Enhanced Grey Wolf Optimization Algorithm for Global Optimization

Article *in* Fundamenta Informaticae · June 2017

DOI: 10.3233/FI-2017-1539

2 authors:

Himani Joshi
G. B. Pant University of Agriculture and Technology, Pantnagar

Sankalap Arora

Fundamenta Informaticae 153 (2017) 235–264
DOI 10.3233/FI-2017-1539
IOS Press

Enhanced Grey Wolf Optimization Algorithm for Global Optimization

Himani Joshi
Research Scholar, DAV University
Jalandhar, Punjab, India
himani25joshi@gmail.com

Sankalap Arora*
Assistant Professor, DAV University
Jalandhar, Punjab, India
sankalap.arora@gmail.com

Abstract. Grey Wolf Optimizer (GWO) is a new meta-heuristic search algorithm inspired by the social behavior of leadership and the hunting mechanism of grey wolves. GWO algorithm is prominent in terms of finding the optimal solution without getting trapped in premature convergence. In the original GWO, half of the iterations are dedicated to exploration and the other half are devoted to exploitation, overlooking the impact of right balance between these two to guarantee an accurate approximation of global optimum. To overcome this shortcoming, an Enhanced Grey Wolf Optimization (EGWO) algorithm with a better hunting mechanism is proposed, which focuses on proper balance between exploration and exploitation that leads to an optimal performance of the algorithm and hence promising candidate solutions are generated. To verify the performance of our proposed EGWO algorithm, it is benchmarked on twenty-five benchmark functions with diverse complexities. It is then employed on range based node localization problem in wireless sensor network to demonstrate its applicability. The simulation results indicate that the proposed algorithm is able to provide superior results in comparison with some well-known algorithms. The results of the node localization problem indicate the effectiveness of the proposed algorithm in solving real world problems with unknown search spaces.

Keywords: Grey wolf optimizer (GWO), Global optimization, Exploration, Exploitation, Wireless sensor network, Node localization.

*Address for correspondence: DAV University, Jalandhar, Punjab, India.

Received October 2016; revised April 2017.

1. Introduction

Optimization algorithms came into existence in an attempt to find an approximate solution to different NP-hard problems like structural optimization problems, economical optimization and node localization problem in wireless sensor networks [1], even with the traditional methods as hill climbing, newton method etc. [2] [3]. Being stochastic in nature, optimization algorithms initialize random options to search the optimal result or near to the desirable optimal result [3]. On the grounds of the simplicity, flexibility, robustness and ability to avoid getting trapped in local optima, optimization algorithms may be used in an ample range of problems and real world applications such as travelling salesman problem [4] and load balancing [5]. The nature of the optimization algorithm segregates it into two chief classifications: deterministic optimization algorithms and stochastic optimization algorithms. Deterministic optimization algorithms return similar initial values, thereby producing repeated solutions. In contrast to deterministic optimization algorithms, stochastic optimization algorithms produce different results even if the initial values are same [6]. Stochastic optimization algorithms are further categorized into heuristic and meta-heuristic optimization algorithms. Meta-heuristic algorithms like Particle Swarm Optimization (PSO) algorithm [7, 8], Firefly Algorithm (FA) [9, 10, 11], Flower Pollination Algorithm (FPA) [12, 13], Cuckoo Search (CS) [14, 11], Ant Colony Optimization (ACO) algorithm [15], Butterfly Optimization Algorithm (BOA) [16, 17], Genetic Algorithm (GA) [18], Bing-Bang Big Crunch (BBBC) algorithm [19], Gravitational Search Algorithm (GSA) [20], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [21] and Self-adaptive DE (jDE) [22]. Meta-heuristic algorithms depend on prior potential due to which they work higher than simple heuristic algorithms as Simulated Annealing (SA) [23], Hill Climbing (HC) [24] that are based on hit and trial method. Heuristic algorithms may get trapped in local optima through non-availability of background knowledge. Exploration and exploitation are two main concepts on which performance of optimization algorithm depends. As the name suggests exploration means to explore the promising space to get better solution among the solution set whereas exploitation means to exploit the solutions achieved from exploration phase. The stability in exploration and exploitation helps to converge to global optima [25].

Grey Wolf Optimization (GWO) algorithm is a stochastic meta-heuristic optimization algorithm that simulates the behavior of grey wolves (*Canis lupus*). Grey wolves are apex predators and are at highest level of the food chain. They live and hunt in the pack that includes $5 - 12$ grey wolves [26]. GWO is proven to be superior or competitive to other classical metaheuristics such as DE, GA and PSO. The reduced number of search parameters is an important advantage of GWO algorithm which is reflected in various applications such as optimization of controller gain [27], optimal power flow problem [28], optimal tuning of PID fuzzy controllers [29], reactive power dispatch [30], hyperspectral selection of band [31]. Despite these advantages, these metaheuristic algorithms can easily get trapped in the local optima when solving complex multimodal problems. Therefore, hybridisation of one or more local search approaches is done in order to avoid the local optima problem and to accelerate the convergence speed of these algorithms [12, 17, 32, 33, 34, 35, 36, 37]. Though GWO algorithm avoids the dilemma of getting trapped in local minima, still, it has to cope up the limitation of providing proper exploration and exploitation trade-off. Some efforts have been made to improve the performance of GWO in the past. A Hybrid Multi-objective Grey Wolf Optimizer was developed for solving the dilemma of dynamic welding scheduling problems [38, 39]. Hybridized algorithm was designed to improve the balance of exploration and exploitation. There are different optimization tasks that are solved using GWO algorithm. The effectiveness of GWO algorithm is achieved in terms of solution quality and convergence behavior [40, 41]. Improved Grey Wolf Optimizer algorithm (IGWO) [42] follows a new strategy to update the position of search agents. The adaptive values of parameters in GWO caused the transition in between the exploration and exploitation thereby making a need to improve the basic GWO algorithm. Therefore, IGWO concentrates in exploration and exploitation trade-off. Modified Grey Wolf Optimizer algorithm (mGWO) [43] engages the exponential function for atrophy of a parameter over a course of iterations making improvement in the convergence rate. In this paper, an Enhanced Grey Wolf Optimization (EGWO) algorithm is proposed to provide improved performance as compared to GWO algorithm. The focal point of the proposed algorithm is to have better convergence rate with better results. Enhancements are made so as to fill the gap of an existing algorithm. In proposed EGWO algorithm, decisions and selections followed by grey wolves during hunting the prey are modulated along with parameter liable for exploration and exploitation. To evaluate and validate the performance of proposed algorithm, five meta-heuristic algorithms such as GWO, PSO, FA, CMA-ES and jDE are considered for comparison in this study. Later, it is applied in Wireless Sensor Network (WSN) for range based position estimation of un-localized nodes. The organization of the paper is presented as: Section 2 discusses the GWO algorithm. In Section 3, the proposed EGWO algorithm is outlined. In Section 4, simulations and comparisons on benchmark functions are presented. In Section 5, application of node localization in WSN is presented. Section 6 concludes the work and future work is suggested.

2. Grey wolf optimizer

GWO algorithm is a new metaheuristic algorithm developed by Mirjalili et al. in 2014 [26]. It is a population-based stochastic algorithm for finding the optimal result from the solution set (population). GWO algorithm is inspired from grey wolves that belong to Canidae family which simulates the behavior of leadership quality and the social hunting mechanism of grey wolves in three steps as tracking, encircling and attacking [44]. There are particularly four types of grey wolves namely alpha (α), beta (β), delta (δ) and omega (ω) having a strict social dominant hierarchy as shown in Fig. 1.

The α grey wolf is a dominant grey wolf making decisions regarding sleeping time, hunting that is adopted by other submissive grey wolves. The second level in the hierarchy is considered to be for β grey wolf that helps α to make decisions. It follows the commands of α and dominates other grey wolves. Once α becomes old or passes away, the next most eligible grey wolf to become α is β. δ is the third level in a hierarchy of grey wolves that listens and submit to α or β wolves but dominate ω wolves. If the wolf is neither α nor β and also not δ then it is ω which is the lowest level in the hierarchy of grey wolves that submit to other dominant grey wolves [26]. Including the social hierarchy of wolves, team hunting is another intriguing social habit of grey wolves. Main hunting steps followed by grey wolves are given below.

- Approaching the prey.

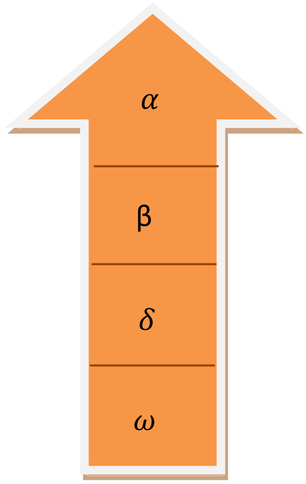

Figure 1. Hierarchy of grey wolf

- Encircling and harassing the prey till the time it stops moving.
- Attacking the prey.

2.1. Mathematical model

The social team hunting behavior of grey wolves is mathematically modeled with the help of the fittest wolf α, β and δ wolves in order to obtain the optimum solution. ω wolves follow other dominant wolves for the hunting mechanism.

2.1.1. Encircling the prey

For hunting, grey wolves chase and encircle the prey. Mathematically, it is modeled as given in Eq. (1) and (2):

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X_p}(t) - \overrightarrow{X}(t)| \tag{1}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{2}$$

here t is the indicated current iteration and $t+1$ represents next iteration. $\vec{X}$is the position vector of grey wolf and $\vec{X_p}$ is the position vector of prey. $\vec{A}$ and $\vec{C}$ are coefficient vectors where they are depicted as:

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r_1} - \overrightarrow{a} \tag{3}$$

$$\overrightarrow{C} = 2.\overrightarrow{r_2} \tag{4}$$

where $\vec{r_1}$ and $\vec{r_2}$ are the random vectors in the range of $[0, 1]$ and components of $\vec{a}$ are linearly decreased from 2 to 0 over the courses of iterations [26].

2.1.2. Hunting

After the location of the prey is estimated, grey wolves encircle it for hunting. Hunting mechanism is guided by α, β, δ grey wolves. In the entire search space, there is no idea about the location of the prey, but an assumption is made that α, β and δ wolves have knowledge about the prey's location. Therefore, the three best solutions obtained are saved and other agents, i.e., ω grey wolves keep updating their own positions according to the best solutions.

Hunting process is mathematically guided using Eq. (7) which is derived from Eq. (5) and (6).

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \overrightarrow{X}|, \overrightarrow{D_\beta} = |\overrightarrow{C_2}.\overrightarrow{X_\beta} - \overrightarrow{X}|, \overrightarrow{D_\delta} = |\overrightarrow{C_3}.\overrightarrow{X_\delta} - \overrightarrow{X}| \tag{5}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha}(t) - \overrightarrow{A_1}.\overrightarrow{D_\alpha}, \overrightarrow{X_2} = \overrightarrow{X_\beta}(t) - \overrightarrow{A_2}.\overrightarrow{D_\beta}, \overrightarrow{X_3} = \overrightarrow{X_\delta}(t) - \overrightarrow{A_3}.\overrightarrow{D_\delta} \tag{6}$$

$$\overrightarrow{X}(t+1) = (\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3})/3 \tag{7}$$

Exploration and exploitation are handled while searching and attacking the prey using parameters $\vec{a}$ and $\vec{C}$. Parameter $\vec{a}$ is decreased from 2 to 0 so as to balance exploration and exploitation. Grey wolves diverge from each other to search for prey when $|\vec{A}| > 1$ and when $|\vec{A}| < 1$ converge towards each other so as to attack. Randomness helps to avoid getting trapped in the local minima.

