Towards realistic mimicking of grey wolves hunting process for bounded single objective optimization

Ashish Mani
Amity University Uttar Pradesh
Department of electrical and electronics engineering
Noida, India.
amani@amity.edu

Anjali Jain
Amity University Uttar Pradesh
Department of electrical and electronics engineering
Noida, India.
ajain5@amity.edu

Abstract— **Canis lupus (grey) wolves hunt for their prey in a pack. There exist evolutionary algorithms based on the hunting pattern of Canis lupus wolves known as Grey wolf Optimizer (GWO). It is a powerful optimizer and had produced competitive results for many difficult problems. There already exist several variants of GWO because of its simplicity and exploitative qualities. GWO has been proved to be a very good exploiter. However, the modeling of grey wolves concentrates more on exploitation rather than exploration and thus results in the problem of local stagnation. To enhance the exploration capabilities, dynamic behavior is added to prey. It was assumed that the prey was static in the original GWO wherein the prey always tries to run away from the predator. The proposed algorithm enhances diversity by varying the position of prey with the help of Levy distribution rather than considering it to be static. This paper aims at introducing a novel, realistic version of GWO wherein the new population is generated with the help of Levy flight distribution of prey as well as with the different class of wolves by a suitable modification in the hierarchy of wolves. Also, the three primary strides of chasing, looking, circling, and assaulting of prey, are executed. However, the new population which is created has a better exploration because of levy flight distribution of prey position. The algorithm is tested on a well-known suite of 23 benchmark problems. The outcomes show that the proposed algorithm, GWOLF can give either better or competitive results as contrasted with the original GWO and PSO meta-heuristics.**

Keywords—Grey wolf optimization, GWO variant, levy flight, benchmark problem, evolutionary algorithm, Canis lupus wolves

I. Introduction

Metaheuristic algorithm provides a variety of solution to solve many real-world engineering problems. The basis for most of the algorithms is the 'survival of the fittest' theory of evolutionary algorithms (EA). The different domain in EA includes theories based on swarm's intelligence, natural behavior of biologically inspired processes, as well as logical behavior of many physical algorithms which exist in nature [1].

There exist distinctive deterministic algorithms to find optimal solutions to the problems however the main concern with the available algorithms is that they are not producing the best results for all types of optimization problems. The response to this inquiry is that most problems do not generally have explicit properties of functions, for example, the function which is to be optimized may not be differentiable, continuous, or convex. Subsequently, most known deterministic techniques do not give reasonable solutions in such cases. Henceforth, numerous nature-inspired metaheuristic algorithms are useful in taking care of such problems which are non-differentiable, non-continuous, and non-convex. These procedures are guided by the natural phenomenon. Some of the acclaimed metaheuristics are Genetic Algorithm (GA) [2], Particle Swarm Optimization (PSO) [3], Biogeography-based advancement (BBO) [4], Ant state Optimization (ACO) [5], Di□erential Evolution (DE) [6], Simulated Annealing (SA) [7] and so on.

Although there are different Nature-Inspired Optimizers with stochastic behavior available in the literature, yet Grey Wolf Optimization (GWO) algorithm cannot be ignored because of its exploitative conduct and simplicity when contrasted with other meta-heuristic approaches [8]. As the GWO metaheuristic depends on the leadership hierarchy system of grey wolves, hence the alpha, beta, and delta wolves help in investigating the search regions more e□ciently because the search equations of GWO depend on the position of these driving wolves. In GWO rather than the direction of a solitary search agent, three search agents (wolves) go about as guiding agents (wolves) for the pack to look through prey (optima), that is, in GWO, the inquiry procedure is guided by the three driving wolves, which are known as pioneers for the pack and aides in investigation yet sometimes, as other meta-heuristics, GWO method faces the issue of stagnation of wolf pack in neighborhood arrangements.

To adjust the exploration and exploitation, researchers have proposed different variations of GWO, for example, exponential change instead of linear change in the estimation of parameter 'a' [9], alteration in the estimation of 'C' parameter [10], alteration in the estimation of alpha wolves with a mutation operator [11]. Chaotic and random search is also added by the researchers to enhance the exploratory capabilities of the original GWO [12-14]. GWO method's variations have been utilized to solve large scale numerical optimization benchmark problems defined in CEC 2014, CEC2017 competitions [15]. Levy distribution is utilized in one of the variations of the GWO to solve CEC 2014 and CEC 2011[16].

The common issue with most of the meta-heuristics is the problem of premature convergence and stagnation in local optima. GWO also faces the same issue as reported in earlier literature. Exploration and exploitation balance plays a vital role in defining the performance of population-based metaheuristics. The performance of metaheuristics is usually poor if exploitation is good and exploration is poor and vice versa. To achieve good performance on a class of optimization problems, exploitation, and exploration capabilities of a metaheuristic should be balanced. The position-update equation of GWO is utilized to create new candidate individuals based on the information of the past global best individual (α wolf), the second-best individual (β wolf), and the third-best individual (δ wolf) is good at exploitation, however poor at exploration. Thus, in this paper, we increased one more level of the hierarchy of grey wolves to additionally enhance the exploitation and prey is considered to have a dynamic behavior and is modeled with the help of levy distribution wherein alpha wolves are attempting to follow it and beta wolves are following alpha and the hierarchy goes on for delta and omega wolves. This adjustment is more realistic and closer to the actual behavior of Canis lupus wolves which can bring more information and help in producing promising candidate individuals to enhance the exploration of the original GWO.

