

A New Galactic Swarm Optimization Algorithm Enhanced with Grey Wolf Optimizer for Training Artificial Neural Networks

Geraldine Bessie Amali.D^a, Mahammed Mohsina^b, Umadevi K S^c

a School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India – 632014

E-mail: geraldine.amali@vit.ac.in

b School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India – 632014

E-mail: mahammed.mohsina2019@vitstudent.ac.in

c School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India – 632014

E-mail: umadeviks@vit.ac.in

Abstract: This paper proposes a new Galactic Swarm Optimization (GSO) algorithm enhanced with Grey Wolf Optimizer (GWO). The proposed algorithm is used to train a feedforward Neural Network for function approximation. Galactic swarm optimization is a popular swarm algorithm that has been used to solve optimization problems. It is motivated by stars movement and the superclusters of a galaxy in the universe. The algorithm allows using multiple levels of exploitation and exploration of search space. At the explorative level, different sub-populations independently explore search space and at the exploitation level, the best solution of different sub-populations is considered as a super swarm and moves towards finding the best solution position found by the super swarm. The algorithm uses Particle Swarm Optimization (PSO) algorithm's update equation in both levels. PSO has been proven to get stuck in the local minimum due to its ability to converge prematurely. In this work, Galactic swarm optimization enhanced with grey wolf optimizer is proposed. The use of the entire pack of wolves for exploring the search space in the GWO has proven to escape local minima. Thus, the GSO's explorative phase is done with the GWO and for the exploitation phase quickly converging PSO is used. The proposed algorithm ability is tested by training a feedforward neural network for function approximation of benchmark optimization problems. The proposed GSOGWO outperformed the classical GSO in most of the functions.

Keywords: Artificial Neural Networks, Global Optimization, Galactic Swarm Optimization, Grey Wolf Optimizer algorithm, Swarm algorithms.

1. Introduction

Optimization is an important tool in any decision-making scenario. Global optimization refers to locating the most optimal point in a multidimensional space. Global optimization has numerous applications in science and engineering and it is the core of many social science disciplines such as business, economics, politics, and governance. Optimization yields best results in terms of maximizing quality or minimizing the cost for solving problems [15]. Deterministic algorithms have been used in the past to solve optimization problems. However, their ability to get stuck in the local minima has led to a wide of metaheuristic search approaches. Metaheuristic algorithms have the ability to escape local minimum due to their stochastic nature. It is classified into on 4 types. They are evolutionary based, physics based, swarm intelligence based and Human based. Evolutionary algorithms are motivated by Evolution [1]. Differential evolution is one of most popular algorithms which has been utilized in many applications in engineering and industry areas [22]. Physics based algorithms are inspired by physical laws such as the electromagnetic-force algorithm and Gravitational-force algorithm [2]. Swarm intelligence algorithmic rules mimic social motion on swarms, herd & group of animals in nature [1]. Human based algorithm that mimics the human behaviour is also used as search techniques [8]. One of the most famous swarm intelligence algorithms is Particle Swarm Optimization (PSO) that is inspired by the behaviour of a bird flock and schooling fishes to tackle optimization problems [20]. Metaheuristic algorithms are used to find the balance on local search randomization and used on global optimization [9,10]. It approaches multiple basic levels of operation namely exploration and exploitation. Local search approaches are used for exploiting the search space and to identify the best possible solution out of the possible solutions that have been found in exploration phase. Exploring the search space helps escape local minima and help to identify optimal global values. Thus, a combination of these two

approaches helps achieve global optimal solutions over a random variety of solutions. In order to get good results, researchers are trying to improving the rate of the algorithms on hybridization or some enhancements over the architecture of the algorithm on more multiswarm developing to achieve better models for continuous optimization problem and discrete problems [18]. In this paper, the Galactic swarm optimization algorithm is used for exploring the search space and Grey Wolf Optimizer is used for exploitation. The proposed GSOGWO is used to train the feedforward neural network for function approximation.

