t generations respectively. α is a scaling factor, varied between [0, 1], used

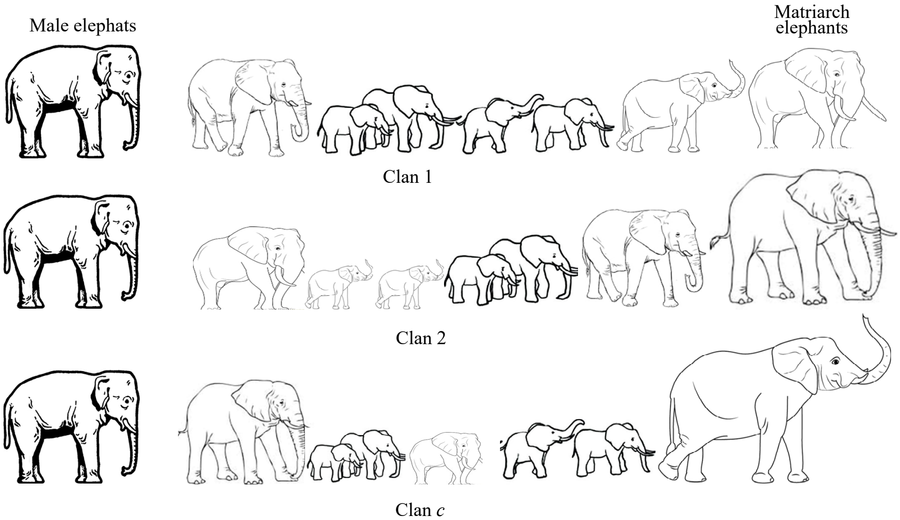

FIGURE 12.1
Elephant herd moving in clans.

to determine the influence of the matriarch on an individual elephant of clan c. p_{best} is the position of the leader elephant of the clan that holds the best fitness achieved so far. $r\epsilon[0, 1]$ is the uniform distribution.

The position of matriarch elephants, i.e., $p^t_{jc} = p_{best}$ is updated as

$$p^{t+1}_{jc} = \beta \times p_{center,c} \tag{12.2}$$

where, $p_{center,c}$ represents the central position of clan c, and $\beta\epsilon[0, 1]$ is the scaling factor influenced by $p_{center,c}$. The matriarch makes use of β to update its own position by ensuring the security of that clan. The centre of a clan c can be determined as

$$p_{center,c} = \frac{\sum_{j=1}^{n_c} p^t_{jc}}{n_c} \tag{12.3}$$

where, n_c is representing the number of elephants in clan c.

12.2.2 Separation of male elephants from the clan

As discussed in previous sections, the male elephants will leave their family when they grown-up, to live a solitary life or with male groups. The separation process needs to be mathematically modelled in the proposed optimization method. The elephant individuals with the worst fitness in each clan will leave their respective clans. In this separation, the new position of worst elephant j in cth clan can be determined as

$$p^{t+1}_{worst,jc} = p_{min,c} + rand \times (p_{max,c} - p_{min,c} + 1) \tag{12.4}$$

where, $p_{min,c}$ and $p_{max,c}$ are the lower and upper bounds of elephant individuals in clan c. $rand\epsilon[0, 1]$ represents a uniform and stochastic distribution.

12.2.3 Pseudo-code of EHO algorithm

The pseudo-code of standard EHO is presented in Algorithm 11.

Algorithm 11 Pseudo-code of EHO.

determine the objective function $OF(.)$
set the number of clans N, where $c\epsilon[1, N]$, and number of elephants in each clan n_c
determine the values of scaling factors α, β, and maximum number of generations, G_{max}
set the lower and upper bounds for each variable/clan c, $[p_{min,c},\ p_{max,c}]$
randomly generate the positions for all elephants in each clan, as follows
for each j-th elephant **do**
    for each c-th clan **do**
        $pp_{jc} = p_{min,c} + (p_{max,c} - p_{min,c}) \cdot rand$
    end for
    $Fitness_j = OF(pp_j)$
end for
determine the best and worst elephants with their locations $bestloc$ and $worstloc$
set generation $t = 1$;
while $t \leq G_{max}$ **do**
    update the position of elephants in all clans, as follows