Therefore, in the proposed work, the novel realistic model of the behavior of Canis lupus wolves is proposed to improve exploitation as well as exploration. This paper is focused on improvisation of the theory given in the earlier version of GWO in adopting and copying the leadership hierarchy and chasing mechanism of grey wolves in nature more realistically. The proposed metaheuristic is simple, powerful, and accurate. It outperforms or provides competitive results for unimodal functions as well as for multimodal functions on both low-dimensional and high-dimensional optimization problems. The main contributions of this paper are as follows:

- A novel, realistic framework modification of GWO as compared to previous version of GWO and PSO is proposed.
- Prey is modeled as moving whereas it is considered to be static in original GWO.
- Exploitation is improved by adding one more level in the hierarchy of grey wolves.
- Exploration is enhanced with the help of dynamic prey position modeled using levy distribution.

The remainder of this paper is composed as follows- Section II describes the overview of GWO and the techniques to improve GWO. Section III explains about proposed metaheuristic. The proposed GWOLF was tested using a set of well-known test functions, and the results are discussed in Section IV. Finally, the paper's conclusion is given in Section V.

II. Grey Wolf Optimizer (GWO)

A. Overview

Grey Wolf optimization algorithm is inspired by the behavior of Canis lupus wolves. This algorithm is modeled as per their social hierarchy and hunting behavior [8]. In GWO algorithm, the solution which is best amongst all the solution is named as alpha (α). Alpha wolves are said to be dominant wolves and hence are considered to be the best in managing the pack, the second and third best solutions are called beta (β) and delta (δ), respectively. Beta wolves are subordinate to alpha wolves and are good discipliner of the pack. Delta wolves submit to alpha and beta wolves and other individuals in the population are assumed as omega (ω). Omega wolves are lowest in ranking and play the role of scapegoat. Social hierarchy of wolves is shown in Fig 1.

For the modeling of encircling behavior of the wolves mathematically, the position of wolves in $(t+1)^{th}$ iteration is given as follows [8]:

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.|\vec{C}.\left(\overrightarrow{X_p}(t) - \vec{X}(t)\right)| \qquad (1)$$

where $\vec{X}(t)$ is defined as the position vector of a grey wolf in t^{th} iteration and t is the index for the current iteration,

$\overrightarrow{X_p}(t)$ is defined as the position vector of the prey, the coefficient vectors are $\vec{A}$ and $\vec{C}$ which are defined as $\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a}$, $\vec{C} = 2.\overrightarrow{r_2}$

where random vectors $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ lie in [0, 1], and as per (2), $\vec{a}$ is varying from 2 to 0 and is dependent on the iterations, i.e.,

$$a(t) = 2 - \frac{2t}{MaxIter} \qquad (2)$$

where $MaxIter$ indicates the total number of iterations. The positions of the other wolves are updated according to the positions of α, β, and δ as follows [8]:

$$\vec{X}_1 = \overrightarrow{X_\alpha}(t) - \overrightarrow{A_1}.\overrightarrow{C_1}.\left(\overrightarrow{X_\alpha}(t) - \vec{X}(t)\right) \qquad (3)$$