2. Related Work

Artificial Neural Networks (ANN's) are part of computing systems that are used to simulate the ways of how humans gather information and analyse information. It solves difficult or impossible problems for humans. ANN have self learning capability that enables them to produce good results and enable more data, thus ANN's have thousands or hundreds of neuron's called processing units. The neural network technique is advantageous over other techniques used for pattern recognition and classification [10]. The performance of the NN depends heavily on the interconnection weights and biases. Approximating the weights of the neural network is the major challenge [19]. The success of ANN's depended on training algorithms that approximate the weights of the NN depending on the error generated by the NN in approximating the target function. The cost function of neural net training is used to select the best association between biases and weights and to minimize classification errors [9]. In the quantum tunnelling particle swarm optimization algorithm [6,20] the author trained a feed-forward neural network (FNN) of having 3 layers of neurons organized into the input layer, hidden layer, and output layer. Backpropagation is a gradient based method and it is regarded as one of the most widely techniques used to train artificial neural networks [10]. The performance of the networks can be increased using feedback information obtained from the distinct between the actual and the desired output. This information is used to change the connection between the neurons of the input layer to the actual result that coordinates the desired one [17]. SeyedaliMirjalili proposes grey wolf optimization on training Multi Layered Perceptron (MLP) on eight standard datasets having five classifications and 3 approximate functions on dataset and used to benchmarking execution of implemented methods and results are compared to different Metaheuristics and simulation shows GWO with MLP gives better accuracy in classification [21]. In Galactic Swarm Optimization with Tree Seed Algorithm, Author did hybridization TSA as a search method within the GSO and experimental study on 12 benchmark functions, the GSO with TSA method was compared with the traditional TSA method. Results show GSO with the TSA method is more successful than the traditional TSA method [14]. Binh Minh Nguyen proposes a new hybrid algorithm called galactic swarm with evolution whale optimization to tackle Global optimization problems, HGEW gives good outcomes but can't ensure good result in the global optimum. Additionally, they have used Levy-Flight on HGEW to enhance the better ability of search on best global optima [16]. This paper is coordinated as follows: firstly, the mathematical formulation on problem is shown, secondly, metaheuristic algorithm is discussed, thirdly a new hybrid galactic swarm algorithm enhanced with Grey Wolf Optimizer is introduced and proposed algorithm is compared on different benchmarking artificial neural network function approximate problems.

3. Problem Formulation

Feed-forward Neural Network (FNN) is basic artificial neural network. In this model, information flows in only one- direction. The network is typically made of an input layer, output layer and one or more hidden layers. In this paper, an FNN is modelled with one input, one output, and one hidden layer. The goal in using a NN is to make the NN mimic the target output. In other words, it is essential to reduce the error generated by the NN. The error is obtained by comparing the output of the NN with the target output. The more the NN deviates from the actual output, the more will be the error obtained. Thus, training a neural network can be viewed as an optimization problem, where the objective is to minimize the mean square error between the NN's output and the target output.

The output of the NN depends on the weights of the NN performance. Thus, identifying the weights of the NN can be envisioned as a search problem, where the best solution is the optimal values of the NN that reduce the mean square error the most. In this paper, by applying the optimization algorithm GSO(GWO-PSO) we are minimizing the Mean Square Error (MSE) between target output and the NN's output. The architecture of feed-forward neural network used in this work consists of two inputs, one Hidden layer, and one output, and sigmoid function activation. Supervised learning, is done with inputs and outputs in the form of (X^i, Y^i) , where $i=1$ to M is no of training samples. X is input & Y is targeted output on i^{th} training samples. let $F(X)^i$ is the neural net output and Θ is the sigmoid function at input X . This output is represented as

$$F(X) = \Theta \left(\sum_{j=1}^n W_{ij} \cdot X_j + b_i \right) \quad (1)$$

where W_{ij} is inter-connection of weights on i^{th} & j^{th} neuron; b_i is bias of i^{th} neuron. Mean Square Error (MSE) is represented as follows:

$$E(W) = \frac{\sum_{i=1}^M (Y_i - F(X_i))^2}{M} \quad (2)$$

Equation (2) represents the cost function of a neural network that measure an average of difference on the neural net output and target on training samples. Forward propagation is applied to the output network for the random generated weights. The weights of the neural net are adjusted and considered on the minimization of cost-function. Because each neuron is applied to an activation function to input weight to calculate output on function $F(X)$ and is calculated to the neural network and $E(W)$ for the error function.

4. Methods

4.1 Grey Wolf Optimizer

A Grey Wolf Optimizer is proposed by SeyedaliMirjalili. It inspires on Grey wolves called *Canis-lupus*. The GWO algorithm mimic a law called hunting mechanism on grey wolf. The wolf as highest predator for searching for the food chain, it has the strongest capacity to attack prey [21]. Grey wolf prefers to reside in a pack of group size is 5 to 12. Mainly it is 4 types of wolves as highest level as alpha(α), it is liable to take decisions to sleep and hunt, second level is beta(β), it orders to primary level wolves and gives feedback to alpha, third level omega(ω) has to send all the dominating wolf's. These are the wolf's that are allowed to eat and remaining wolves considered as delta(δ) and these are used for re-enacting the leadership hierarchy [24]. Additionally, three main stages are there in hunting called searching the prey(food), en-circling the prey, and attack for prey(food) implements for performing optimization [24].