3. Enhanced grey wolf optimization

The outcome of the existing GWO algorithm is an optimal solution that avoids getting trapped in local optima. But still, the problem of exploration and exploitation trade-off mechanism needs to be improved. Therefore, in order to overcome this problem of existing GWO algorithm, an improved algorithm, i.e., Enhanced Grey Wolf Optimization (EGWO) algorithm is proposed. In proposed EGWO algorithm different advancements are done so as to find a solution to problems of efficiency and convergence rate. The $\overrightarrow{a}$ parameter is considered as a random vector in the range $[0, 1]$ whose values are crucial in providing balance between exploration and exploitation. Adaptive values of $\overrightarrow{a}$ parameter maintain exploration to prevent getting trapped in local optima and cope up the accuracy problem. It is a critical parameter in fine-tuning of solution vectors, and can also be potentially used in adjusting the convergence rate of the algorithm. With the intention to simulate the hunting behavior of grey wolves an assumption is made that the fittest wolf α has better knowledge about the potential position of prey, and for this reason only the fittest solution is saved. The global best solution obtained from the entire population helps in achieving the global optimum. The EGWO algorithm is proposed in order to provide enhanced performance in terms of avoiding getting trapped in pre-mature convergence, convergence rate and accuracy. Hunting mechanism is achieved from Eq. (8-10),

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \overrightarrow{X}|, \tag{8}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha}(t) - \overrightarrow{A_1}.\overrightarrow{D_\alpha}, \tag{9}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_1} \tag{10}$$

To sum up, the proposed EGWO algorithm for determining optimization problems is described below.

- The proposed EGWO algorithm simulates the hunting behavior of grey wolves, and save the best fit solution to find global optima.
- The proposed EGWO algorithm gains the position of the prey assuming that α has better knowledge about the same.
- The proposed EGWO algorithm has one parameter, i.e., $\overrightarrow{a}$ that needs to be adjusted in the range the $[0, 1]$.
- Exploration and exploitation are balanced by parameter $\overrightarrow{a}$.

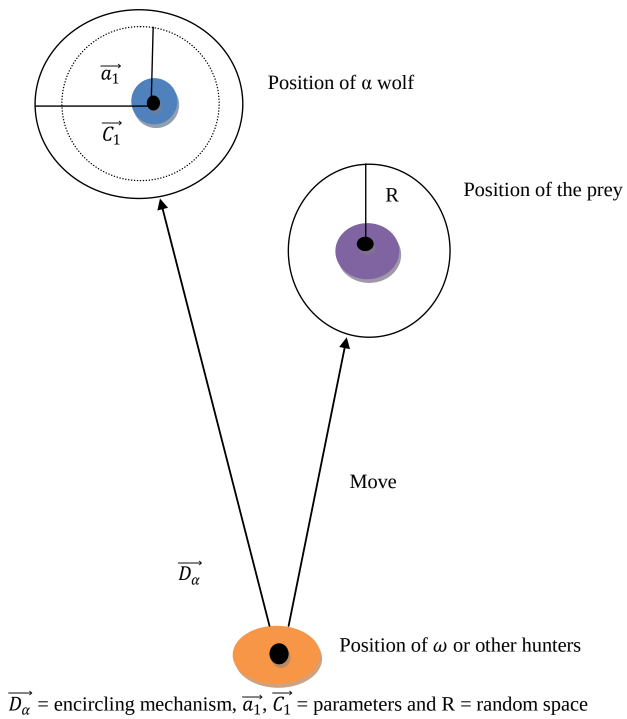

Figure 2. Position updating in proposed EGWO algorithm

Fig. 2 spectacles, how the whole wolf pack updates its position according to the position of prey estimated in random space by α grey wolf and the flowchart for proposed EGWO is presented in Fig. 3.

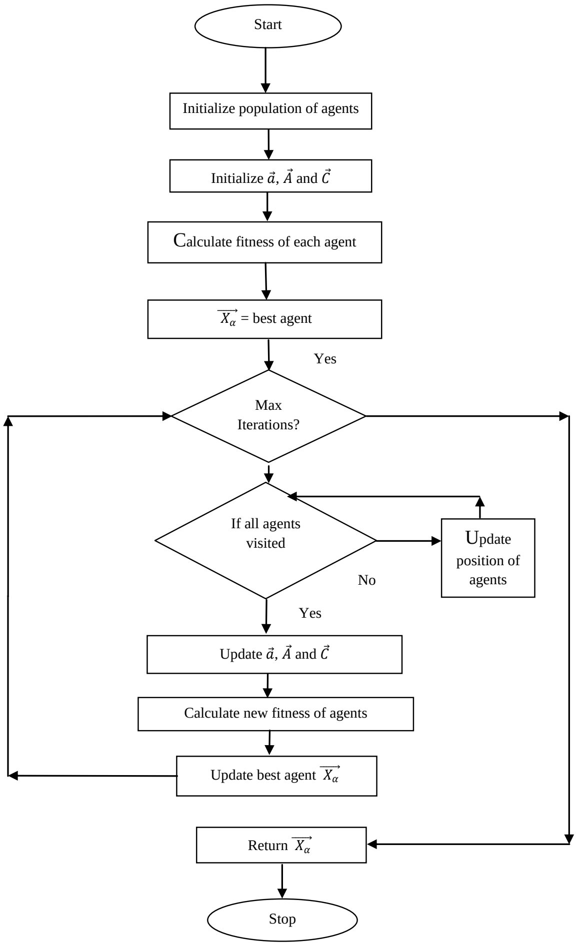

Figure 3. Flowchart of proposed EGWO algorithm

4. Simulation results and discussion

Every novel optimization algorithm must be validated on benchmark functions and compared with other standard optimization algorithms. There is no standard set of benchmark functions available; however various benchmark functions that are recommended are used in this study [45]. In this paper, the proposed EGWO algorithm is benchmarked using twenty-five different benchmark functions. The benchmark functions which are selected in this study are chosen in such a way that the proposed algorithm is tested on almost all types of problems. Considering this viewpoint a diverse subset of benchmark functions is chosen in this study. This subset can be classified into four major categories [46]. In the first category, function can either have single optima or multiple optima. In second category, the number of dimensions can be low or high. High dimensional problems are very difficult to solve that's why most of the benchmark functions used in this work are high dimensional. Another category is that some functions are separable and some are non-separable. In the last category, functions with noisy data are used. These functions are alike real world problems which contains noisy data which makes them difficult to solve. New algorithms must be tested on all these kinds of test functions in order to properly validate and demonstrate the efficiency of the algorithm. These benchmark functions are described in Table 1 where dim specifies the dimensions of the test functions and $range$ indicates the search space between lb and ub which is the specified range of variables. $Optima$ indicate the global optima that the algorithm searches. The results of proposed EGWO algorithm along with other famous standard optimization algorithms viz. GWO, FA, PSO, CMA-ES and jDE are reported in Table 2.

Table 1. Benchmark functions used in the current study

S. No.	Benchmark function	Type	Formula	Dim	Range	Optima
f_1	Sphere	M, S	$f(x)=\sum_{i=1}^{n} x_i^2$	30	(-100,100)	0
f_2	Step	U, S	$f(x)=\sum_{i=0}^{n-1}(\lfloor x_i \rfloor+0.5)^2$	30	(-100,100)	0
f_3	Quartic function with noise	U, S	$f(x)=\sum_{i=0}^{n-1} x_i^4+N(0,1)$	30	(-1.28,1.28)	0
f_4	Alpine	M,S	$f(x)=\sum_{i=0}^{n}\lvert x_i sin(x_i)+0.1x_i\rvert$	30	(-10,10)	0
f_5	Rosenbrock	U,N	$f(x)=\sum_{i=1}^{n-1} 100(x_i-x_{i-1}^2)^2+(1-x_{i-1})^2$	30	(-30,30)	0
f_6	Beale	U, N	$f(x)=(1.5-x_0+x_0x_1)^2+(2.25-x_0+x_0{x_1}^2)^2+(2.625-x_0+x_0{x_1}^2)^2$	2	(-4.5,4.5)	0
f_7	Cigar	U, N	$f(x)=x_0^2+\sum_{i=1}^{n} x_i^2$	30	(-10,10)	0
f_8	Schwefel 2.22	U, N	$f(x)=\sum_{i=0}^{n-1}\lvert x_i\rvert+\prod_{i=0}^{n-1}\lvert x_i\rvert$	30	(-10,10)	0
f_9	Goldstein Price	M, N	$f(x)=[1+(x_0+x_1+1)^2(19-14x_0+3x_0^2-14x_1+6x_0x_1+32x_1^2]\times[30+(2x_0-3x_1)^2(18-32x_0+12x_0^2+48x_1-36x_0x_1+27x_1^2)]$	2	(-2,2)	3
f_{10}	Matyas	U, N	$f(x)=0.25(x_0^2+x_1^2)-0.48x_0x_1$	2	(-10,10)	0
f_{11}	Schwefel 2.26	M, S	$f(x)=-\sum_{i=0}^{n-1} x_i sin\sqrt{x_i}$	30	(-500,500)	-12569.5

Table 1. Benchmark functions used in the current study contd.

S. No.	Benchmark function	Type	Formula	Dim	Range	Optima
f_{12}	Schwefel 1.2	U, N	$f(x)=\sum_{i=0}^{n-1}\left\{\sum_{j=0}^{j<i} x_i\right\}^2$	30	(-100,100)	0
f_{13}	Michalewitz	M, S	$f(x)=-\sum_{i=0}^{n} sin(x_i)sin^{20}\left(\frac{ix_i^2}{\pi}\right)^{2m}$; $m=10$	30	(0, π)	-0.966n
f_{14}	Bohachevsky	M, N	$f(x)=\ x_i^2+2.0x_{i+1}^2-\ 0.3\cos(3\pi x_i)\cos(4\pi x_{i+1})+0.7$	2	(-100,100)	0
f_{15}	Rastrigin	M, S	$f(\vec{x})=\sum_{i=1}^{n}(x_i^2-10cos(2\pi x_i)+10)$	30	(-5.12,5.12)	0
f_{16}	Booth	U, N	$f(x)=(x_0+2x_1-7)^2+(2x_0+x_1-5)^2$	2	(-10,10)	0
f_{17}	Easom	M, S	$f(x)=-cos(x_0)cos(x_1)exp(-(x_0-\Omega)^2-(x_1-\Omega)^2)$	2	(-100,100)	-1
f_{18}	Schaffer	M, N	$f(x)=(x_0^2+x_1^2)^{\frac{1}{4}}(50(x_0^2+x_1^2)^{0.1}+1)$	2	(-100,100)	0
f_{19}	Sum Squares	U, S	$f(x)=\sum_{i=0}^{n-1} ix_i$	30	(-10,10)	0
f_{20}	Griewank	M,N	$f(x)=\frac{1}{4000}\sum_{i=1}^{n-1}(x_i-100)^2-\prod_{i=1}^{n-1}cos\left(\frac{x_i-100}{\sqrt{i-1}}\right)+1$	30	(-600,600)	0
f_{21}	Leon	U,N	$f_x=100(x_{i+1}-x_i^3)^2+(x_i-1.0)^2$	30	(-1,2,1.2)	0
f_{22}	Ackley	M,N	$f_x=-20e^{-0.02}\sqrt{D^{-1}\sum_{i=1}^{D}x^2-e^{D^{-1}}\sum_{i=1}^{D}\cos(2\pi x_i)+20+e}$	30	(-32,32)	0
f_{23}	Zettl	U,N	$f_x=(x_i^2+x_{i+1}^2-2x_i)^2+0.25x_i$	30	(-5,5)	-0.00379
f_{24}	Trid	M,N	$f_x=\left[\sum_{i=0}^{D}(x_i-1)^2\right]-\left[\sum_{i=1}^{D}x_ix_{i-1}\right]$	30	$(-D^2,D^2)$	-1500
f_{25}	Schwefel 2.21	U,N	$f_x=max(\lvert x_i\rvert)$	30	(-10,10)	0