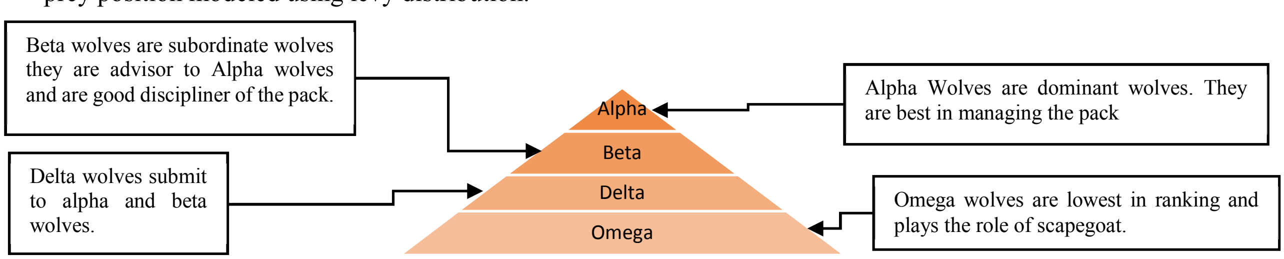

Fig. 1. Leadership hierarchy of the wolves in GWO

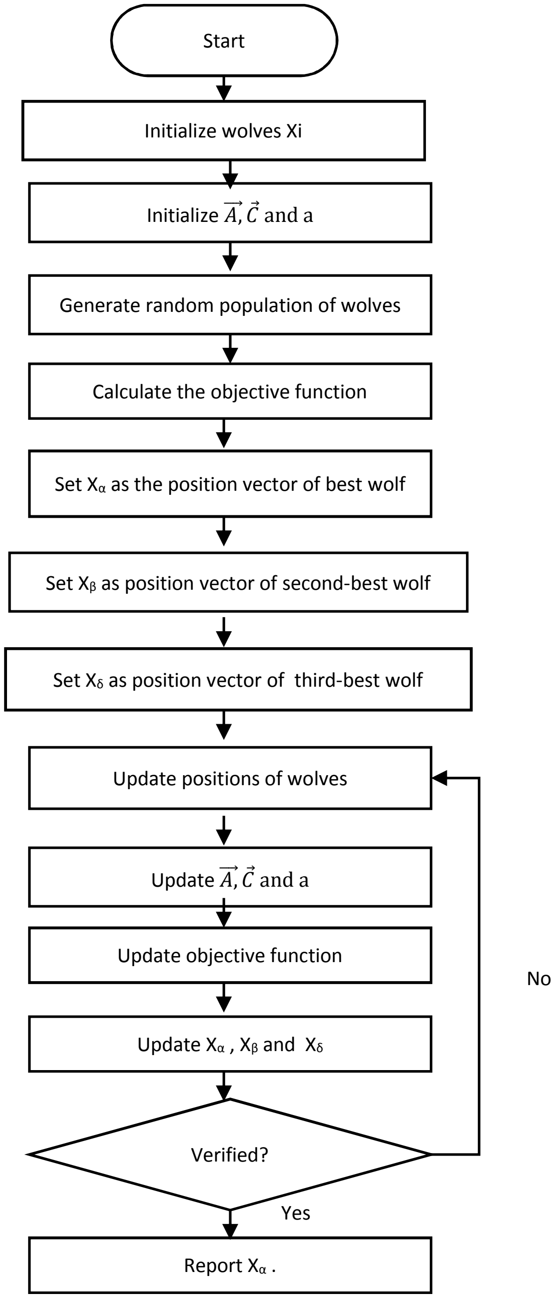

Fig. 2. Flow chart of original GWO technique [8].

Initialize the grey wolf populations Xi (i=1,2,.........,n)
Initialize $\vec{A}, \vec{C}$ and a
Calculate the fitness of each search agent
X_α= the position vector of best search agent
X_β= the position vector second best search agent
X_δ= the position vector of third best search agent
***While** (t<Maximum number of iterations)*
    ***for** each search agent*
        update the position of the search agent by equation (6)
    end for
    Update $\vec{A}, \vec{C}$ and a
    Calculate the fitness of all search agents
    Update X_α, X_β and X_δ
    t=t+1
End while
Return X_α

Fig. 3. Pseudo code for original GWO [8].

$$\vec{X}_2 = \overrightarrow{X_\beta}(t) - \overrightarrow{A_2}.\overrightarrow{C_2}.\left(\overrightarrow{X_\beta}(t) - \vec{X}(t)\right) \quad (4)$$

$$\vec{X}_3 = \overrightarrow{X_\delta}(t) - \overrightarrow{A_3}.\overrightarrow{C_3}.\left(\overrightarrow{X_\delta}(t) - \vec{X}(t)\right) \quad (5)$$

Where the position of alpha wolves, beta wolves and delta wolves are $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$ respectively. The new population for the next generation will be generated as under,

$$\vec{X}(t+1) = \frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3} \quad (6)$$

Exploration and exploitation behaviors are defined by the values of $\vec{A}$ and $\vec{C}$. The vector $\vec{A}$ is a random value inside [-2a, 2a], where the elements are linearly decreasing from 2 to 0. The exploration happens when $|\vec{A}|>1$ and $|\vec{C}|>1$. The exploitation tendency is improved when $|\vec{A}|<1$ and $|\vec{C}|<1$

Fig 2,3 shows the flowchart of the GWO algorithm and pseudo code respectively.

B. Techniques to Balance Exploitation and Exploration

1) Addition of kappa wolves to improve exploitation: Though in the proposed algorithm, delta wolves are classified in two sub-levels, one sub-level of wolves is called modified delta (δm),, which helps alpha and beta while chasing the prey wheren the other sub-level is kappa (κ) which is responsible for taking care for the weak wolves. Kappa wolves are also added to the original algorithm as shown in (8) so that exploitation is further enhanced,

$$\vec{X}_4 = \overrightarrow{X_\kappa}(t) - \overrightarrow{A_4}.\overrightarrow{C_4}.\left(\overrightarrow{X_\kappa}(t) - \vec{X}(t)\right) \quad (7)$$

The new position of rest wolves will be given as the average of all the positions as in the original GWO in (6).

$$\vec{X}(t+1) = \frac{\vec{X}_1+\vec{X}_2+\vec{X}_3+\vec{X}_4}{4} \quad (8)$$

The simulation results after adding kappa wolves are not shown individually because of space constraints.

2) Levy flight operator to improve exploration: As per Vishwanathan et.al., Levy walk or Levy flight adds improvement in exploring random objects. Hence, the behavior of hunting wolves may be revised as per Levy flight concept to add exploration to their behavior [18] and therefore, Levy flight operator has been explicitly discussed in the current section.