4.1.1 Encircling prey:

Grey wolves encircle the prey on hunting. the mathematical model of en-circling behaviour of equation are shown [24]

$$A_p = |B \cdot z_i(k) - z(k)| \quad (3)$$

$$Z_p(k + 1) = z_p(k) - A_p \quad (4)$$

Here k is indicated by current-iteration, B & W are the co-efficient vector, A_p -vector position of prey and z is position vector on wolves. r_1 and r_2 are random values, T_{max} is the maximum number of iterations

$a = 2 - 2t/T_{max}$ is linearly decreased from 2 to 0

$$W_1 = 2ar_1 - a, B_1 = 2r_2 \quad (5)$$

$$W_2 = 2ar_1 - a, B_2 = 2r_2 \quad (6)$$

$$W_3 = 2ar_1 - a, B_3 = 2r_2 \quad (7)$$

4.1.2 Hunting:

Grey wolves are having propensity for recognition the location(area) of prey and to encompass. the Hunting are normally guided on alpha, beta, delta may likewise take an interest on hunting incidentally. Regardless, they are unsure on area of ideal prey. Therefore, the first 3 initial best solutions acquired & oblige another search agent (counting omega) for updating the position as per situation of the best pursuit agents. the proposed equation is in this regard [21].

$$A\alpha = |B_1 \cdot z_\alpha(k) - z| \quad (8)$$

$$A\beta = |B_2 \cdot z_\beta(k) - z| \quad (9)$$

$$A\delta = |B_3 \cdot X_\delta(k) - z| \quad (10)$$

$$Z_1 = X_\alpha - W_1(A_\alpha) \quad (11)$$

$$Z_2 = X_\beta - W_2(A_\beta) \quad (12)$$

$$Z_3 = X_\delta - W_3(A_\delta) \quad (13)$$

$$Z(k + 1) = Z_1 + Z_2 + Z_3 / 3 \quad (14)$$

4.1.3 Search for Prey:

Grey wolve generally searches prey as per position on alpha, beta, omega & delta. They separate one another to searching for food(pre) and combines for attacking prey. In the mathematical model, we utilize A as random values in range $[2a, -2a]$. In randomization esteem $|A| < 1$ the wolves are constrained for attacking prey(food) and if $|A| > 1$, the population members are not obligated to separate the prey.