Implementation of the proposed EGWO algorithm and other algorithms is done in QT Creator 2.4.0. Every algorithm is executed for 25 independent runs with a fixed population size of 30. As meta-heuristic algorithms are stochastic in nature, same results are not produced due to which statistical results are calculated. For PSO, parameters considered for fast convergence rate are $\omega = 0.7$ and $c_1 = c_2 = 1.494$ [47]. The parameters studied for FA are $\alpha = 0.25$, attractiveness $\beta_0 = 1$ and absorption coefficient $\gamma = 0.20$ [9] and for GWO, $\vec{r_1}$, $\vec{r_2}$ are in the range [0, 1], $\vec{a}$ that decreases from 2 to 0 [26]. The CMA-ES optimization algorithm does not require tedious tuning of the parameter. The strategy internal parameter is automated completely. D, λ, μ, σ_{start} and C_{cov} are the parameters of the CMA-ES algorithm that needs to be calculated [21]. While in jDE optimization algorithm there are two parameters that need to be adjusted are F and CR for obtaining consistent performance. Using jDE optimization algorithm each individual has its own F and CR control parameters [22]. Parameters $\vec{a}$, $\vec{r_1}$, $\vec{r_2}$ are taken as random vectors in range $[0, 1]$ for the proposed EGWO algorithm. The number

Table 2. Results of benchmark functions

Function		EGWO (Proposed)	GWO	PSO	FA	CMA-ES	jDE	*p*-values+
f_1	Mean	**1.174E-36**	2.880E-26	9.287E-03	3.864E+08	8.15E-11	3.73E-01	3.130E-02
	Std. Dev	1.052E-36	3.810E-26	2.598E-02	1.924E+08	2.74E-11	2.75E-01	
f_2	Mean	**0.000E+00**	2.166E-01	4.860E-05	6.000E-01	8.69E-11	**0.00E+00**	1.250E-01
	Std. Dev	0.000E+00	1.196E-01	1.350E-05	6.633E-01	7.15E-11	0.00E+00	
f_3	Mean	**4.272E-05**	8.545E-05	6.104E-05	1.004E-01	8.86E-01	2.73E-01	3.130E-02
	Std. Dev	1.495E-05	8.929E-05	8.074E-05	3.451E-02	2.79E-01	1.48E-01	
f_4	Mean	1.451E+02	7.434E+00	2.079E+01	1.800E+00	**8.53E-06**	6.04E-02	3.130E-02
	Std. Dev	6.295E+01	1.022E+01	3.240E+00	2.750E+00	1.89E-06	3.37E-02	
f_5	Mean	4.859+01	2.387E+01	7.920E+06	2.730E+01	**1.72E+01**	3.34E+01	3.130E-02
	Std. Dev	9.442+00	2.713E-01	3.530E+06	1.813E+00	4.92E-01	9.04E+00	
f_6	Mean	2.669E-02	1.241E-02	3.690E-03	**0.000E+00**	**0.000E+00**	**0.000E+00**	2.500E-01
	Std. Dev	1.726E-02	7.436E-02	6.461E-03	0.000E+00	0.000E+00	0.000E+00	
f_7	Mean	**1.983E-37**	2.050E-28	2.474E+04	3.540E-05	1.01E-12	3.65E-06	2.500E-01
	Std. Dev	2.705E-37	1.730E-28	9.697E+03	6.530E-05	3.94E-13	3.46E-06	
f_8	Mean	**2.836E-23**	3.080E-16	3.695E+01	3.845E-03	1.791E-05	1.65E-04	1.250E-01
	Std. Dev	1.470E-23	4.293E-16	5.861E+00	1.812E-03	5.640E-06	1.14E-04	
f_9	Mean	1.962E+01	3.915E+00	4.208E+00	5.700E+00	**3.000E+00**	**3.000E+00**	3.130E-02
	Std. Dev	9.664E+00	4.895E-01	5.059E+00	8.100E+00	0.000E+00	0.000E+00	
f_{10}	Mean	6.072E-150	1.170E-126	5.748E-03	**0.000E+00**	3.68E-95	1.18E-77	1.000E+00
	Std. Dev	1.214E-149	2.080E-126	1.440E-02	0.000E+00	0.000E+00	1.21E-77	
f_{11}	Mean	-6.282E+03	-5.093E+03	-6.081E+02	-6.187E+03	**-1.35E+30**	-1.45E+04	3.130E-02
	Std. Dev	3.596E+02	2.232E+02	9.864E+01	1.216E+03	3.29E+27	0.000E+00	
f_{12}	Mean	**6.189E-05**	2.450E-04	3.565E+02	1.911E+01	4.750E-04	3.512E-03	3.130E-02
	Std. Dev	1.205E-04	2.260E-04	5.253E+02	1.031E+01	2.379E-09	3.204E-04	
f_{13}	Mean	**-9.044E+00**	-8.337E+00	-6.380E+00	-8.386E+00	-6.41E+00	-8.66E+00	3.130E-02
	Std. Dev	6.401E-01	7.550E-01	8.090E-01	8.886E-01	3.45E-01	0.000E+00	
f_{14}	Mean	2.876E+02	3.323E+02	5.811E+03	**0.000E+00**	**0.00E+00**	5.55E-17	2.500E-01
	Std. Dev	5.893E+02	1.518E+02	2.946E+03	0.000E+00	0.00E+00	0.00E+00	
f_{15}	Mean	1.349E+02	1.880E+02	2.563E+02	2.647E+01	1.63E+02	**2.06E+01**	3.130E-02
	Std. Dev	1.274E+01	5.290E+01	2.797E+01	8.744E+00	6.37E+00	4.66E+00	
f_{16}	Mean	5.734E-07	4.447E+03	2.112E-02	**0.000E+00**	**0.000E+00**	**0.000E+00**	2.500E-01
	Std. Dev	4.429E-07	5.945E+02	1.684E-02	0.000E+00	0.000E+00	0.000E+00	
f_{17}	Mean	**-1.583E+00**	-1.376E+00	-7.879E-01	-8.000E-01	-1.00E+00	-1.00E+00	3.130E-02
	Std. Dev	4.766E-01	4.632E-01	3.869E-01	4.000E-01	0.000E+00	0.000E+00	
f_{18}	Mean	4.573E-52	5.870E-29	3.239E+02	4.497E-02	**1.97E-58**	4.47E-27	5.000E-01
	Std. Dev	6.518E-52	1.050E-28	1.160E+02	1.336E-02	1.32E-58	7.55E-26	

\+ This *p*-value is generated by the Wilcoxon signed rank test at 5% significance level

Table 2. Results of benchmark functions contd.

	Function	EGWO (Proposed)	GWO	PSO	FA	CMA-ES	jDE	*p*-values+
f_{19}	Mean	2.21E-04	4.38E-01	5.11E+03	3.93E-02	**1.28E-11**	2.48E-07	6.25E-02
	Std. Dev	2.65E-04	2.49E-01	2.28E+03	5.58E-02	2.63E-12	1.30E-07	
f_{20}	Mean	1.31E-08	9.30E-01	2.32E-01	**0.00E+00**	6.09E-10	4.66E-08	6.25E-02
	Std. Dev	3.12E-08	3.20E-01	1.82E-01	0.00E+00	2.52E-10	2.21E-08	
f_{21}	Mean	1.11E-05	1.53E-03	**0.00E+00**	**0.00E+00**	3.79E-07	**0.00E+00**	2.50E-01
	Std. Dev	7.39E-06	1.96E-03	0.00E+00	0.00E+00	5.29E-07	0.00E+00	
f_{22}	Mean	1.72E+00	1.72E+00	1.72E+00	**1.31E-03**	3.15E-06	1.72E+00	3.13E-02
	Std. Dev	0.00E+00	0.00E+00	0.00E+00	1.20E-04	1.22E-06	0.00E+00	
f_{23}	Mean	**-3.79E-03**	**-3.79E-03**	**-3.79E-03**	**-3.79E-03**	**-3.79E-03**	**-3.79E-03**	3.13E-02
	Std. Dev	4.00E-09	1.79E-08	4.34E-19	0.00E+00	0.00E+00	0.00E+00	
f_{24}	Mean	-1.52E+05	-4.29E+05	-2.02E+05	-2.34E+06	-4.93E+02	**-4.24E+07**	3.13E-02
	Std. Dev	2.91E+04	7.01E+04	0.00E+00	1.08E+06	0.00E+00	4.08E+07	
f_{25}	Mean	**2.43E-08**	9.28E-04	4.97E-01	2.89E-02	1.69E-05	5.24E-01	3.13E-02
	Std. Dev	4.08E-08	2.04E-03	1.74E-01	4.59E-02	3.88E-06	2.57E-01	

+ This *p*-value is generated by the Wilcoxon signed rank test at 5% significance level

of iterations is fixed to 500 for computing the results on different optimization algorithms. The mean and standard deviation values obtained by EGWO and other metaheuristic algorithms on different benchmark functions are listed in Table 2, with the p-values generated by the Wilcoxon signed rank test at 5% significance level. The mean values in bold font indicate superiority, while the p-values suggest that the Wilcoxon test reject the hypothesis that all six algorithms are performing similarly in the corresponding benchmark function.