12.3 Source-code of EHO algorithm in Matlab

In Listing 12.1, the source-code for the EHO algorithm in presented. Here, the $OF(.)$ is representing the address of the objective function.

```
% EHO algorithm
clc ;
clear ;
%------------------------------------------------------------------
% declaration of the parameters of EHO algorithm
%------------------------------------------------------------------
nc=2;            % number of clans
N=10;            % number of elephants in each clan
alpha=0.5;       % scaling factor alpha
beta=0.1;        % scaling factor beta
LB=[-5 -5];      % lower bounds for all clans
UB=[5 5];        % upper bounds for all clans
```

for each j-th elephant **do**
  for each c-th clan **do**
    if any$((j \ = bestloc)\&(j \ = worstloc))$ **then**
      $pp\_new_{jc} = pp_{jc} + alpha \cdot (pbest_c - pp_{jc}) \cdot rand$
    else if $j == bestloc$ **then**
      $pp_{center,c} = mean(pp_c)$
      $pp\_new_{jc} = beta \cdot pp_{center,c}$
    else if $j == worstloc$ **then**
      $pp\_new_{jc} = LB_c + (UB_c - LB_c + 1) \cdot rand$
    end if
  end for
  evaluate the fitness of new individual j as
  $Fitness_j = OF(pp\_new_j)$
end for
determine the new best and worst elephants
if Is new best better than previous best **then**
  replace the best individual with new one
end if
set old population $pp = pp\_new$
set iteration $t = t + 1$
end while
return the *pbest* as a result

```
Gmax = 50;      % maximum number of generations/iterations
%------------------------------------------------
% generate random population of elephants
%------------------------------------------------
for j=1:N
for c=1:nc
% random location for elephant j in clan c
pp(j,c)=LB(c)+rand*(UB(c)-LB(c));
end
% fitness of jth elephants pair in all clans
Fitness(j)=OF(pp(j,:));
end
% best fitness value and its location
fbest=min(Fitness)
bestloc=find(fbest==Fitness);
% worst fitness value and its location
fworst=max(Fitness);
worstloc=find(fworst==Fitness);
% position of fittest elephant or matriarch
pbest=pp(bestloc,:);
% position of weakest elephant or male
pworst=pp(worstloc,:) ;
%------------------------------------
% EHO generation starts .....
%------------------------------------
pp_new=pp;
for gen=1:Gmax
% clan updating and male separation operators
for j=1:N
for c=1:nc
if any((j\~=bestloc)&(j~=worstloc))
```

```
% update elephants positions except best and worst elephants
pp_new(j,c)=pp(j,c)+alpha*(pbest(c)-pp(j,c))*rand;
elseif j==bestloc
pp_center=mean(pp(:,c));
% update leader or matriarch as suggested
pp_new(j,c)=beta*pp_center;
elseif j==worstloc
% update worst elephant as suggested
pp_new(j,c)=LB(c)+(UB(c)-LB(c)+1)*rand;
end
end
% fitness calculation for new population
Fitness(j)=OF(pp_new(j,:));
end
% find the best fitness value
fbest_new=min(Fitness);
% determine the location of best elephant
bestloc_new=find(fbest_new==Fitness);
% find the worst fitness value
fworst=max(Fitness);
% location of worst elephant
worstloc=find(fworst==Fitness);
% fittest elephant or matriarch
pbest_new=pp_new(bestloc_new,:);
%weakest elephant or male
pworst=pp_new(worstloc,:) ;
% Preserve the fittest elephant and position
if fbest_new<fbest
pbest=pbest_new;
fbest=fbest_new;
bestloc=bestloc_new;
end
% replace the old population with new one
pp=pp_new;
% store the best elephant fitness of each generation
b(gen)=fbest;
% store the mean fitness values of elephants
m(gen)=mean(Fitness);
end
% display the position of best elephant or solution
disp(pbest );
% display the fitness of best elephant
disp( fbest );
% plot the best fitness values of all generations
subplot (1 ,2 ,1)
plot(b)
% plot the mean fitness values of all elephants
subplot (1 ,2 ,2)
plot (m)
```

Listing 12.1
Source-code of EHO in Matlab.