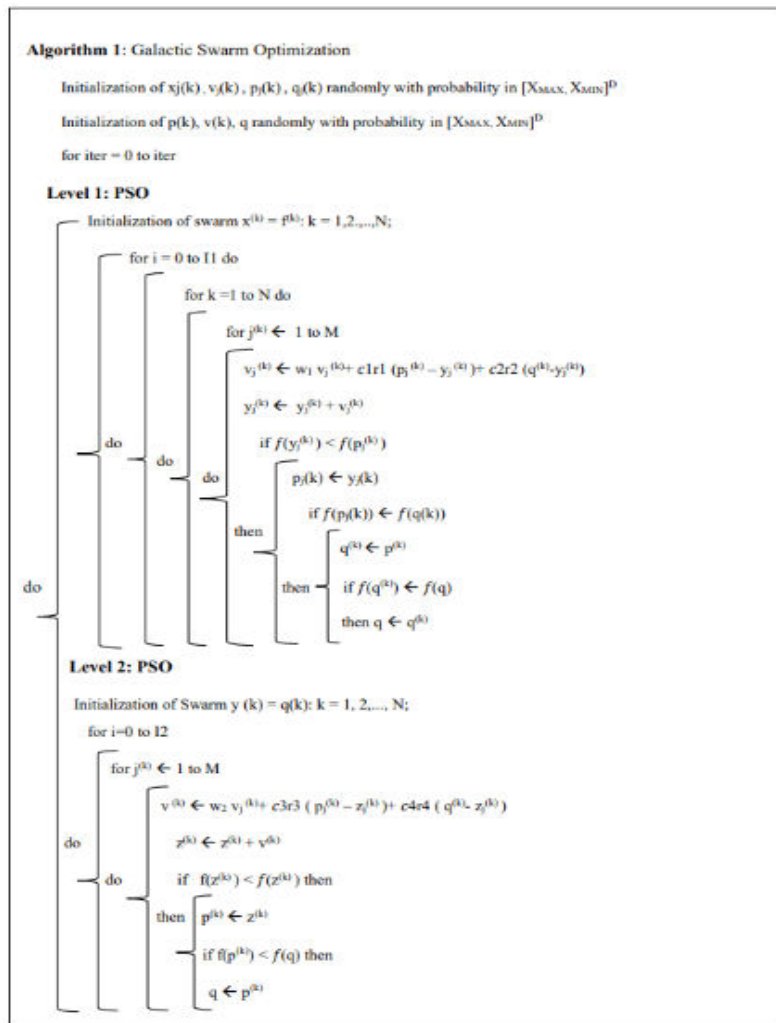
4.2. Galactic Swarm Optimization

Muthiah-Nakarajan proposes a new Galactic Swarm Optimization on Global optimization [12]. GSO inspires the moment of stars, motion clusters & superclusters in the whole universe. In the GSO, algorithm stars revolves around the cosmic(galaxy) system. The cosmic revolve around the superclusters. Each cosmic system presented the global-best solutions, GSO having many algorithms to mimic the idea of a searching solution, beginning some random area in bounds. GSO is motivated by the behaviour of stars in universes of superclusters [12]. It exploits first PSO on utilizing various cycle patterns of exploration & exploitation on discovering better solution and new solutions. Because of the execution and effectiveness of algorithms, GSO has used for real-life issues too as in looking through worldwide advancement.

Firstly, GSO is affected under gravity, stars in the universe and pulled to the other star having great prominent-gravity. From this idea, development of stars inside the universe is development on cosmic systems to replicate in GSO calculation on accompanying the standards:

- In individuals, every system pulled into greater ones (better arrangements) in the universe. Fascination measure is performed on utilizing the PSO calculation.
- Global best on all cosmic systems are picked to treat as a super swarm. the PSO calculation is utilizing another time to address development of particles in the super swarm.

Muthiah-Nakarajan [12], original GSO framework pseudocode is presented in the Algorithm



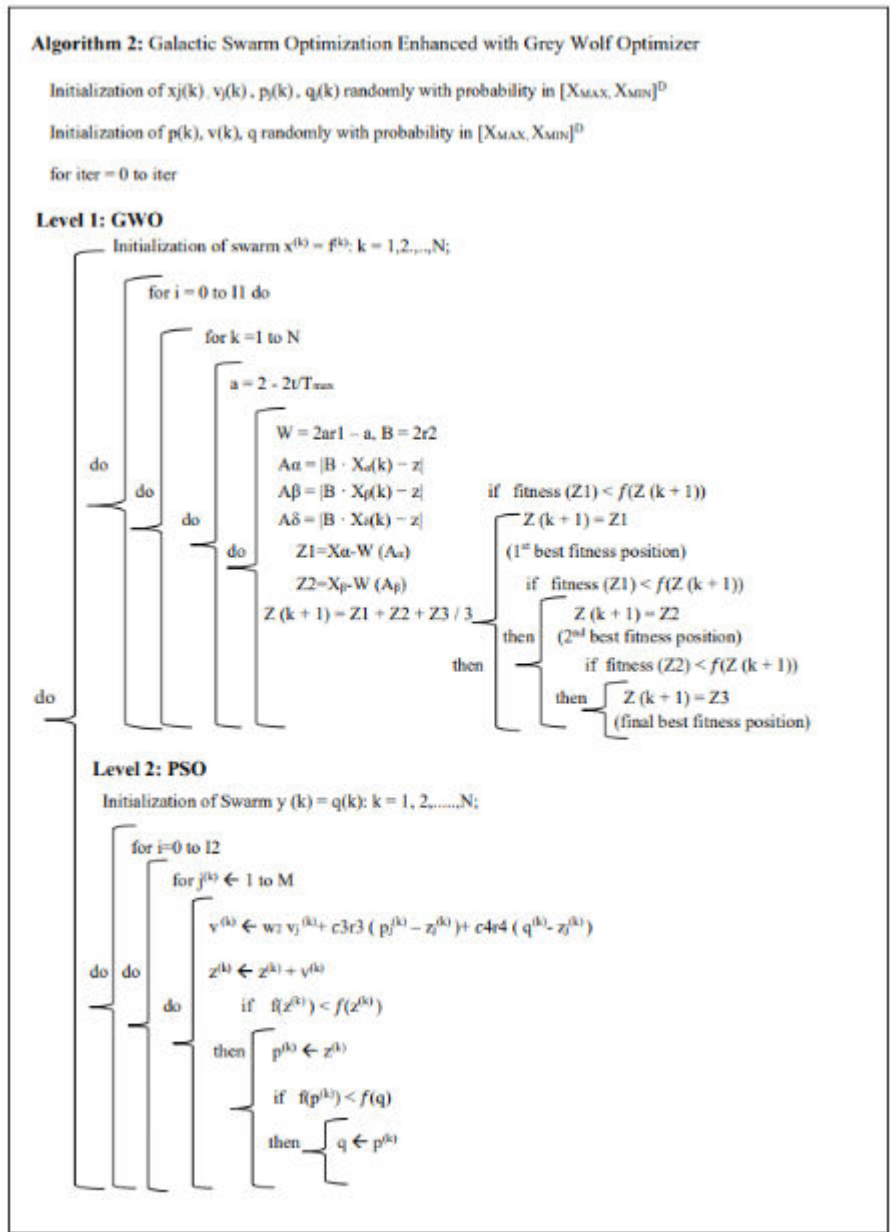
4.3. Hybridization of Galactic Swarm Optimization Enhanced with Grey Wolf Optimizer