Results in Table 2 showed that the proposed EGWO algorithm performed superior than other considered algorithms on ten out of twenty-five benchmark functions. PSO and FA algorithms performed better on two and eight benchmark functions respectively, while GWO algorithm performs best for a single benchmark functions in total. However, the CMA-ES optimization algorithm performs better on ten benchmark functions and jDE algorithm performs superior on eight benchmark functions. Keeping the parameter $\vec{a}$ as a random vector in the range [0, 1] and altering the hunting mechanism, the balance between exploration and exploitation is augmented and the problem of slow convergence rate and less efficiency is removed and hence the results are improved. This indicates that the proposed EGWO algorithm is better than the existing GWO and other efficient optimization algorithms. The proposed EGWO algorithm shows better trade-off between exploration-exploitation in comparison to basic GWO algorithm, making it more efficient in terms of providing global optima. Furthermore, the promising meta-heuristic optimization algorithm CMA-ES performs well on some benchmark functions, but the jDE optimization algorithm do not function well and convergences to local optima resulting in less optimized solution. The results of benchmark functions verified that the proposed EGWO can yield more accurate and stable results than those of GWO, FA and PSO for solving majority of the problems when compared to other standard optimization algorithms. Multimodal and unimodal benchmark functions are used for verifying the ability of exploration and exploitation of optimization algorithms. The proposed EGWO algorithm performs better and provides competitive results than other optimization algorithms as PSO, FA, GWO, CMA-ES and jDE on unimodal benchmark functions. Proposed EGWO algorithm outperforms and gives superior performance on the majority of multimodal benchmark functions as shown in Table 2. In order to investigate the convergence rate, sudden changes are expected in the initial steps of search agents. This demonstrates that the algorithm is set to explore the search space. Further, at the end phase of optimization, the changes should decrease to constant focusing on exploitation. Fig. 4-8 shows this behavior as abrupt changes decrease with increase in iterations showing that the proposed EGWO algorithm converges to global optima. The number of iterations is set to 100 in order to show the convergence rate of the proposed EGWO algorithm in comparison to other standard meta-heuristic optimization algorithms with better precision in graphs. Results in Fig. 4 - 8 reveals, convergence rate achieved using different benchmark functions that are applied to various standard meta-heuristic optimization algorithms. The proposed EGWO algorithm performs in a preferable way in terms of finding the optimal solution without getting trapped in local optima at a high convergence rate.

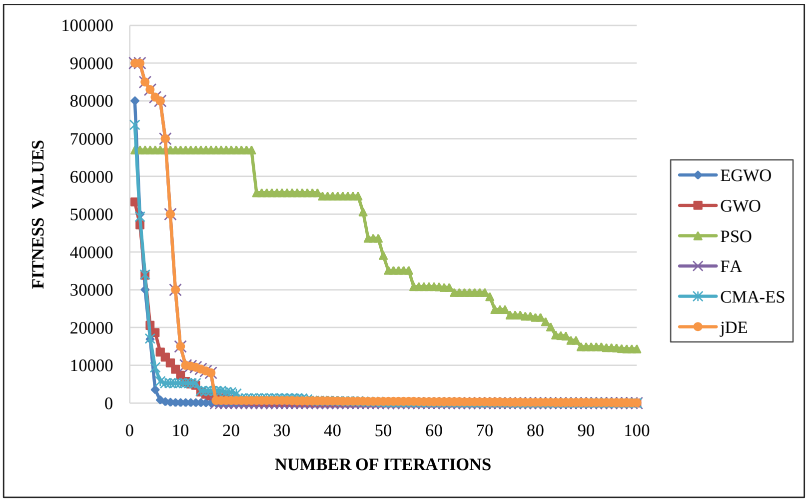

Figure 4. Fitness versus Number of iterations for f_1 Sphere function

From Fig. 4 convergence rate of proposed EGWO algorithm and other five meta-heuristic optimization algorithm viz. PSO, FA, GWO, CMA-ES and jDE using f_1 Sphere benchmark function is exhibited. Global optima for this function is 0. It is noticed that the convergence rate of the proposed EGWO algorithm is fast when compared to other standard algorithms. FA optimization algorithm produced inferior values. Throughout the iterations process, the proposed EGWO algorithm do not fall in the local optima and maintain the efficiency, but other standard optimization algorithms do not provide fast convergence behavior.

Fig. 5 presents the convergence rate attained using benchmark f_3 Quartic function with noise. Global minima obtained using f_3 is 0. The graph indicates that the proposed EGWO converge fast towards optima. Due to the balance between the exploration and exploitation the local minima is

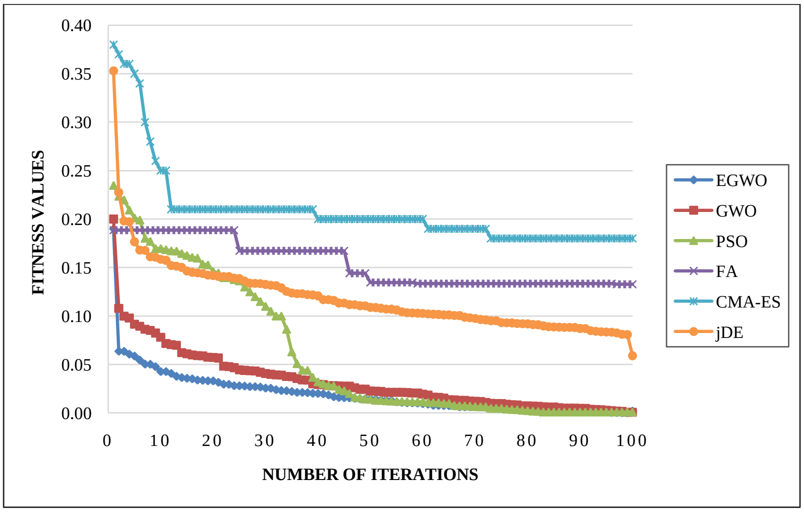

Figure 5. Fitness versus Number of iterations for f_3 Quartic function

avoided. It is seen from the graph that CMA-ES produce inferior results at the initial point and converges slowly due to the trade-off exploration and exploitation problem.

Fig 6 indicates the values obtained on f_7 Cigar benchmark function where the proposed EGWO algorithm outperforms other optimization algorithms. The convergence rate of the proposed EGWO algorithm is faster. The GWO convergence rate is more than that of PSO, FA, and jDE. Convergence

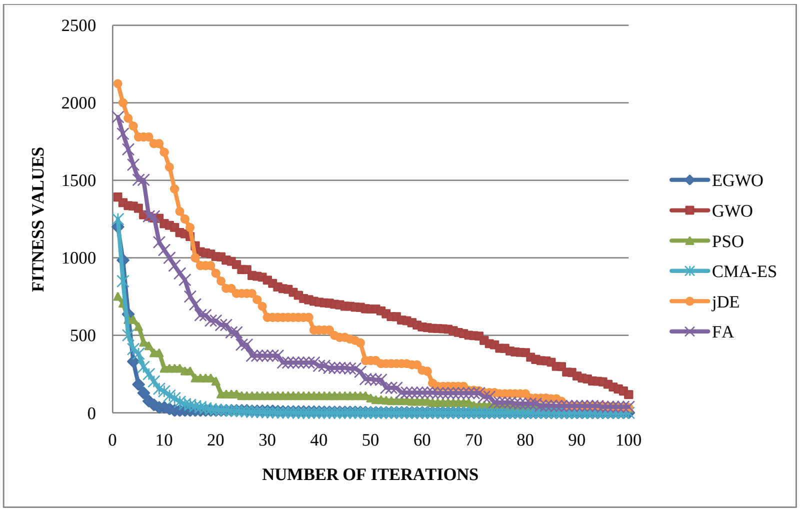

Figure 6. Fitness versus Number of iterations for f_7 Cigar benchmark function

behavior of proposed EGWO algorithm shows that the search agents extensively search the most promising area of the search space and later exploit the best solution. But the convergence rate of other optimization algorithms is not better than the convergence rate of the proposed EGWO.

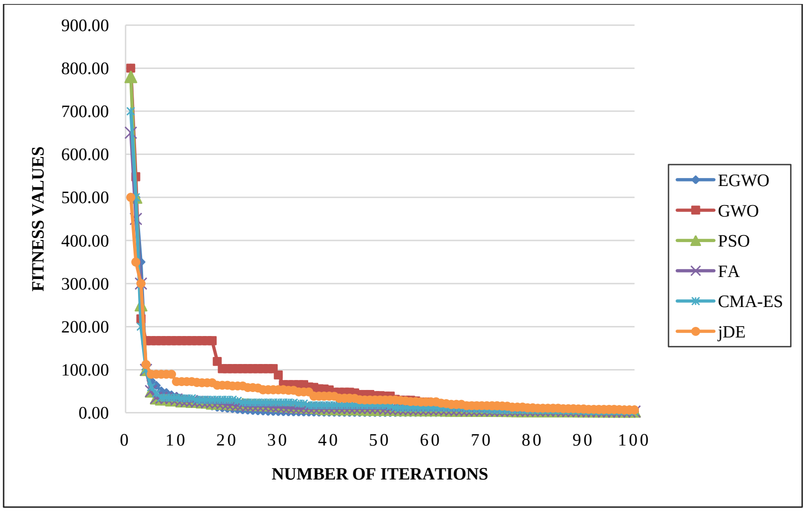

Figure 7. Fitness versus Number of iterations for f_8 Schwefel 2.22 function

From Fig. 7 the performance evaluation of proposed EGWO, PSO, GWO, FA, CMA-ES and jDE algorithms is shown using f_8 Schwefel 2.22 benchmark. From the graph, it is noticed that the

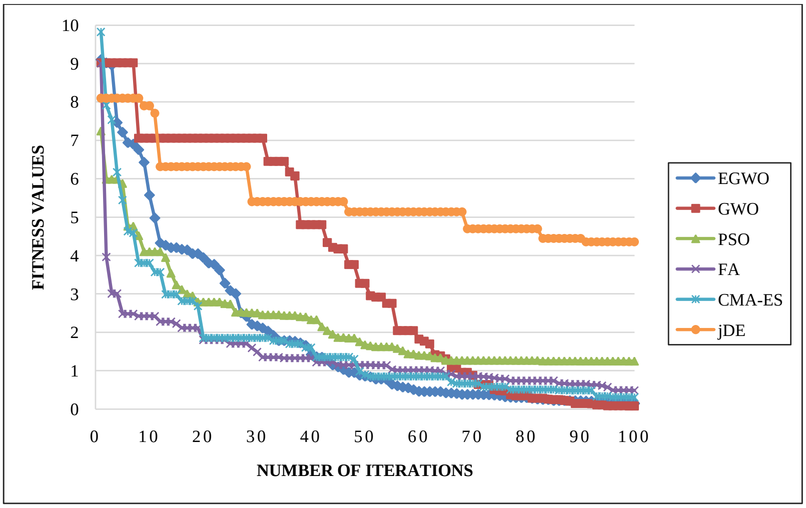

Figure 8. Fitness versus Number of iterations for f_{25} Schwefel 2.21 function

proposed EGWO algorithm is better in terms of convergence rate. CMA-ES performs better than other algorithms involving GWO in the optimization process, while EGWO performs the second best in this benchmark. FA performs much inferior than the other methods initially, while later it catches up with other approaches. EGWO slightly outperforms GWO and the other approaches. Adaptive values of the parameter $\vec{a}$ provide superior results when compared to PSO, GWO, FA, CMA-ES and jDE optimization algorithm.

From Fig. 8 it is demonstrated that the convergence values of proposed the EGWO algorithm are superior to PSO, FA, GWO, CMA-ES AND jDE algorithm on f_{25} Schwefel 2.21 function. The convergence rate of the proposed EGWO algorithm is fastest of all. Convergence of FA, PSO and CMA-ES algorithm to global optima is little late than the proposed EGWO algorithm. Throughout the iterations, half of the iterations is for exploration while the other half is for exploitation depending on the adaptive values of the parameter. Due to this mechanism proposed EGWO algorithm provide better exploration and exploitation resulting in local minima avoidance.