12.4 Source-code of EHO algorithm in C++

```
#include <iostream>
#include <algorithm>
#include <math.h>
#include <time.h>
```

```
using namespace std;
// definition of the objective function OF(.)
float OF(float x[], int size_array)
{float t=0;
for(int i=0; i<size_array; i++){
t=t+(x[i]*x[i]-10*cos(2*M_PI*x[i]));}
return 10*size_array + t;}
// generate pseudo random values from the range [0, 1)
float r(){return (float)(rand()%1000)/1000;}
// main program function
int main()
{// initialization of the EHO algorithm parameters
int N=10, nc=2, Gmax=20; float alpha=0.1, beta=0.01;
float LB[nc], UB[nc], pp[N][nc], Fitness[N], pp_new[N][nc];
// initialization of pseudo random generator
srand (time(NULL));
// initialization of the constraints
for(int j=0; j<nc; j++){LB[j]=-5.12; UB[j]=5.12;}
// generate the random positions for elephants of all clans
for(int j=0; j<N; j++)
{
for(int c=0; c<nc; c++)
{
pp[j][c]=LB[c]+r()*(UB[c]-LB[c]);
}
// evaluate all positions of elephants (solutions)
Fitness[j]=OF(pp[j],nc);
}
// determine the matriarch elephant
// best fitness value
float fbest=*min_element(Fitness,Fitness+N);
// worst fitness value
float fworst=*max_element(Fitness, Fitness+N);
// position of the fittest elephant or matriarch
int bestloc=min_element(Fitness, Fitness+N)-Fitness;
// position of the weakest elephant or male who has to leave the clan
int worstloc=max_element(Fitness, Fitness+N)-Fitness;
// initialization of additional variables
float pbest[nc], pworst[nc], pp_center, sum, fbest_new, pbest_new[nc];
// remember the best "pbest" and the worst "pworst" solution
for(int i=0; i<nc; i++){pbest[i]=pp[bestloc][i];
pworst[i]=pp[worstloc][i];}
// EHO generation starts
for(int gen=0; gen<Gmax; gen++)
{
// clan updating and male separation operators
for(int j=0; j<N; j++)
{
for(int c=0; c<nc; c++)
{
if ((j!=bestloc) && (j!=worstloc))
{
// updating the position of the elephants except best and worst elephant
pp_new[j][c]=pp[j][c]+alpha*(pbest[c]-pp[j][c])*r();
}
else if (j==bestloc)
{
sum=0;
for(int i=0; i<N; i++){sum=sum+pp[i][c];}
pp_center=sum/N;
// updating the position of leader or matriarch
pp_new[j][c]=beta*pp_center;
}
else if (j==worstloc)
{
// updating the position of the worst elephant
```

```
pp_new[j][c]=LB[c]+(UB[c]-LB[c]+1)*r();
}
}
// fitness calculation for new positions
Fitness[j]=OF(pp_new[j],nc);
}
// determine the fittest elephant
// find the best fitness value
fbest_new=*min_element(Fitness,Fitness+N);
// find the worst fitness value
fworst=*max_element(Fitness, Fitness+N);
// determine the location of best elephant
int bestloc_new=min_element(Fitness, Fitness+N)-Fitness;
// location of worst elephant
worstloc=max_element(Fitness, Fitness+N)-Fitness;
for(int i=0; i<nc; i++){
// fittest elephant or matriarch
pbest_new[i]=pp_new[bestloc_new][i];
// weakest elephant or male
pworst[i]=pp_new[worstloc][i];}
// preserve the best elephant with its position and fitness
if (fbest_new<fbest)
{
for(int i=0; i<nc; i++){pbest[i]=pbest_new[i];}
fbest=fbest_new; bestloc=bestloc_new;
}
// replace the old population with new one
for(int j=0; j<N; j++){
for(int c=0; c<nc; c++){pp[j][c]=pp_new[j][c];}}
// display the best result at each iteration
cout<<"Iteration "<<gen<<" - The best: "<<fbest<<endl;
}
// display the final solution and final the best result
cout<<"Position of the best elephant (solution):"<<endl;
for(int c=0; c<nc; c++){cout<<pbest[c]<<endl;}
cout<<"Fitness of the best elephant (solution):"<<fbest<<endl;
getchar();
return 0;
}
```

Listing 12.2
Source-code of EHO algorithm in C++.