Optimization techniques both exploitation & exploration phases of the algorithm are incredible (powerful) to achieve a good result. Exploitation deals on local optima solutions that obtain by exploring in search space and at other phase, exploration deals on global optima searching of solution in search space. GSO algorithm is not a method but also layout for optimization problems. As in the framework, more optimization algorithms are proposed in the primary phase, secondary phase, or both phases. PSO was used in the primary and secondary level in the original layout of the GSO algorithm, this algorithm having good exploitation characteristics, mostly parameters on PSO algorithms work on exceptionally explorative action at starting within consistent steady progress to highly & exploitative transition action towards final. The exploration action allows to search for the closest to good local minima and global minima, exploitation-action helping to localize the global minimum effectively. The schema breaks down on the multimodal function when the model merges to a low local minimum on the exploration of stage results in suboptimal solutions. Hence, the new implementation is implemented on the GWO algorithm. This algorithm has incredible exploration features than PSO. In the implementation, we are obtaining a new incredible (powerful) model by training neural networks. In the proposed algorithm, searching for prey (food) is taken as the source on the different position such as alpha, beta, and gamma as candidate solution in PSO algorithm and by training the best solutions of the algorithm with inputs, hidden and output on weights we are calculating the mean square approximate error on multimodal functions to get an accurate output of solution.

The proposed algorithm having parameters N is no of subswarm, M is a size on subswarm k, I_i is a number of iterations at k, c_1, c_2, c_3, c_4 are acceleration coefficients at two levels, r_i is randomly distributed at [-1,1], f is the objective function, $x_j^{(k)}$ is position on particle j in sub swarm k, $v_j^{(k)}$ is velocity of particle at $x_j^{(k)}$, $g^{(k)}$ is global best solution (gbest) of subswarm i, g is the global best solution of a super swarm.

Table 1. Represents parameters used by the proposed algorithm

N-no of sub swarm	M-size on sub swarm	I1-iteration	I2-iteration	EP _{max}	c1,c2,c3,c4
10	5	198	1000	50	2.05



5. Results & Discussion

The overall performance on the GSO(PSO-PSO) algorithm was compared with GSO(GWO-PSO) on global optimization algorithm on different benchmarks on feed-forward neural network function approximate problems. The structure of the feed-forward network consists of two inputs, ten hidden layers, and one output. This feed-forward network was trained for learning the test suite function on the proposed hybrid GSO(GWO-PSO) and original GSO(PSO-PSO) optimization algorithm. The Dejonge test suite has challenges on global optimization problems as flat regions, narrow ridge on local minimum. Dejonge test functions are compared on the execution of the proposed algorithms and parameters are given as Mean Square Error (MSE) obtained by neural net associated

output & trains on inputs on the algorithm. Below are comparison results on the plotting of 20 benchmark problems approximating the functions and trained by ANN on GSOPSO and GSOGWO algorithms. Rastrigini Function is a multimodal function of having a cosine module to produce frequently local minima. Ackley is a non-separable and differential function of having multiple minima. Bohachevsky is valley shaped unimodal function. Griewank Function is similar to Rastigini of having local minima distributed. Schwefel function is a multivariate bowl-shaped function or plate-shaped function. A shubert function is a multivariate function having many local minima.

Table 2: Represents the benchmark functions with global optima values used for comparison