In addition, statistical analysis on these values obtained by the six methods on twenty five benchmark functions based on the Wilcoxon signed rank's test reveals that the differences in the obtained average and best function minima are statistically significant at the confidence level of 5% which are shown in Table 3 [48, 49, 50]. This statistical test allocates the rank while comparing the standard optimization algorithms. In Table 3, "$+$" indicates that the EGWO is better than the considered algorithm and "$-$" indicates that the EGWO is not better and "$\approx$" indicates that the EGWO has performed equivalently in comparison with the considered algorithm. The last row of Table 3 reflects the superiority of our proposed EGWO over GWO, PSO, FA, CMA-ES and jDE, as it contains 75 "$+$" signs and 12 '$\approx$' signs out of 125 comparisons. GWO beats the proposed EGWO algorithm on four functions (f_4, f_5, f_6 and f_{24}), PSO beats EGWO on five functions (f_4, f_5, f_6, f_{21} and f_{24}), FA beats EGWO on ten functions (f_4, f_5, f_6, f_{14}, f_{15}, f_{16}, f_{20}, f_{21} and f_{24}), CMA-ES beats EGWO on nine functions (f_4, f_5, f_6, f_{11}, f_{14}, f_{16}, f_{19}, f_{20} and f_{21}) while jDE beats EGWO on ten functions (f_4, f_5, f_6, f_{11}, f_{14}, f_{15}, f_{16}, f_{19}, f_{21} and f_{24}). Further the rank summary of all the algorithms is shown in Table 4. In this ranking, the algorithm with a lower ranking is regarded as the better algorithm in solving the corresponding function. If algorithms A, B, and C perform better than algorithm D, we conclude that all of A, B, and C outperform D on the particular benchmark function. However, if A outperforms B but both A and B perform similarly with C, all of A, B, and C are placed in the same rank. From Table 4 the statistical assessment results from the rank test shows that the performance of proposed EGWO algorithm is superior then GWO, PSO, FA, CMA-ES, and jDE. The proposed EGWO algorithm is ranked first among other studied algorithms. CMA-ES scored second rank, followed by jDE, FA, GWO and PSO.

At the end, results revealed that the proposed EGWO algorithm is prominent algorithm than other meta-heuristic algorithms viz. PSO, FA, GWO, CMA-ES and jDE in finding optimal solution with higher convergence rate. The proposed EGWO algorithm provides improved balance between exploration and exploitation so as to get the fast convergence rate and better accuracy than the existing GWO algorithm.

Table 3. Results of the non-parametric statistical test

	EGWO vs. GWO	**EGWO vs. PSO**	**EGWO vs. FA**	**EGWO vs. CMA-ES**	**EGWO vs. jDE**
f_1	+	+	+	+	+
f_2	+	+	+	+	≈
f_3	+	+	+	+	+
f_4	-	-	-	-	-
f_5	-	-	-	-	-
f_6	-	-	-	-	-
f_7	+	+	+	+	+
f_8	+	+	+	+	+
f_9	+	+	+	≈	≈
f_{10}	+	+	+	+	+
f_{11}	+	+	+	-	-
f_{12}	+	+	+	+	+
f_{13}	+	+	+	+	+
f_{14}	+	+	-	-	-
f_{15}	+	+	-	+	-
f_{16}	+	+	-	-	-
f_{17}	+	+	+	+	+
f_{18}	+	+	+	+	+
f_{19}	+	+	+	-	-
f_{20}	+	+	-	-	+
f_{21}	+	-	-	-	-
f_{22}	≈	≈	-	≈	≈
f_{23}	≈	≈	≈	≈	≈
f_{24}	-	-	-	+	-
f_{25}	+	+	+	+	+
Total number of + signs	**19**	**18**	**14**	**13**	**11**

Table 4. Rank summary of statistical assessment results

	EGWO	GWO	PSO	FA	CMA-ES	jDE
f_1	1	2	4	6	3	5
f_2	1.5	5	4	6	3	1.5
f_3	1	3	2	4	6	5
f_4	6	4	5	3	1	2
f_5	6	2	5	3	1	4
f_6	6	5	4	2	2	2
f_7	1	2	6	5	3	4
f_8	1	2	6	5	3	4
f_9	2	4	6	5	2	2
f_{10}	1.5	3	6	1.5	4	5
f_{11}	3	5	6	4	1	2
f_{12}	1	2	6	5	3	4
f_{13}	1	4	6	3	5	2
f_{14}	4	5	6	1.5	1.5	3
f_{15}	3	5	6	2	4	1
f_{16}	4	6	5	2	2	2
f_{17}	1	2	6	5	3.5	3.5
f_{18}	1	3	6	5	2	4
f_{19}	3	5	6	4	1	2
f_{20}	3	6	5	1	2	4
f_{21}	5	6	2	2	4	2
f_{22}	4	4	4	1	4	4
f_{23}	2.5	2.5	2.5	2.5	5.5	5.5
f_{24}	5	3	4	2	6	1
f_{25}	1	3	5	4	2	6
Total	**67.5**	**91.5**	**119.5**	**78.5**	**71.5**	**75.5**

5. Node localization in WSN

Wireless Sensor Network (WSN) consists of independent similar or diverse types of nodes that monitor the environment. These nodes sense, measure and then gather information from the environment. Based on decisions, sensed data is transmitted to users. Sensors employ single or multi-hop mechanisms to pass on information from one to other sensor hubs. WSNs comprise of a variety of sensors of either the same or differing sorts, interconnected by communication net. Nodes can be deployed in two manners, structured and unstructured manner. In a structured manner, the nodes are arranged in pre-defined method whereas in an unstructured manner the nodes are arranged randomly i.e., in ad-hoc method [51]. Different features of WSN including self-organization, fast deployment, and fault tolerance makes it efficient for a number of applications. Amongst all, some of the applications of WSN are military target tracking, natural disaster relief and bio-medical health monitoring [51]. Nodes that have no knowledge about their location are referred to as target nodes whose position needs to be estimated with the help of some of the deployed anchor nodes or beacons whose position is known. The position estimation of sensor nodes is one of the challenging problems and is known as the localization problem in WSN [52]. In a large portion of the applications, the center function of a WSN is to identify and report an event which can be definitively acclimatized and reacted to just if the precise area of the event is known. The problem that arises after deploying nodes is to find their actual location. The awareness of the location of nodes is the most important aspect in WSNs as it is largely needed to know from where the accurate information is coming. Global positioning system is one approach which is used to calculate the location information of nodes, however, it is not preferred because of its cost, size, and power consumption. WSN localization is achieved in two phases, ranging phase and position estimation phase. In ranging phase, the distance between target nodes and neighboring anchor nodes are determined which plays an important role in the localization process. In position estimation phase, the position of target nodes is estimated using the ranging information. The range between target nodes and anchor nodes can be determined using any of the localization techniques out of angle-of-arrival, received signal strength indicator, triangulation and maximum likelihood estimation [53]. There exist measurement errors after whatever ranging methods are used to determine the distance. The accuracy of position estimation phase is dependent on imprecise range measurements. The position estimation of nodes can be executed using two methods, namely geometric mean and optimization algorithm. Optimization algorithms like PSO, Genetic algorithm (GA) [54], Bees Optimization Algorithm (BOA) [55], Particle Swarm Optimization Algorithm (HPSO) and Biogeography Based Optimization (BBO) [56], Butterfly Optimization Algorithm [57] are used to find the closest area of the target node from the exact vicinity with some error where the error can be minimized using these algorithms. In WSN, energy is significant on the grounds that it is scare. Additionally, processing and memory capacities are constrained. In localization algorithm, high energy is required for calculation and communication. Node to node measure estimation is required for node localization and accuracy is normal for every application. Ecological conditions never stay indistinguishable, where sensor nodes are conveyed. A powerful robust localization algorithm must be acquainted with the node to node measure estimation and with least hardware in every single ecological condition, i.e., either isotropic or anisotropic. Versatile localization algorithms are required for extensive scale networks. While building up the localization algorithms, following parameters are required for the applications: exactness, energy effectiveness, scalability, trade-off among cost or above-recorded parameters. These difficulties make the WSN localization look into one of a kind and fascinating even after a time of serious research.

5.1. EGWO based node localization

The estimation of the position of unknown target nodes is considered as an optimization problem which may be solved using optimization algorithm. The objective function of position estimation is the sum square error between the target and anchor nodes. The main focus of node localization in WSN is to minimize the objective function and localize a maximum number of target nodes with the minimum localization error between target nodes and nodes whose position need to be estimated. A flowchart of range based distributed position estimation of target nodes using proposed EGWO algorithm is shown below in Fig. 9.

Considering there are M+N number of sensor nodes where M are the number of beacons and N, the number of target nodes whose coordinates (x, y) are to be estimated using proposed EGWO algorithm. The target nodes localization is achieved in following steps:

1. The target nodes and anchor nodes are randomly deployed within the sensor area. Each and every anchor node and target node have a transmission range R. Anchor nodes compute their area cognizance and transmit their coordinates.
2. The target nodes falling within the transmission range of three or more anchor nodes are considered as localizable nodes.
3. A target node measures distance from anchor nodes presents within its range using Eq. (11). The measured distance $\hat{d}_i$ is affected with Additive White Gaussian noise n_i. The noise measurement has a random value that is distributed randomly in the range $d_i = d_i \pm d_i(\frac{P_n}{100})$ where P_n is the noise percentage that affects the distance. Distance is measured between the anchor and target nodes using Eq. (12).
$$\hat{d}_i = d_i + n_i, \tag{11}$$
$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{12}$$
where (x, y) and (x_i, y_i) are the coordinates of target node and anchor node, respectively whereas d_i is actual distance.
4. The centroid of those anchor nodes is calculated which are present within the transmission range of target node using Eq. (13).
$$(x_c, y_c) = (\frac{1}{M}\sum_{i=1}^{M} x_i, \frac{1}{M}\sum_{i=1}^{M} y_i) \tag{13}$$
where M ≥ 3 is number of anchor nodes that are present in the transmission area of target node.
5. A group of K agents with polar coordinates is initialized keeping centroid as the origin.