Levy Flight is a class of scale-free walks with randomly oriented steps according to Levy distribution [16]. Animal motions have random walks which can be represented by Levy distribution [17]. A power law equation is used to represent the levy distribution as in (9)

$$L(s)\ |s|^{-1-\theta}, 0 < \theta < 2, \quad (9)$$

Here, θ is termed as index for controlling the stability in Levy distribution and s is the variable [16].

$$L(s,\gamma,\mu) = \begin{cases} \frac{\gamma}{2\pi} exp\left[\frac{-\gamma}{2(s-\mu)}\right]\frac{1}{(s-\mu)^{\frac{3}{2}}} & 0 < \mu < s < \infty \\ 0 & s \leq 0 \end{cases} \quad (10)$$

Where γ is called a scale parameter and is always greater than 0 and the shift parameter is defined by μ

Mantegna technique [18] is an accurate strategy to provide stochastic variables who's probability densities are converging to Levy steady distribution that is controlled by parameter ζ (0.3< ζ <1.99). Therefore, the Mantegna strategy is utilized here to obtain LF throughout the searching process. In this regard, in (11), the step size can be formulated as [16]:

$$rand\big(size(\mathrm{D})\big) \oplus Levy(b) \sim 0.01 \frac{\mu}{v^{-\theta}} \left(\vec{X}(t) - \overrightarrow{X_\alpha}(t)\right) \quad (11)$$

where u and v values should be attained based on normal distributions;

$$\mu \sim N(0, \sigma_\mu^2), v \sim N(0, \sigma_v^2), \quad (12)$$

With

$$\sigma_\mu = \left[\frac{\Gamma(1+\theta) sin\left(\frac{\pi\theta}{2}\right)}{\Gamma\left(\frac{1+\theta}{2}\right)\theta \times 2^{\frac{\theta-1}{2}}}\right], \sigma_v = 1, \quad (13)$$

Where conventional gamma function is defined by Γ.

III. Proposed Algorithm

The proposed metaheuristic uses original GWO [8] with one more level in delta wolf for exploitation and levy flight operator for raising the exploration capability. Hence the proposed algorithm is named Grey Wolf Optimization with Levy Flight (GWOLF). The hierarchy of Canis lupus grey wolves for the proposed algorithm is shown in Fig 4. It is said that in spite of alpha, beta, delta, and omega wolves, delta wolves are further split into delta^m and kappa wolves to improve the exploitation of the algorithm. The proposed algorithm respects the concept of original GWO but on the other side, it is also claimed that the proposed concept is closer to the reality of the procedure which is followed by grey wolves. The fundamental concept behind the proposal of GWOLF is that the prey considered here is not static but it will try to run away from the alpha wolves and hence it is considered to be dynamic wherein original GWO, there is no discussion about the prey position. The alpha wolves will try to get hold of prey. Hence alpha wolves are updated as per the prey position wherein the prey is not static and hence it is modeled with the help of Levy distribution. The beta, delta^{m,} and kappa wolves will be updated as per (4), (5) and (7) respectively. This distribution will not only follow the behavioral hunt of grey wolves but also enhances the exploration in the algorithm.

The best position is saved as the position of alpha wolves. Second, third and fourth best position is saved as position of beta, delta^m and kappa wolves. The alpha and beta position are considered to be closest to the prey but prey is running far from the alpha position. The prey position is updated by adding a Levy flight to the alpha position.

$$\overrightarrow{X_p} = \overrightarrow{X_\alpha}(t) + a' \oplus Levi(b) \quad (14)$$

Where a' is said to be step size and lies in between (0,1) and b is power law index which lies in between (0,2).

Equation (3) is modified as

$$\vec{X}_1 = \overrightarrow{X_\alpha}(t) - \overrightarrow{A_1}.\overrightarrow{C_1}.\left(\overrightarrow{X_p}(t) - \vec{X}(t)\right) \quad (15)$$

Rest of the wolves will be following the same strategy to update their values as per (4-5) and (7)

The new position of rest wolves will be given as a weighted value of alpha, beta, delta^m, and kappa wolves rather than taking an average of all the positions as in original GWO in (6).

$$\vec{X}(t+1) = 0.4 * \vec{X}_1 + 0.3 * \vec{X}_2 + 0.2 * \vec{X}_3 + 0.1 * \vec{X}_4 \quad (16)$$

The flowchart and pseudo code for GWOLF are reported in Fig 6 and Fig 7 respectively.

TABLE I. Parameters of Optimizers

Optimizer	Description
PSO	*ω1=0.2, ω2=0.9, c1=2, c2=2*
GWO	*a=2*
GWOLF	*b (Power law index)=2, n (Number of steps)=1, a'=2*