S.No	Test Functions		Global minima
1	Rastrigin	$f(a,b)=10n+\sum_{i=1}^n(a^{2i}+b^{2i}-10\cos(2\pi ai+2\pi bi))$	[0,0]
2	BohachevskyN.1	$f(a,b)=a^2+2b^2-0.3\cos(3\pi a)-0.4\cos(4\pi b)+0.7$	[0,0]
3	Griewank	$f(a,b)=1+\sum_{i=1}^n(a^{2i}+b^{2i}/4000)-n\pi i=\cos(a_i)*\cos(b_i/\sqrt{2})+1$	[0,0]
4	Ackley N. 2	$f(a,b) = -200e^{-0.2\sqrt{a^2+b^2}}$	[0,0]
5	Styblinski tank	$f(a) = f(a_1, \dots, a_n) = 1/2^n \sum_{i=1}^n (a_i^4 - 16a_i^2 + 5a_i)$	[-2.9035, -2.9035]
6	Schwefel 2.22	$f(a) = \sum_{i=1}^n a_i + \prod_{i=1}^n a_i $	[0,0]
7	Shubert 3	$f(a) = f(a_1, \dots, a_n) = \sum_{i=1}^n \sum_{j=1}^5 \sin((j+1) a_i + j)$	[≈ -29.67]
8	Rosenbrock	$f(a, b) = 100(b-a^2)^2 + (a-1)^2$	[1,1]
9	Beale	$f(a, b) = (1.5-a+ab)^2 + (2.25-a+ab^2)^2 + (2.625-a+ab^3)^2$	[0,0]
10	Three hump Camel	$f(a, b) = 2a^2 - 1.05a^4 + a^6/6 + ab + b^2$	[0,0]
11	Sphere	$f(a, b) = a^2 + b^2$	[0,0]
12	Alpine N .1	$f(a) = f(a_1, \dots, a_n) = \sum_{i=1}^n a_i \sin(a_i) + 0.1 a_i $	[0,0]
13	Schwefel 2.20	$f(a) = \sum_{i=1}^n a_i $	[0,0]
14	Salmon	$f(a) = f(a_1, \dots, a_n) = 1 - \cos(2\pi \sqrt{\sum_{i=1}^n a_i^2}) + 0.1 \sqrt{\sum_{i=1}^n a_i^2}$	[0,0]
15	Qing	$f(a) = f(a_1, \dots, a_n) = \sum_{i=1}^n (a^2 - i)^2$	[±√I, ±√i]
16	Schwefel 2.21	$f(a) = \max_{i=1, \dots, n} a_i $	[0,0]
17	Zakharov	$f(a) = f(a_1, \dots, a_n) = \sum_{i=1}^n a_i^2 + (\sum_{i=1}^n 0.5i a_i)^2 + (\sum_{i=1}^n 0.5i a_i)^4$	[0,0]
18	Brown	$f(a) = \sum_{i=1}^{n-1} (a_i^2)^{(a_{i+1}+1)} + (a_{i+1}^2)^{(a_{i+1}+1)}$	[0,0]
19	Happy Cat	$f(a) = [(\ a\ ^2 - n)^2]^\alpha + 1/n(\frac{1}{2}\ a\ ^2 + \sum_{i=1}^n a_i) + \frac{1}{2}$	[-1, -1]
20	Ackley N. 3	$f(a,b) = -200e^{-0.2\sqrt{a^2+b^2}} + 5e^{\cos(3a) + \sin(3b)}$	[±0.682, -0.360]

Table 3: Represents the comparison on the mean square approximate errors obtained in function

S.NO	Test Functions	GSOPSO-ANN	GSOGWO-ANN
1	Rastrigin	4.8E-04	0
2	Bohachevsky N.1	1.2E-04	0
3	Griewank	1.0E-04	0
4	Ackley N. 2	0	0
5	Styblinski tank	4.2E-03	0
6	Schwefel 2.22	5.3E-04	2.6E-04
7	Shubert 3	2.5E-04	2.1E-04
8	Rosenbrock	4.5E-05	8.1E-04
9	Beale	6.2E-05	2.9E-04
10	Three hump Camel	3.1E-04	0
11	Sphere	0	0
12	Alpine N .1	2.0E-04	1.2E-04
13	Schwefel 2.20	0	0
14	Salmon	3.1E-04	0
15	Qing	9.8E-04	5.1E-04
16	Schwefel 2.21	2.3E-04	1.8E-04
17	Zakharov	2.9E-04	1.4E-04
18	Brown	0	0
19	Happy Cat	2.7E-05	3.7E-04
20	Ackley N. 3	6.3E-04	4.0E-04