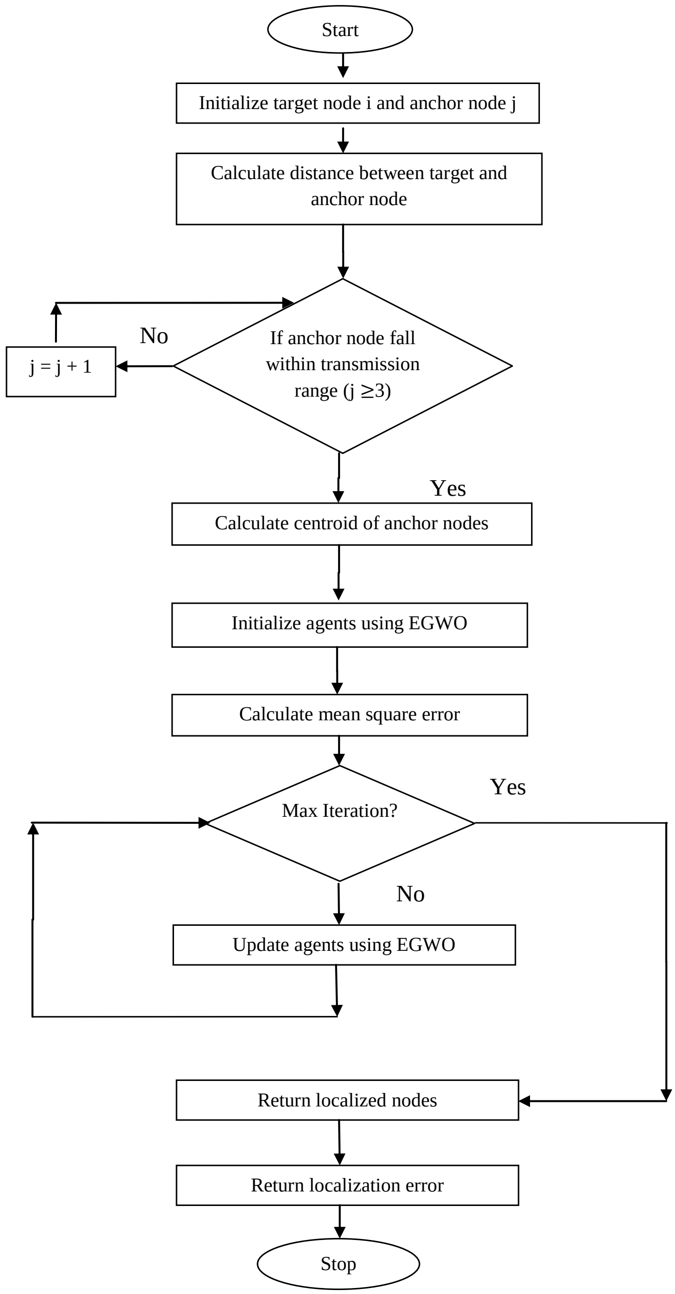

Figure 9. Flowchart for node localization using proposed EGWO algorithm

6. The objective function that is the mean square of error is evaluated for each agent using Eq. (14).

$$f(x,y) = \frac{1}{M}\sum_{i=1}^{M}\left(\sqrt{(x-x_i)^2+(y-y_i)^2}-\hat{d}_i\right)^2 \tag{14}$$

   where (x_i, y_i) are anchor node coordinates and (x, y) are the coordinates of node to be estimated.
7. Proposed EGWO updates the optimal position of the target nodes by reducing the error.
8. Once the location of target nodes is estimated, the total localization error is calculated using Eq. (15).

$$E_L = \frac{1}{N_L}\sum_{i=1}^{L}\left(\sqrt{(x_i-X_i)^2+(y_i-Y_i)^2}\right) \tag{15}$$

   Where (x_i, y_i) are coordinates of estimated node, (X_i, Y_i) are coordinates of target node and N_L is number of localized nodes.
9. Steps 2 to 8 are repeated till no more nodes can be localized. Lastly, a total number of unlocalized target nodes is calculated using Eq. (16).

$$N_{N_L} = N - N_L \tag{16}$$

   Once the target node gets localized, it is considered as an anchor node, thereby decreasing the flip ambiguity problem [54].

5.2. Results and discussion

The simulations of node localization in WSN are conducted using proposed EGWO, GWO, FA and PSO algorithms in QT Creator 2.4.0. For the simulations, a sensor area of 100×100 m is used and the number of target nodes and anchor nodes is set to 50 and 15 respectively. The transmission range for the anchor nodes is considered as 30 units and the number of iterations performed in carrying out simulations is 100. Parameters for different optimization algorithms based node localization are discussed below:

1. PSO based node localization: Each target node runs the PSO algorithm to become a localized node. Parameters that are considered for localization are:
   Population = 30
   Range of nodes = $(0, 100)$
   Number of iterations = 100
   Inertial weight ω = 0.7
   Cognitive and social scaling parameters $c_1 = c_2 = 1.494$

2. FA based node localization: Parameters taken into account for localization of target nodes are:
   Population = 30
   Range of nodes = $(0, 100)$
   Number of iterations = 100
   $\alpha = 0.25$, $\beta_0 = 1$ and $\gamma = 0.20$

3. GWO based node localization: Target nodes run GWO algorithm to localize themselves. Parameters for GWO based node localization are:
   Population $= 30$
   Range of nodes $= (0, 100)$
   Number of iterations $= 100$
   $\vec{a} = 2$ to 0

4. EGWO based node localization:
   Population $= 30$
   Range of nodes $= (0, 100)$
   Number of iterations $= 100$
   $\vec{a} = [0, 1]$

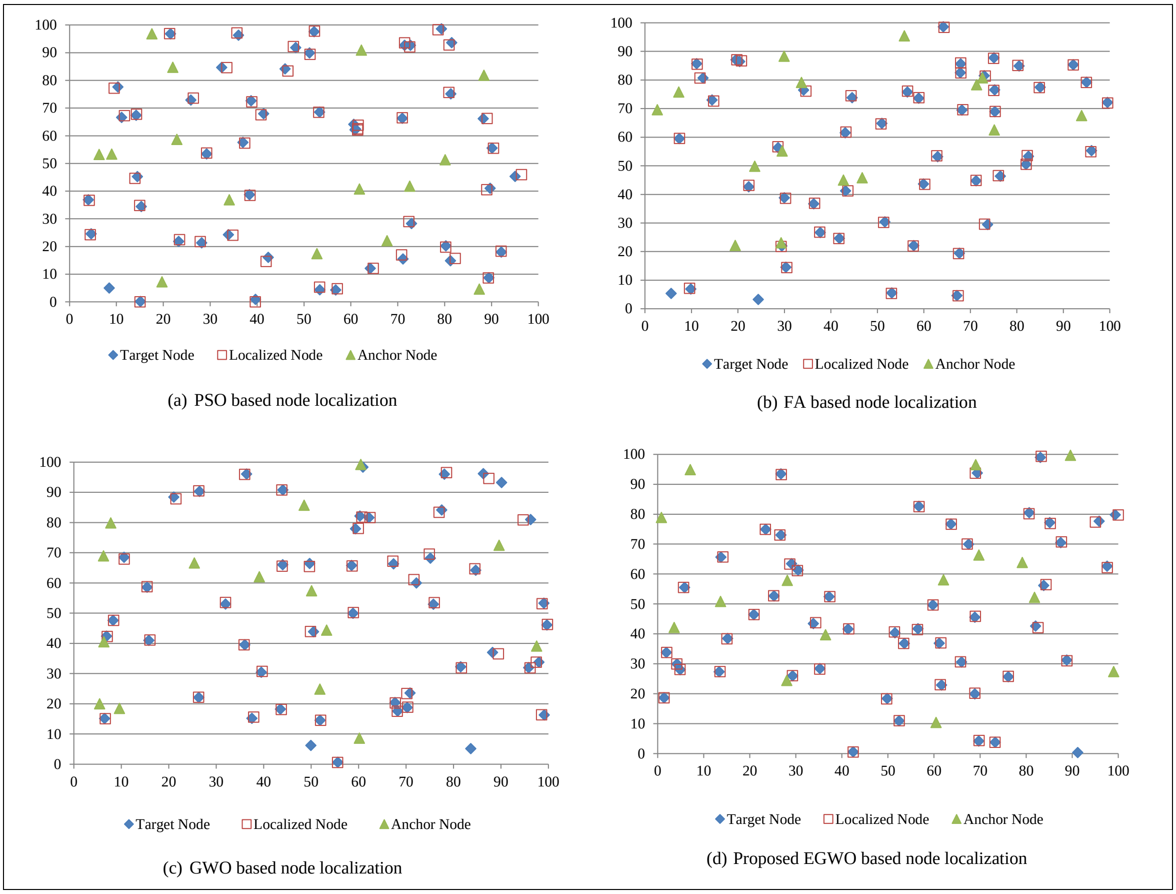

Figure 10. Results of PSO, FA, GWO and proposed EGWO based node localization for $N = 50, M = 15, R = 30$ and $P_n = 2$ in sensor area of 100×100 m

Meta-heuristic algorithms are stochastic in nature due to which same results are not produced in all the performed trials. Even when the initial deployment is identical, same solutions are not produced. By virtue of this, the results achieved during trial runs are averaged. Besides having a random deployment in each iteration, the number of nodes that are localized is different, leading to change in computation time [57]. The results of node localization, i.e., the target nodes, anchor nodes and the position of the localized node with different meta-heuristic algorithms are shown in Fig. 10. The number of anchor nodes and target nodes is kept constant for all the algorithms viz. PSO, FA, GWO and proposed EGWO algorithm. Results of range based node localization using different meta-heuristic optimization algorithms in 3 trials are summarized in Table 5.

Table 5. Simulation results of 3 trials runs for PSO, FA, GWO and proposed EGWO based node localization with $N = 50, M = 15, R = 30$, sensor area of 100×100 m

Target Nodes	Anchor Nodes	Trial	PSO			FA			GWO			EGWO (Proposed)		
			N_L	E_L	T_L	N_L	E_L	T_L	N_L	E_L	T_L	N_L	E_L	T_L
25	10	1	22	0.80715	0.046	19	0.33555	.144	19	0.65325	0.060	20	**0.25085**	0.062
		2	17	0.72821	0.040	20	0.26462	.144	23	0.42589	0.080	23	**0.23183**	0.065
		3	18	0.79765	0.040	21	0.29639	.131	24	0.47714	0.070	22	**0.27355**	0.610
50	15	1	48	0.57879	0.140	50	0.50555	2.500	46	0.42175	0.190	49	**0.18320**	0.196
		2	50	0.75325	0.150	49	0.32698	4.220	44	0.51923	0.170	48	**0.25568**	0.218
		3	47	0.58700	0.150	49	0.25482	1.630	48	0.52500	0.230	50	**0.25430**	0.200
75	20	1	75	0.67414	0.310	74	0.70396	2.970	75	0.47171	0.370	71	**0.19901**	0.343
		2	75	0.72012	0.310	75	0.29186	2.730	75	0.44849	0.380	75	**0.20985**	0.374
		3	73	0.77132	0.300	72	0.27912	5.880	75	0.42836	0.370	75	**0.19798**	0.374
100	25	1	100	0.66822	0.490	100	0.77971	5.660	100	0.36230	0.592	100	**0.20326**	0.592
		2	100	0.61484	0.500	100	0.29919	6.330	100	0.55555	0.590	100	**0.19186**	0.592
		3	100	0.60815	0.500	100	0.38575	3.550	100	0.51878	0.590	100	**0.19924**	0.592

N_L = Number of localized node, E_L = localization error and T_L = computing time (in ms)

Table 6. Summary of 10 Monte Carlo runs for PSO, FA, GWO and proposed EGWO based node localization with $N = 50, M = 15, R = 30$, sensor area of 100×100 m

Algorithm	P_n=5			P_n=2		
	N_{N_L}	E_L	T_L	N_{N_L}	E_L	T_L
PSO	4.8	0.887736	0.133	4.2	0.740206	0.139
FA	5.6	2.632273	1.536	5.6	0.785930	2.725
GWO	6.0	0.945259	0.171	4.8	0.871128	0.171
EGWO	5.1	**0.571330**	0.179	3.4	**0.246451**	0.174

P_n = Additive noise, N_{N_L} = Number of un-localized node, E_L = Localization error and T_L = Computing time

The results obtained from these metaheuristic algorithms in solving the localization problem with different noise conditions are presented in Table 6. The effect of noise added to the measured distance is clearly noticed from the results. If the noise is more, the computed localization error is more and vice-versa, which indicates that a localization error is dependent on noise as shown in Fig. 11. If the noise is more, the accuracy of localizing the target nodes reduces with the decrease in a number of localized nodes. From Fig. 11, it is noted that the localization error for localizing the unknown target nodes is less using proposed EGWO algorithm than other standard stochastic optimization algorithm.