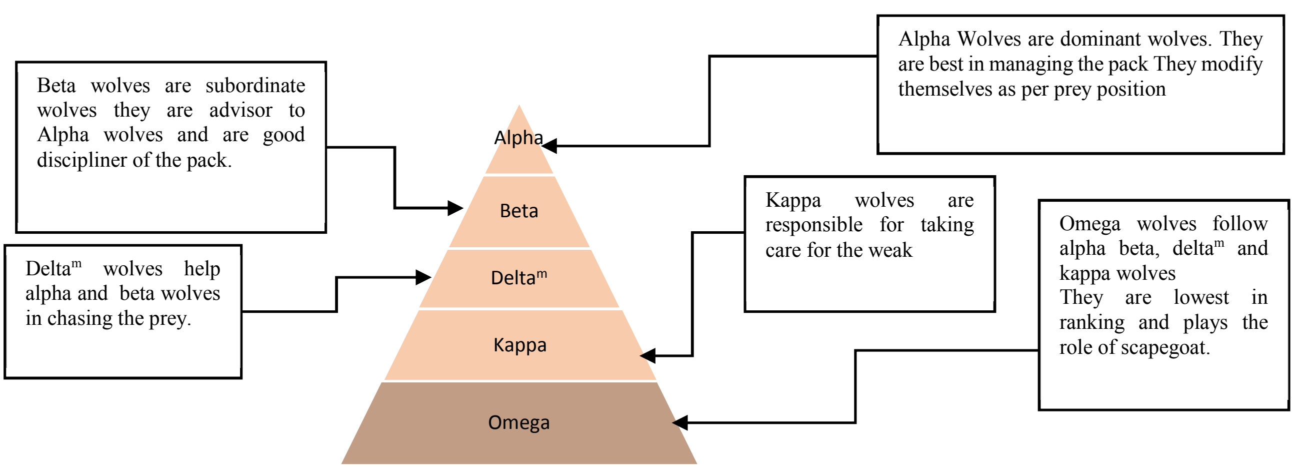

Fig. 4. Social Hierarchy of wolves and their characteristics in GWOLF

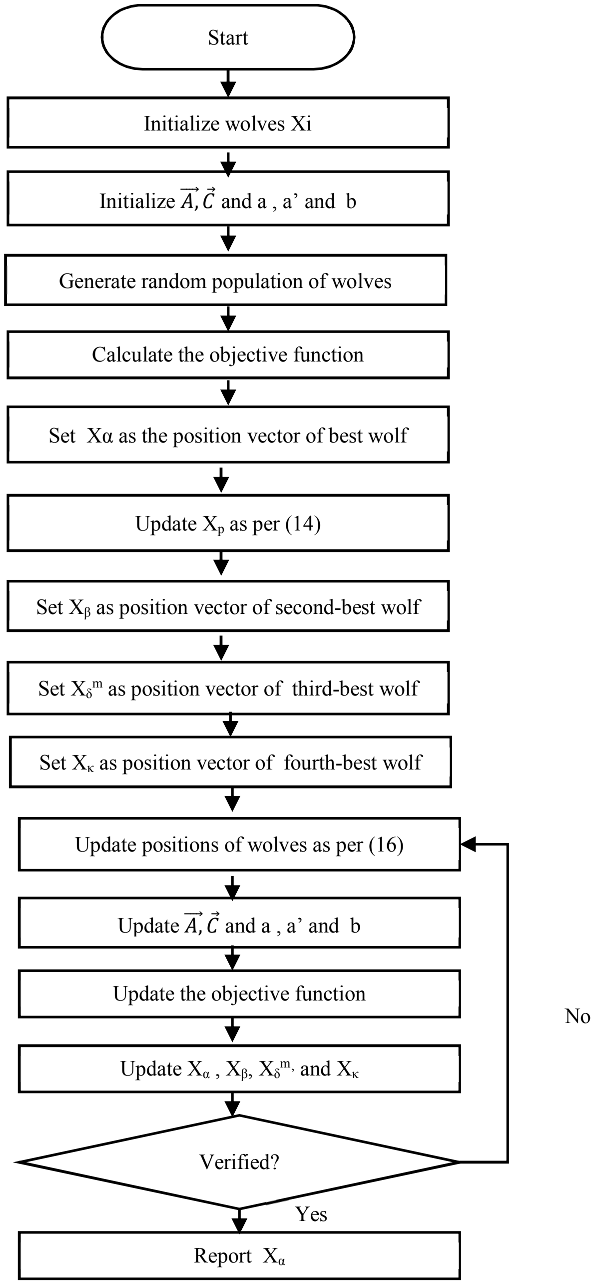

Fig. 5. Flowchart of GWOLF algorithm

Initialize the grey wolf populations Xi (i=1,2,........,n)
Initialize $\vec{A}, \vec{C}$ and a , a’ and b
Calculte the fitness of each search agent
X_α= *position vector of the best search agent*
X_β= *position vector of the second best search agent*
$X_\delta{}^m$ =*position vector of the third best search agent*
X_κ= *position vector of the fourth best search agent*
$X_p = X_\alpha + a'{*}Levy\,(b)$
While *(t<Maximum number of iterations)*
  for *each search agent*
    update the position of the search agent by equation (12)
  end for
  Update a, b, A and C and Prey position
  Calculate the fitness of all search agents
  Update X_α, X_β, $X_\delta{}^m$, X_κ *and* X_ω
  t=t+1
End while
Return X_α