12.5 Step-by-step numerical example of EHO algorithm

Example 3 *Find the global minima of Rastrigin function $f(x)$*

$$f(x_j) = A \times N + \sum_{c=1}^{N} [x_{j,c}^2 - A\cos(2\pi x_{j,c})] \tag{12.5}$$

where, $A = 10$ and $-5.12 \leq x_{j,c} \leq 5.12$

Solution: Here, x_j is a N-dimensional column vector. To realise the complexity of this function $f(x)$, a contour plot is presented in Fig. 12.2 which shows that $f(x)$ is a non-convex function. Finding of global minima, with conventional approaches, is fairly difficult due to its large search space and high number of multiple minimum values. Listing 12.3 presents the Matlab code to determine the function at any x.

```
function [Fit] = OF(x)
N = length(x);
A = 10;
for c = 1:N
ff(c) = x(c)^2-A*cos(2*pi*x(c));
end
Fit = A*N + sum(ff);
```

Listing 12.3
Definition of Restrigin function $OF(.)$ in Matlab.

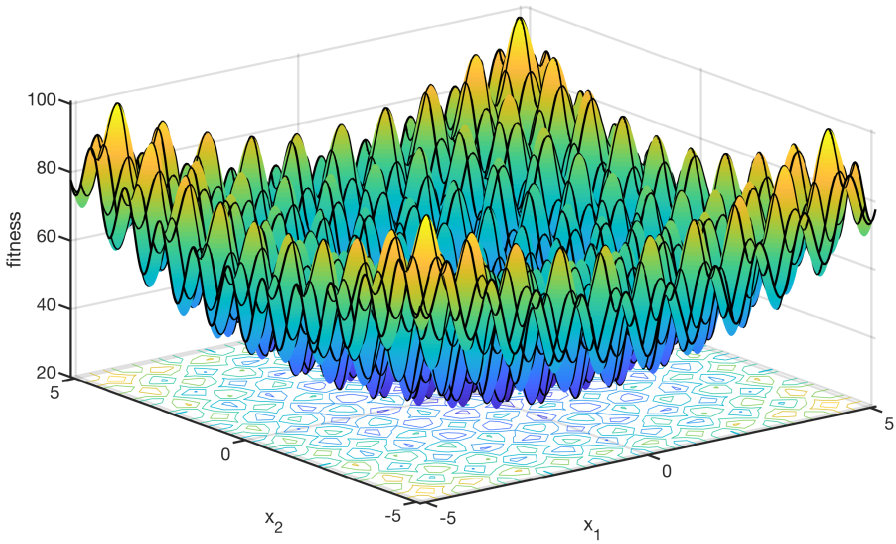

FIGURE 12.2
Contour plot of Restrigin function for $N = 2$.

Now, we demonstrate how the EHO algorithm can applied to determine the optimal minima of the Rastrigin function.

In the first step, let's assume that the value of dimension $N = 4$ which is analogous to number of clans. Each element of the vector, i.e., $x_{j,c}$ $\forall j, c$, is analogous to an elephant in clan c.

In the second step, we assumed that the population size or number of elephants in a clan, $n_c = 5$ and values of $\alpha = 0.5$ and $\beta = 0.1$. For this function, the maximum and minimum bounds of elephants in all clans will be $x_{max} = [5.12, 5.12, 5.12, 5.12]$ and $x_{min} = [-5.12, -5.12, -5.12, -5.12]$ respectively.

In the third step, we generate random but feasible (predefined boundaries) population (positions) of $n_c = 5$ elephants, as presented below:
$x_1 = \{-2.542\ \ -2.146\ 1.199\ \ -2.403\}$

$x_2 = \{3.322\ 4.943\ 2.358\ -1.599\}$
$x_3 = \{0.870\ -4.016\ 4.161\ 3.888\}$
$x_4 = \{3.254\ -2.450\ 0.966\ -4.890\}$
$x_5 = \{-0.765\ -1.918\ -3.466\ -3.290\}$