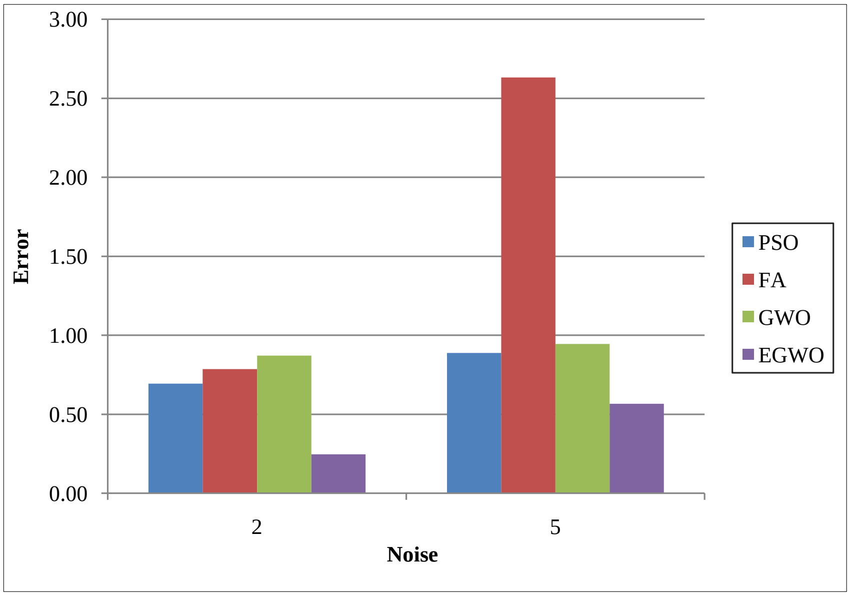

Figure 11. Error versus Noise

From the results in Tables 5 and 6, it is clear that location estimation of un-localized nodes, i.e., target nodes from proposed algorithm is more accurate as compared to nodes localized by PSO, FA and GWO as the proposed EGWO algorithm out searches the search space and then exploit the solution in order to find the position of localized nodes with the minimum localization error. The proposed EGWO algorithm results in more number of localized nodes with large accuracy and minimum localization error, but computing time is more other than FA. The exploration-exploitation trade-off helps to avoid getting trapped in local optima and provide better results making it a better optimization algorithm. Additive white gaussian noise is an important parameter for providing better accuracy in the localization of nodes. Other parameters on which the performance of localization depends are anchor node density and transmission range.

For estimating the performance of different standard optimization algorithms viz. PSO, FA, GWO and proposed EGWO on parameters, namely anchor node density and transmission range the target nodes are fixed to 50 with varying the anchor nodes and transmission range respectively. The number of anchor nodes for performance evaluation of a parameter transmission range is kept 15. Post that, the

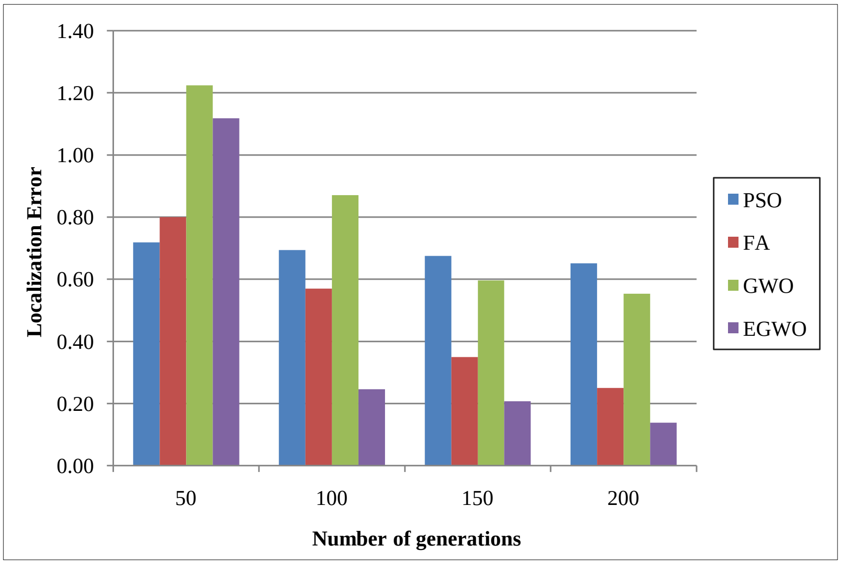

Figure 12. Flowchart for node localization using proposed EGWO algorithm

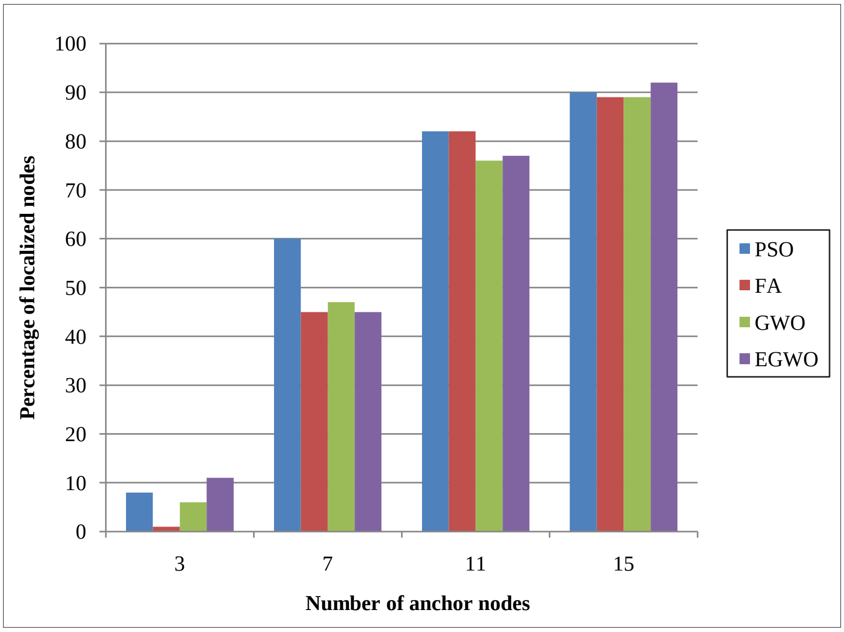

Figure 13. Percentage of localized nodes versus number of anchor nodes

transmission range for determining the performance of applied optimization algorithms on parameter anchor node density is fixed to 30 units. The number of iterations for carrying out the performance evaluation for node localization is taken 100. It is depicted in Fig. 12 that increase in the number of iterations decreases the localization error. The localization error determined from the proposed EGWO algorithm is minimal as compared to error determined from other algorithms. Proposed EGWO algorithm extensively searches the search space and later finds the best solution by not getting trapped in premature convergence. Due to this mechanism, the proposed algorithm finds the location of a un-localized node in less number of iterations.

It is observed that keeping the number of anchor nodes limited does not promote the localization of target nodes. With the increase in the anchor node density, the location estimation of un-localized nodes is preeminent and an immense count of target nodes gets localized as shown in Fig. 13. An insufficient number of anchor nodes around the target node $(M \geq 3)$ edge to failure in localization of maximum number of nodes.

For better localization of the target nodes, transmission range should be more. The transmission range is directly proportional to the number of target nodes localized. If the transmission range is increased, greater number of nodes can be localized as shown in Fig. 14.

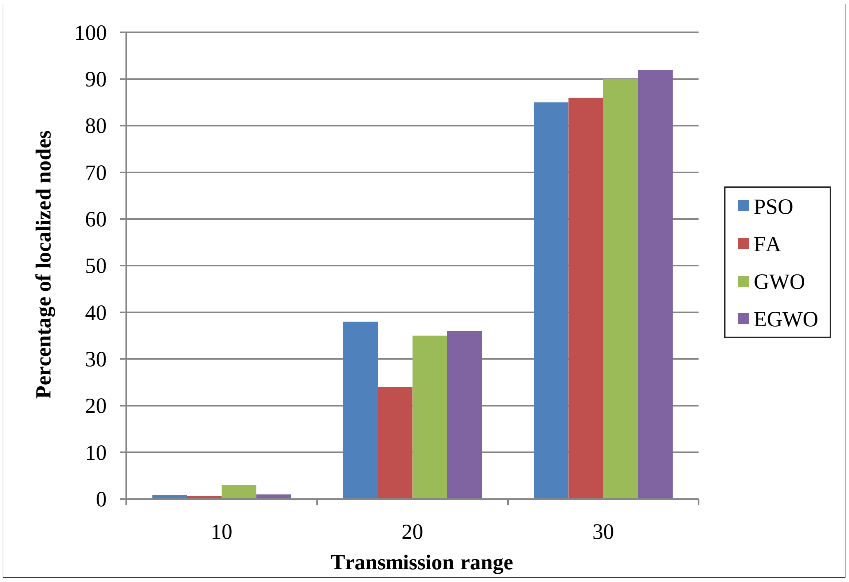

Figure 14. Percentage of localized nodes versus transmission range

6. Conclusions and future work

This paper introduced an enhancement to the Grey Wolf Optimizer which is inspired by the hunting behavior of grey wolves in nature. The basic three steps engaged in the hunting mechanism of existing GWO, i.e., tracking, pursuing, and attacking the prey are modified in the proposed algorithm. The proposed Enhanced GWO (EGWO) algorithm attempts to balance the ability of exploration and exploitation, and thus increases its stability and convergence rate thereby making it efficient for solving global optimization problems. The simulation results for the benchmark test problems revealed that the performance of the proposed algorithm is promising in terms of better exploration and exploitation of the search space in comparison with the other studied algorithms. Moreover, the statistical tests proved the competence and superiority of EGWO algorithm to existing metaheuristic algorithms. The paper also considered the range based node localization problem in WSN in which the node localization is performed using the proposed EGWO algorithm, which is a challenging and NP hard problem. The simulation results show that the proposed method is found to be very effective for node localization in terms of number of nodes localized, maximum localization accuracy, and minimum mean square localization error. The results proved EGWO algorithm has an ability to become an effective tool for solving real word optimization problems. As future work, we intend to perform comparisons with other meta-heuristic algorithms like CARBBO, RCBBO, HPA and solve other problems. The node localization performed in the 2D scenario can be extended to the 3D scenario.

References

[1] Kulkarni RV, Forster A, Venayagamoorthy GK. Computational intelligence in wireless sensor networks: A survey. *IEEE communications surveys & tutorials*, 2011;13(1):68–96. doi:10.1109/SURV.2011.040310.00002.

[2] Yang X. Introduction to mathematical optimization. *Cambridge International Science Publishing*, 2008.

[3] Talbi EG. Metaheuristics: from design to implementation, volume 74. John Wiley & Sons, 2009. ISBN:978-0-470-27858-1.