Fig. 6. Pseudo code for GWOLF algorithm

TABLE II. BEST RESULTS BY GWOLF

Function	Characteristics	Optimum solution	GWOLF Best
F01	Unimodal, High Dimensional	0.00000	0.0000
F02	Unimodal, High Dimensional	0.00000	0.0000
F03	Unimodal, High Dimensional	0.00000	0.0000
F04	Unimodal, High Dimensional	0.00000	0.0000
F05	Unimodal, High Dimensional	0.00000	26.0288
F06	Unimodal, High Dimensional	0.00000	0.0001
F07	Unimodal, High Dimensional	0.00000	0.0006
F08	Multi modal, High Dimensional	-418.9×n	-7299.6109
F09	Multi modal, High Dimensional	0.00000	0.0000
F10	Multi modal, High Dimensional	0.00000	0.0000
F11	Multi modal, High Dimensional	0.00000	0.0000
F12	Multi modal, High Dimensional	0.00000	0.0239
F13	Multi modal, High Dimensional	0.00000	0.3151
F14	Multi modal, Low Dimensional	1.00000	0.9980
F15	Multi modal, Low Dimensional	0.00030	0.0003
F16	Multi modal, Low Dimensional	-1.03160	-1.0316
F17	Multi modal, Low Dimensional	0.39800	0.3979
F18	Multi modal, Low Dimensional	3.00000	3.0000
F19	Multi modal, Low Dimensional	-3.86000	-3.8628
F20	Multi modal, Low Dimensional	-3.32000	-3.3220
F21	Multi modal, Low Dimensional	-10.15320	-10.1530
F22	Multi modal, Low Dimensional	-10.40280	-10.4028
F23	Multi modal, Low Dimensional	-10.53630	-10.5359

IV. RESULTS AND DISCUSSION

For exploring the capabilities of GWOLF, the simulation on various benchmark problems is executed and is compared with competitive algorithms like GWO and PSO as reported earlier [8]. Parameters used by PSO, GWO, and GWOLF are reported in Table I. For these investigations, every metaheuristic is performed on Windows 10 framework utilizing Intel® Core™ i5-8250U CPU @1.60GHz, Installed RAM 8GB and MATLAB 2017b and x64 - based processor.

Table II shows the best results obtained by GWOLF for 30 runs. The values of results close to 8 decimal places are considered to be equal. GWOLF is likewise contrasted with PSO and GWO in regard to average values and standard deviation value in Table III. The runtime performance of GWOLF, GWO, and PSO for benchmark problems is also shown in the convergence curves depicted in Fig.7

A. Analysis for exploitation ability

Functions from F01 to F07 are unimodal functions, i.e., they have only one global optimum. Hence the exploitation capabilities of GWOLF are investigated by looking at the behavior of metaheuristic for these functions. Table III shows the comparison of GWOLF with GWO and PSO on the average value and standard deviation of results obtained from 30 independent runs.

It has been found from Table III that GWOLF performs very well for optimum value and gives better or competitive results as per the average value and standard deviation when compared with GWO and PSO for unimodal functions.

The proposed algorithm shows better average values for F01, F02, and F07 and gives competitive results from F03-F06. It is very well perceived from the outcomes on F01-F07 that the GWOLF is fit for accomplishing satisfactory solutions for unimodal problems and hence works well as a good exploiter.

B. Analysis for exploration

Functions from F08-F23 are multimodal functions having multiple optima. Hence the exploration capabilities of GWOLF is investigated by looking at the behavior of the metaheuristic for these functions.

Based on the results of Table II, it is clearly shown that the proposed GWOLF approaches optimum value. The average value for F09-F12, F19 is better in GWOLF whereas all other metaheuristics show competitive results for F16-F18. It is also observed that GWOLF is giving better average values for F20-F23 than GWO.

Table III shows that the proposed GWOLF either gives better or competitive results as compared to GWO and PSO [8]. Hence it can be concluded that the proposed algorithm is a good explorer.

C. Analysis for convergence

Fig 7 shows the comparative convergence curve for GWOLF, GWO, and PSO. Here the comparison is made with the initial version of GWO and PSO. Here convergence curve shows that GWOLF outperforms GWO and PSO in F01-F05, F09-F10, and F14-F23 in terms of fast convergence. This shows that GWOLF performs well both in exploitation and exploration.

TABLE III. COMPARATIVE RESULTS FOR BENCHMARK PROBLEMS USING GWOLF, GWO [8] AND PSO [8]