In the fourth step, we determine the value of function $f(x_j)\ \forall j$ by using OF(.). For $x_1 = \{-2.542\ -2.146\ 1.199\ -2.403\}$, it is determined as
$OF(x_1) = A.N + \sum_{c=1}^{N}[x_{1,c}^2 - A\cos(2\pi x_{1,c})] = 10 \times 4 + (-2.542)^2 - 10 \times \cos(2\pi \times (-2.542)) + (-2.146)^2 - 10 \times \cos(2\pi \times (-2.146)) + (1.199)^2 - 10 \times \cos(2\pi \times (1.199)) + (-2.403)^2 - 10 \times \cos(2\pi \times (-2.403)) = 66.903$
Similarly, we repeat the computation for all individuals or elephants, as presented below:
$Fit_1 = OF(x_1) = 66.903$
$Fit_2 = OF(x_2) = 92.995$
$Fit_3 = OF(x_3) = 59.591$
$Fit_4 = OF(x_4) = 73.720$
$Fit_5 = OF(x_5) = 69.718$

In the fifth step, the best (an individual with smallest fitness value, Fit_{best}) and worst (a individual with highest fitness value, Fit_{worst}) elephants are identified, as our goal is to minimize the function. It is observed that elephants x_3 and x_2 are the best and worst elephants respectively. Therefore, $x_{best} = \{0.870\ -4.016\ 4.161\ 3.888\}$ with $Fit_{best} = Fit_3 = 59.591$, and $x_{worst} = \{3.322\ 4.943\ 2.358\ -1.599\}$ with $Fit_{worst} = Fit_2 = 92.995$.

In the sixth step, the generation loop of the algorithm starts and the termination criteria is checked whether the number of generations has crossed its maximum allowed limit. If yes, we jump to step thirteen. If no, we move to the seventh step.

In the seventh step, the population of common elephants '1', '4', and '5' is updated by using (12.1), as illustrated below.
$x_{j,c}^{new} = x_{j,c} + \alpha.(x_{best,c} - x_{j,c}).r$
Let's assume that the value of random number $r = \{0.671\ 0.459\ 0.036\ 0.901\}$ then elephant '1' is updated as
$x_{1,1}^{new} = -2.542 + (0.5).(0.870 + 2.542).(0.671) = -1.397$
$x_{1,2}^{new} = -2.146 + (0.5).(-4.016 + 2.146).(0.459) = -2.575$
$x_{1,3}^{new} = 1.199 + (0.5).(4.161 - 1.199).(0.036) = 1.252$
$x_{1,4}^{new} = -2.403 + (0.5).(3.888 + 2.403).(0.901) = 0.431$
Therefore, $x_1^{new} = \{-1.397\ -2.575\ 1.252\ 0.431\}$. Similarly, x_4^{new} and x_5^{new} are calculated and presented below.
$x_4^{new} = \{2.242\ -3.181\ 2.050\ -1.564\}$
$x_5^{new} = \{-0.158\ -2.330\ -0.966\ -2.676\}$
The position of the best elephant is updated by using (12.2) and (12.3) where, $x_{center,c}$ is calculated as

$x_{center,1} = \frac{\sum_j^{n_c} -2.542+3.322+0.870+3.254-0.765}{5} = 0.828;$
$x_{center,2} = \frac{\sum_j^{n_c} -2.146+4.943-4.016-2.450-1.918}{5} = -1.117;$
$x_{center,3} = \frac{\sum_j^{n_c} 1.199+2.358+4.161+0.966-3.466}{5} = 1.044;$
$x_{center,4} = \frac{\sum_j^{n_c} -2.403-1.599+3.888-4.890-3.290}{5} = -1.659;$
Now, we update the position of the best elephant, i.e. '3', by using (12.2) as
$x_{3,1}^{new} = (0.1).(0.828) = 0.083$
$x_{3,2}^{new} = (0.1).(-1.117) = -0.112$
$x_{3,3}^{new} = (0.1).(1.044) = 0.104$
$x_{3,4}^{new} = (0.1).(-1.659) = -0.166$
Therefore, $x_3^{new} = \{0.083\ -0.112\ 0.104\ -0.166\}$
Similarly, we update the position of the worst elephant, i.e. '2' by using (12.4), as determined below. Suppose, the random number in the equation is $rand = r_1 = \{0.706\ 0.032\ 0.277\ 0.046\}$
$x_{2,1}^{new} = -5.12 + (0.706).(5.12 + 5.12 + 1) = 2.815$
$x_{2,2}^{new} = -5.12 + (0.032).(5.12 + 5.12 + 1) = -4.760$
$x_{2,3}^{new} = -5.12 + (0.277).(5.12 + 5.12 + 1) = -2.007$
$x_{2,4}^{new} = -5.12 + (0.046).(5.12 + 5.12 + 1) = -4.603$
Thus, $x_2^{new} = \{2.815\ -4.760\ -2.007\ -4.603\}$.