[4] Yan X, Zhang C, Luo W, Li W, Chen W, Liu H. Solve traveling salesman problem using particle swarm optimization algorithm. *International Journal of Computer Science*, 2012;9:264–271. ISSN: 2348-4845.

[5] Zomaya AY, Teh YH. Observations on using genetic algorithms for dynamic load-balancing. *IEEE transactions on parallel and distributed systems*, 2001;12(9):899–911. doi:10.1109/71.954620.

[6] Yang XS. Nature-inspired metaheuristic algorithms. Luniver press, 2010. ISBN:1905986289.

[7] Shi Y, et al. Particle swarm optimization: developments, applications and resources. In: evolutionary computation, 2001. Proceedings of the 2001 Congress on, volume 1. IEEE, 2001 pp. 81–86. doi:10.1109/CEC.2001.934374.

[8] Zhang Y, Wang S, Ji G. A comprehensive survey on particle swarm optimization algorithm and its applications. *Mathematical Problems in Engineering*, 2015. URL `http://dx.doi.org/10.1155/2015/931256`.

[9] Yang XS. Firefly algorithm, Levy flights and global optimization. In: Research and development in intelligent systems XXVI, pp. 209–218. Springer, 2010. doi:10.1007/978-1-84882-983-1_15.

[10] Arora S, Singh S. Performance Research on Firefly Optimization Algorithm with Mutation. In: International Conference, Computing & Systems. 2014.

[11] Arora S, Singh S. A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search. In: Control Computing Communication & Materials (ICCCCM), 2013 International Conference on. IEEE, 2013 pp. 1–4. doi:10.1109/ICCCCM.2013.6648902.

[12] Kalra S, Arora S. Firefly Algorithm Hybridized with Flower Pollination Algorithm for Multimodal Functions. In: Proceedings of the International Congress on Information and Communication Technology. Springer, 2016 pp. 207–219. doi:10.1007/978-981-10-0767-5_23.

[13] Yang XS. Flower pollination algorithm for global optimization. In: International Conference on Unconventional Computing and Natural Computation. Springer, 2012 pp. 240–249. doi:10.1007/978-3-642-32894-7_27.

[14] Yang XS, Deb S. Cuckoo search via Levy flights. In: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. IEEE, 2009 pp. 210–214. doi:10.1109/NABIC.2009.5393690.

[15] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE computational intelligence magazine*, 2006;1(4):28–39. doi:10.1109/MCI.2006.329691.

[16] Arora S, Singh S. Butterfly algorithm with Levy Flights for global optimization. In: Signal Processing, Computing and Control (ISPCC), 2015 International Conference on. IEEE, 2015 pp. 220–224. doi:10.1109/ISPCC.2015.7375029.

[17] Arora S, Singh S. An improved butterfly optimization algorithm with chaos. *Journal of Intelligent & Fuzzy Systems*, 2017;32(1):1079–1088. doi:10.3233/JIFS-16798.

[18] Dasgupta D, Michalewicz Z. Evolutionary algorithms in engineering applications. Springer Science & Business Media, 2013.

[19] Erol OK, Eksin I. A new optimization method: big bang–big crunch. *Advances in Engineering Software*, 2006;37(2):106–111. doi:10.1016/j.advengsoft.2005.04.005.

[20] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. *Information sciences*, 2009;179(13):2232–2248. doi:10.1016/j.ins.2009.03.004.

[21] Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 2003;11(1):1–18. doi:10.1162/106365603321828970.

[22] Brest J, Zamuda A, Fister I, Boskovic B. Some improvements of the self-adaptive jde algorithm. In: Differential Evolution (SDE), 2014 IEEE Symposium on. IEEE, 2014 pp. 1–8. doi:10.1109/SDE.2014.7031537.

[23] Davis L. Genetic algorithms and simulated annealing. 1987.

[24] Xi B, Liu Z, Raghavachari M, Xia CH, Zhang L. A smart hill-climbing algorithm for application server configuration. In: Proceedings of the 13th international conference on World Wide Web. ACM, 2004 pp. 287–296. doi:10.1145/988672.988711.

[25] Zhang Y, Agarwal P, Bhatnagar V, Balochian S, Yan J. Swarm intelligence and its applications. *The Scientific World Journal*, 2013. URL `http://dx.doi.org/10.1155/2013/528069`.

[26] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 2014;69:46–61. URL `https://doi.org/10.1016/j.advengsoft.2013.12.007`.

[27] Sharma Y, Saikia LC. Automatic generation control of a multi-area ST–Thermal power system using Grey Wolf Optimizer algorithm based classical controllers. *International Journal of Electrical Power & Energy Systems*, 2015;73:853–862. doi:10.1016/j.ijepes.2015.06.005.

[28] El-Fergany AA, Hasanien HM. Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms. *Electric Power Components and Systems*, 2015;43(13):1548–1559. URL `http://dx.doi.org/10.1080/15325008.2015.1041625`.

[29] Noshadi A, Shi J, Lee WS, Shi P, Kalam A. Optimal PID-type fuzzy logic controller for a multi-input multi-output active magnetic bearing system. *Neural Computing and Applications*, 2016;27(7):2031–2046. doi:10.1007/s00521-015-1996-7.

[30] Sulaiman MH, Mustaffa Z, Mohamed MR, Aliman O. Using the gray wolf optimizer for solving optimal reactive power dispatch problem. *Applied Soft Computing*, 2015;32:286–292. doi:10.1016/j.asoc.2015.03.041.

[31] Medjahed S, Saadi TA, Benyettou A, Ouali M. Gray Wolf Optimizer for hyperspectral band selection. *Applied Soft Computing*, 2016;40:178–186. doi:10.1016/j.asoc.2015.09.045.

[32] Zhang Y, Wu L, Wang S, Huo Y. Chaotic artificial bee colony used for cluster analysis. In: Intelligent Computing and Information Science, pp. 205–211. Springer, 2011. doi:10.1007/978-3-642-18129-0_33.

[33] Zhang Y, Wu L. A hybrid TS-PSO optimization algorithm. *Journal of Convergence Information Technology*, 2011;6(5):169–174. doi:10.4156/jcit.vol6.issue5.18.

[34] Cui Z, Cai X. Integral particle swarm optimization with dispersed accelerator information. *Fundamenta Informaticae*, 2009;95(4):427–447. doi:10.3233/FI-2009-158.

[35] Zhang Y, Wu X, Lu S, Wang H, Phillips P, Wang S. Smart detection on abnormal breasts in digital mammography based on contrast-limited adaptive histogram equalization and chaotic adaptive real-coded biogeography-based optimization. *Simulation*, 2016;92(9):873–885. doi:10.1177/0037549716667834.

[36] Wang S, Li P, Chen P, Phillips P, Liu G, Du S, Zhang Y. Pathological Brain Detection via Wavelet Packet Tsallis Entropy and Real-Coded Biogeography-based Optimization. *Fundamenta Informaticae*, 2017;151(1-4):275–291. doi: 10.3233/FI-2017-1492.

[37] Wang S, Zhang Y, Dong Z, Du S, Ji G, Yan J, Yang J, Wang Q, Feng C, Phillips P. Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection. *International Journal of Imaging Systems and Technology*, 2015;25(2):153–164. doi:10.1002/ima.22132.

[38] Lu C, Xiao S, Li X, Gao L. An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production. *Advances in Engineering Software*, 2016;99:161–176. doi:10.1016/j.advengsoft.2016.06.004.

[39] Lu C, Gao L, Li X, Xiao S. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Engineering Applications of Artificial Intelligence*, 2017;57:61–79. URL `https://doi.org/10.1016/j.engappai.2016.10.013`.

[40] Precup RE, David RC, Petriu EM. Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity. *IEEE Transactions on Industrial Electronics*, 2017;64(1):527–534. doi:10.1109/TIE.2016.2607698.

[41] Jayakumar N, Subramanian S, Ganesan S, Elanchezhian E. Grey wolf optimization for combined heat and power dispatch with cogeneration systems. *International Journal of Electrical Power & Energy Systems*, 2016;74:252–264. URL `https://doi.org/10.1016/j.ijepes.2015.07.031`.

[42] Mittal N, Singh U, Sohi BS. Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, 2016; pp. 1–16. URL `http://dx.doi.org/10.1155/2016/7950348`.

[43] Muangkote N, Sunat K, Chiewchanwattana S. An improved grey wolf optimizer for training q-Gaussian Radial Basis Functional-link nets. In: Computer Science and Engineering Conference (ICSEC), 2014 International. IEEE, 2014 pp. 209–214. doi:10.1109/ICSEC.2014.6978196.

[44] Kohli M, Arora S. Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*, 2017. URL `https://doi.org/10.1016/j.jcde.2017.02.005`.

[45] Arora S, Singh S. An Improved Butterfly Optimization Algorithm for Global Optimization. *Advanced Science, Engineering and Medicine*, 2016;8(9):711–717. URL `https://doi.org/10.1166/asem.2016.1904`.

[46] Arora S, Singh S. An Effective Hybrid Butterfly Optimization Algorithm with Artificial Bee Colony for Numerical Optimization. *International Journal of Interactive Multimedia and Artificial Intelligence*, 2017;4(4):14–21. doi:10.9781/ijimai.2017.442.

[47] Gopakumar A, Jacob L. Performance of some metaheuristic algorithms for localization in wireless sensor networks. *International Journal of Network Management*, 2009;19(5):355–373. doi:10.1002/nem.714.

[48] Jadon SS, Bansal JC, Tiwari R. Escalated convergent artificial bee colony. *Journal of Experimental & Theoretical Artificial Intelligence*, 2016;28(1-2):181–200. URL `http://dx.doi.org/10.1080/0952813X.2015.1020523`.

[49] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 2009;214(1):108–132. doi:10.1016/j.amc.2009.03.090.

[50] James J, Li VO. A social spider algorithm for global optimization. *Applied Soft Computing*, 2015;30:614–627. doi:10.1016/j.asoc.2015.02.014.

[51] Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. *Computer networks*, 2008;52(12):2292–2330. doi:10.1016/j.comnet.2008.04.002.

[52] Alrajeh NA, Bashir M, Shams B. Localization techniques in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2013.

[53] Singh P, Tripathi B, Singh NP. Node localization in wireless sensor networks. *International journal of computer science and information technologies*, 2011;2(6):2568–2572. doi:10.1109/IMTC.2011.5944076.

[54] Sun W, Su X. Wireless sensor network node localization based on genetic algorithm. In: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on. IEEE, 2011 pp. 316–319.

[55] Moussa A, El-Sheimy N. Localization of wireless sensor network using bees optimization algorithm. In: Signal Processing and Information Technology (ISSPIT), 2010 IEEE International Symposium on. IEEE, 2010 pp. 478–481. doi:10.1109/ISSPIT.2010.5711760.

[56] Kumar A, Khosla A, Saini JS, Singh S. Meta-heuristic range based node localization algorithm for wireless sensor networks. In: Localization and GNSS (ICL-GNSS), 2012 International Conference on. IEEE, 2012 pp. 1–7. doi:10.1109/ICL-GNSS.2012.6253135.

[57] Arora S, Singh S. Node Localization in Wireless Sensor Networks Using Butterfly Optimization Algorithm. *Arabian Journal for Science and Engineering*. 2017, pp. 1–11. doi:10.1007/s13369-017-2471-9.