Function	**Algorithm**	**Average**	**Standard Deviation**
F1	GWOLF	9.5160E-29	1.9019E-28
	GWO	6.5900E-28	6.3400E-05
	PSO	1.3600E-04	2.0200E-04
F2	GWOLF	3.0680E-17	6.5632E-18
	GWO	7.1800E-17	2.9014E-02
	PSO	4.2144E-02	4.5421E-02
F3	GWOLF	3.2387E-05	6.3868E-05
	GWO	3.2900E-06	7.9150E+01
	PSO	7.0126E+01	2.2119E+01
F4	GWOLF	6.5551E-07	7.1800E-17
	GWO	5.6100E-07	1.3151E+00
	PSO	1.0865E+00	3.1704E-01
F5	GWOLF	2.7014E+01	8.7328E-01
	GWO	2.6813E+01	6.9905E+01
	PSO	9.6718E+01	6.0116E+01
F6	GWOLF	9.5093E-01	3.8722E-01
	GWO	8.1658E-01	1.2600E-04
	PSO	1.0200E-04	8.2800E-05
F7	GWOLF	1.9336E-03	1.0222E-03
	GWO	2.2130E-03	1.0029E-01
	PSO	1.2285E-01	4.4957E-02
F8	GWOLF	-5.7584E+03	7.1995E+02
	GWO	-6.1231E+03	4.0874E+03
	PSO	-4.8413E+03	1.1528E+03
F9	GWOLF	1.5920E-01	8.7198E-01
	GWO	3.1052E-01	4.7356E+01
	PSO	4.6704E+01	1.1629E+01
F10	GWOLF	6.2705E-14	1.2914E-14
	GWO	1.0600E-13	7.7835E-02
	PSO	2.7602E-01	5.0901E-01
F11	GWOLF	1.4915E-03	8.1694E-03
	GWO	4.4850E-03	6.6590E-03
	PSO	9.2150E-03	7.7240E-03
F12	GWOLF	5.3249E-02	2.2069E-02
	GWO	5.3438E-02	2.0734E-02
	PSO	6.9170E-03	2.6301E-02
F13	GWOLF	7.6677E-01	2.5320E-01
	GWO	6.5446E-01	4.4740E-03
	PSO	6.6750E-03	8.9070E-03
F14	GWOLF	4.6474E+00	4.5903E+00
	GWO	4.0425E+00	4.2528E+00
	PSO	3.6272E+00	2.5608E+00
F15	GWOLF	5.0856E-03	8.5741E-03
	GWO	3.3700E-04	6.2500E-04
	PSO	5.7700E-04	2.2200E-04
F16	GWOLF	-1.0316E+00	3.7905E-08
	GWO	-1.0316E+00	-1.0316E+00
	PSO	-1.0316E+00	6.2500E-16
F17	GWOLF	3.9789E-01	2.8764E-06
	GWO	3.9789E-01	3.9789E-01
	PSO	3.9789E-01	0.0000E+00
F18	GWOLF	3.0000E+00	4.5115E-05
	GWO	3.0000E+00	0.0E+00
	PSO	3.0000E+00	1.3300E-15
F19	GWOLF	-3.8610E+00	2.8212E-03
	GWO	3.8626E+00	3.8628E+00
	PSO	-3.8628E+00	2.5800E-15
F20	GWOLF	-3.2880E+00	5.8242E-02
	GWO	-3.2865E+00	3.2506E+00
	PSO	-3.2663E+00	6.0516E-02
F21	GWOLF	-9.5657E+00	1.8248E+00
	GWO	-1.0712E+01	9.1402E+00
	PSO	-6.8651E+00	3.0196E+00
F22	GWOLF	-1.0401E+01	1.2418E-03
	GWO	-1.0402E+01	8.5844E+00
	PSO	-8.4565E+00	3.0871E+00
F23	GWOLF	-1.0264E+01	1.4812E+00
	GWO	-1.0534E+01	8.5590E+00
	PSO	-9.9529E+00	1.7828E+00