In the eighth step, we compute the new fitness values of elephants for newly generated positions $x_j^{new}\ \forall j$ using $OF(.)$. The newly computed fitnesses are given below
$Fit_1^{new} = OF(x_1^{new}) = 76.424$
$Fit_2^{new} = OF(x_2^{new}) = 89.186$
$Fit_3^{new} = OF(x_3^{new}) = 10.787$
$Fit_4^{new} = OF(x_4^{new}) = 56.782$
$Fit_5^{new} = OF(x_5^{new}) = 47.613$

In step nine, the best x_{best}^{new}, and worst x_{worst}^{new}, elephants of the newly generated population are determined. From step eight, it is found that x_3^{new} and x_2^{new} are fittest (Fit_{best}^{new}) and weakest (Fit_{worst}^{new}) elephants respectively in this population. Therefore,
$x_{best}^{new} = \{0.083\ -0.112\ 0.104\ -0.166\}$
$x_{worst} = x_{worst}^{new} = \{2.815\ -4.760\ -2.007\ -4.603\}$

In the tenth step, Fit_{best}^{new} is compared with Fit_{best}; if $Fit_{best}^{new} < Fit_{best}$ then do following
$Fit_{best} = Fit_{best}^{new}$
$x_{best} = x_{best}^{new}$
If no, do nothing.

In step eleven, store $x_j^{new}\ \forall j$ into $x_j\ \forall j$ and $Fit_j^{new}\ \forall j$ into $Fit_j\ \forall j$ as
$x_1 = x_1^{new} = \{-1.397\ -2.575\ 1.252\ 0.431\}$
$x_2 = x_2^{new} = \{2.815\ -4.760\ -2.007\ -4.603\}$.

$x_3 = x_3^{new} = \{0.083\ -0.112\ 0.104\ -0.166\}$
$x_4 = x_4^{new} = \{2.242\ -3.181\ 2.050\ -1.564\}$
$x_5 = x_5^{new} = \{-0.158\ -2.330\ -0.966\ -2.676\}$
and
$Fit_1 = Fit_1^{new} = 76.424$
$Fit_2 = Fit_2^{new} = 89.186$
$Fit_3 = Fit_3^{new} = 10.787$
$Fit_4 = Fit_4^{new} = 56.782$
$Fit_5 = Fit_5^{new} = 47.613$

In the twelfth step, we return to the sixth step.

In the thirteenth step, we print the x_{best} with Fit_{best} as an optimal solution and stop the algorithm.

12.6 Conclusions

The chapter presents a simple tutorial of the EHO algorithm to solve mathematical optimization problems. The herding behaviour of elephants is presented in some set of mathematical equations along with its algorithm. In order to understand the basic programming involved in EHO, source-codes of Matlab and C++ are also provided. The problem solving ability of EHO is demonstrated by step-by-step numerical examples in which we solved the Restrigin function. We feel that this chapter will help an individual who wants to make use of the EHO algorithm to solve mathematical optimization problems.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of United Kingdom (Reference Nos.: EP/R001456/1 and EP/S001778/1).

References

1. *Elephant Social Structure*, [Available Online] https://www.elephant-world.com/elephant-social-structure/

2. N. K. Meena, S. Parashar, A. Swarnkar, N. Gupta and K. R. Niazi. "Improved elephant herding optimization for multiobjective DER accommodation in distribution systems". *IEEE Transactions on Industrial Informatics*, vol. 14(3), 2018, pp. 1029-1039.
3. C.E. O'Connell-Rodwell. "Keeping an "ear" to the ground: seismic communication in elephants". *Physiology*, vol. 22(4), pp.287-294, , 2007.
4. C. Ernst. "Understanding elephants UH scholars study wild and working animals on two continents". *Malamalama*, vol. 32(2), May 2007.
5. G. Wang, S. Deb and L. d. S. Coelho. "Elephant herding optimization," *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, Bali, 2015, pp. 1-5. doi: 10.1109/ISCBI.2015.8