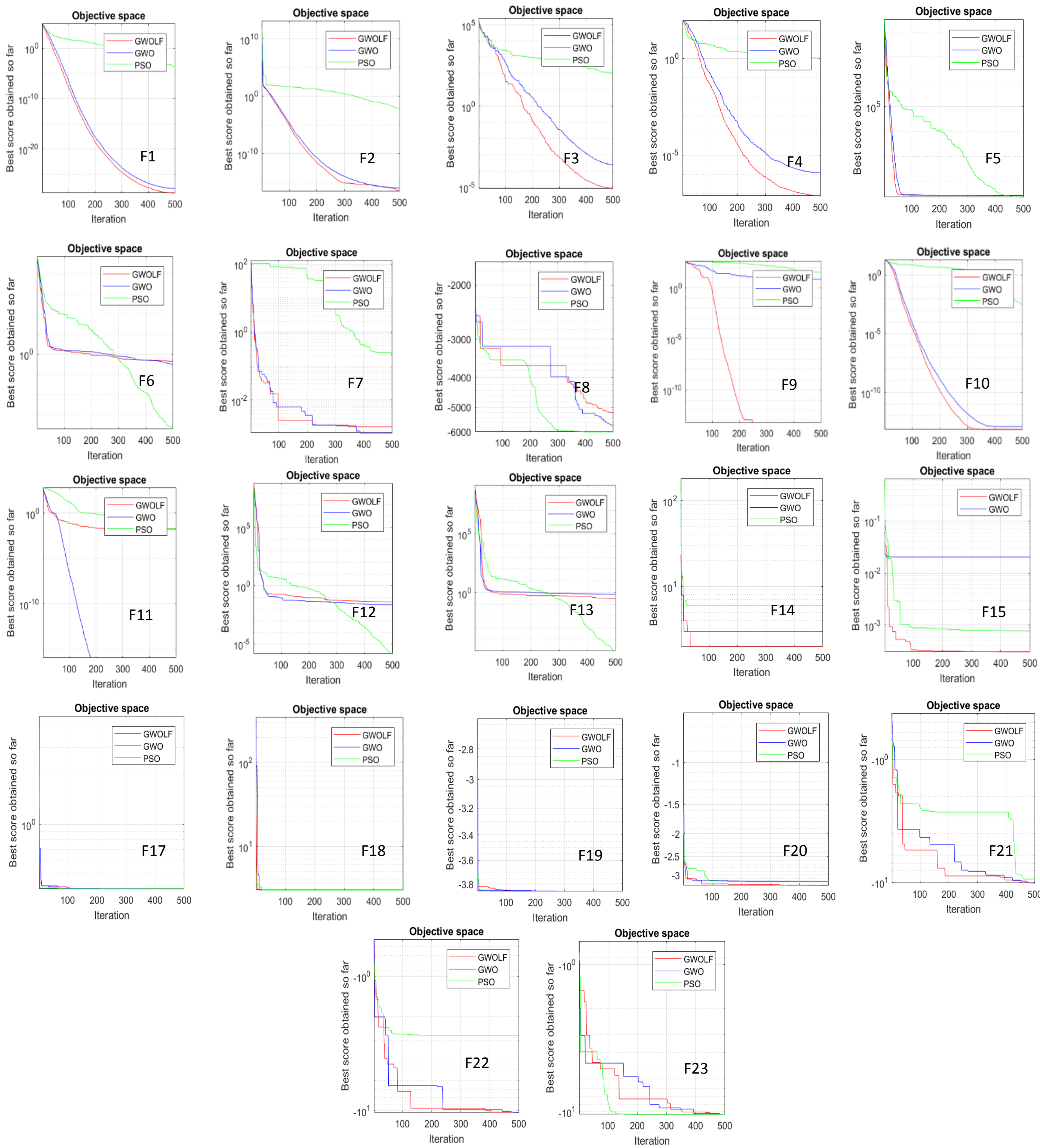

Fig. 7. Comparative convergence curve for different benchmark problems by GWOLF, GWO [8] and PSO [8] algorithms.

V. Conclusion

Grey wolf optimizer has been applied to various applications in different fields successfully, although there exists the problem of exploration and stagnation into local optima. Hence a novel modification in GWO has been incorporated which not only makes the exploitation better but also helps in enhancing the exploration capabilities of the original GWO. This paper talks about the dynamic nature of prey and the hierarchy of how alpha wolves, then beta and after that delta and omega wolves are chasing the prey as per the realistic theory of Canis lupus Grey wolf's hunt strategy. In this paper, firstly exploitation is enhanced by adding one more level in the hierarchy of Canis lupus wolves. Secondly, exploration is enhanced by adding the dynamic behavior of prey wherein the alpha wolves are the first to follow the prey and then other wolves will be following alpha wolves thereafter. Dynamism is added to prey with the help of Levy flight distribution. The proposed metaheuristic has been verified on 23 benchmark problems. It has been found that the proposed metaheuristic, GWOLF gave either better or competitive results compared to the conventional GWO and PSO.

Hence from the results showcased in this paper, conclude that GWOLF is an efficient optimizer for both exploitation and exploration problem.

Future scope of work includes implementation of GWOLF for multi objective optimization problem.

REFERENCES

[1] Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. Applied Computational Intelligence and Soft Computing, 2016.

[2] Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning.

[3] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1942-1948). IEEE.

[4] Simon, D. (2008). Biogeography-based optimization. IEEE transactions on evolutionary computation, 12(6), 702-713.

[5] Dorigo, M., & Di Caro, G. (1999, July). Ant colony optimization: a new meta-heuristic. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 2, pp. 1470-1477). IEEE.

[6] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.

[7] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680.

[8] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

[9] Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. Applied Computational Intelligence and Soft Computing, 2016.

[10] Long, W., Jiao, J., Liang, X., Cai, S., & Xu, M. (2019). A random opposition-based learning grey wolf optimizer. IEEE Access, 7, 113810-113825.

[11] Hu, P., Chen, S., Huang, H., Zhang, G., & Liu, L. (2018). Improved alpha-guided Grey wolf optimizer. IEEE Access, 7, 5421-5437.

[12] Gupta, S., & Deep, K. (2019). A novel random walk grey wolf optimizer. Swarm and evolutionary computation, 44, 101-112.

[13] Gupta, S., & Deep, K. (2019). An opposition-based chaotic grey wolf optimizer for global optimisation tasks. Journal of Experimental & Theoretical Artificial Intelligence, 31(5), 751-779.

[14] Gupta, S., & Deep, K. (2019). An efficient grey wolf optimizer with opposition-based learning and chaotic local search for integer and mixed-integer optimization problems. Arabian Journal for Science and Engineering, 44(8), 7277-7296.

[15] Long, W., Cai, S., Jiao, J., & Tang, M. An efficient and robust grey wolf optimizer algorithm for large-scale numerical optimization. Soft Computing, 1-30.

[16] Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. Applied Soft Computing, 60, 115-134.

[17] Bhateja, A. K., Bhateja, A., Chaudhury, S., & Saxena, P. K. (2015). Cryptanalysis of vigenere cipher using cuckoo search. Applied Soft Computing, 26, 315-324.

[18] Viswanathan, G. M., Afanasyev, V., Buldyrev, S. V., Murphy, E. J., Prince, P. A., & Stanley, H. E. (1996). Lévy flight search patterns of wandering albatrosses. Nature, 381(6581), 413-415.