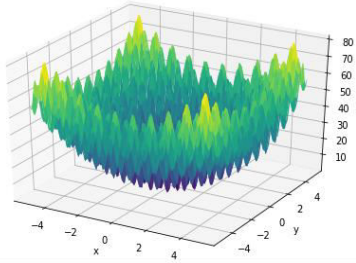


Figure 1:
Plotting of Rastrigin function

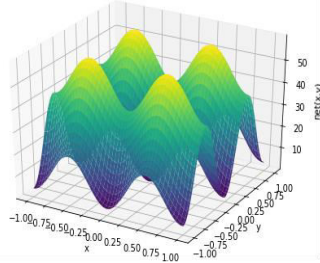


Figure 2:
Plotting of the Rastrigin function trained on GSOPSO-ANN

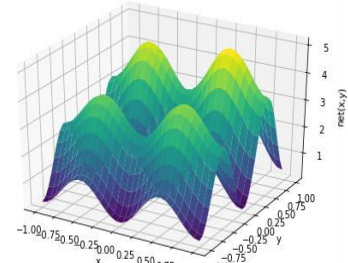


Figure 3:
Plotting of the Rastrigin function trained on GSOGWO-ANN

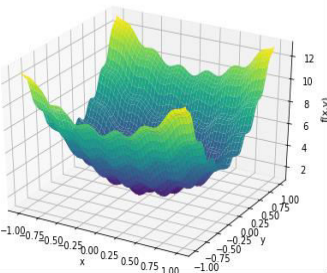


Figure 4:
Plotting of Bohachevsky N. 1 function

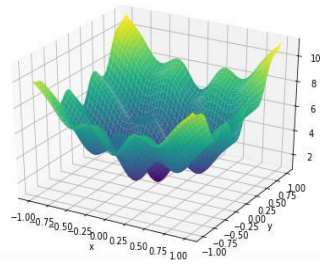


Figure 5:
Plotting of the Bohachevsky N. 1 function trained on GSOPSO-ANN

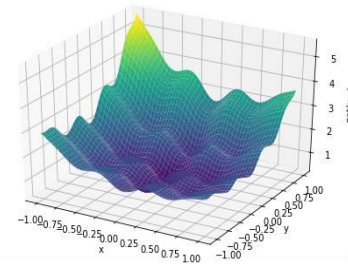


Figure 6:
Plotting of the Bohachevsky N. 1 function trained on GSOGWO-ANN

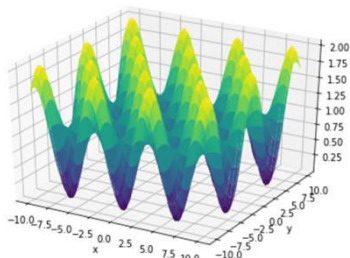


Figure 7:
Plotting of Griewank function

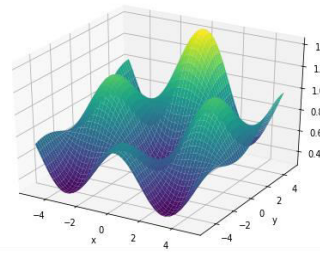


Figure 8:
Plotting of the Griewankfunction trained on GSOPSO-ANN

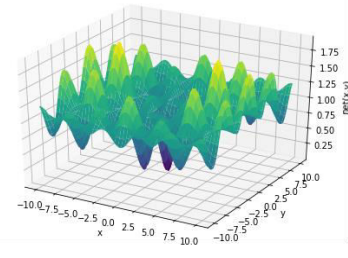


Figure 9:
Plotting of the Griewankfunction trained on GSOGWO-ANN

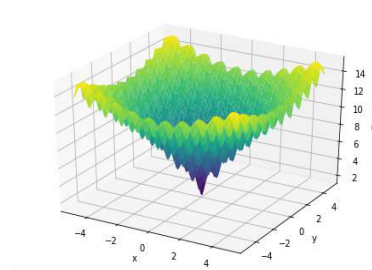


Figure 10:
Plotting of Ackley N. 2
function

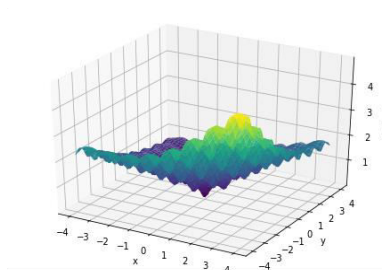


Figure 11:
Plotting of the Ackley N.
2function trained
on GSOPSO-ANN

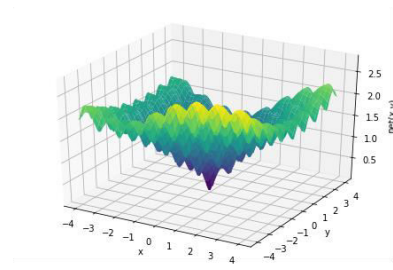


Figure 12:
Plotting of the Ackley N.
2function trained
on GSOGWO-ANN

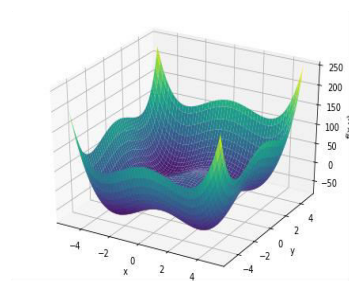


Figure 13:
Plotting of Styblinski-Tank
function

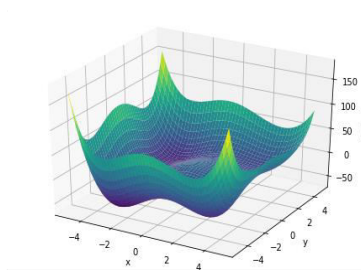


Figure 14:
Plotting of the
Styblinski-Tank function
trained on
GSOPSO-ANN

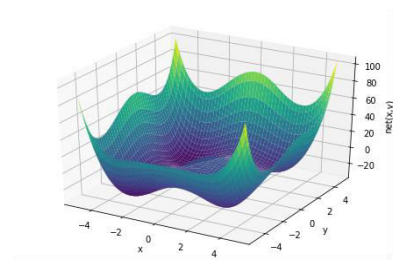


Figure 15:
Plotting of the
Styblinski-Tankfunction
trained on
GSOGWO-ANN

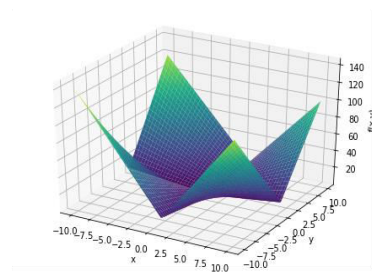


Figure 16:
Plotting of Schwefel 2.22
function

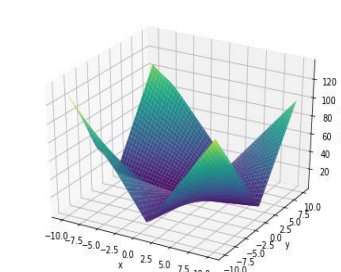


Figure 17:
Plotting of the Schwefel 2.22
function trained
on GSOPSO-ANN

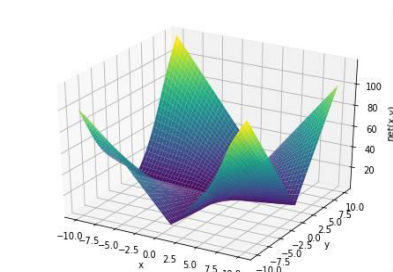


Figure 18:
Plotting of the Schwefel 2.22
function trained
on GSOGWO-ANN

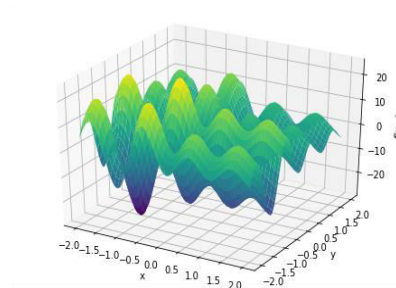


Figure 19:
Plotting of Shubert 3
function.

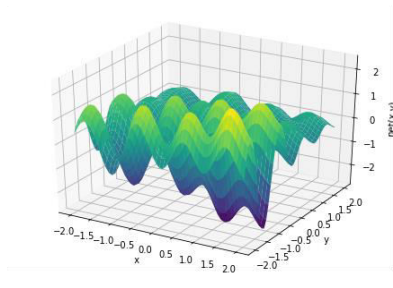


Figure 20:
Plotting of the Shubert 3
function trained
on GSOPSO-ANN

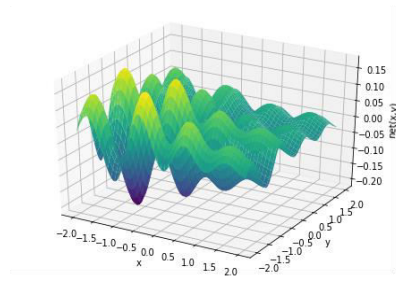


Figure 21:
Plotting of the Shubert 3
function trained
on GSOGWO-ANN

6. Conclusion

This paper proposes a new Galactic Swarm Optimization (GSO) algorithm enhanced with Grey Wolf Optimizer (GWO). The proposed algorithm was used to train a feedforward Neural Network for function approximation. The algorithm's performance was tested on 20 benchmark functions. Mean square error was considered as the performance measure. The algorithm outperformed the classical GSO on all the functions. It provided significantly better solution than the classical GSO thereby proving its explorative and exploitative abilities. Parallelizing the proposed hybrid algorithm on multi-core architecture and GPU will be considered for future work.

References

1. ArockiaPanimalar, S. (2017). Nature Inspired Metaheuristic Algorithms, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 04, p-ISSN: 2395-0072.
2. Anupam Biswas, Mishra, K. K., Shailesh Tiwari, & Misra, A. K. (2013). Physics-Inspired Optimization Algorithms. Hindawi Publishing Corporation Journal of Optimization, Article ID 438152, <http://dx.doi.org/10.1155/2013/438152>.
3. Bagchi C., Geraldine Bessie Amali D., Dinakaran M. (2019). Accurate Facial Ethnicity Classification Using Artificial Neural Networks Trained with Galactic Swarm Optimization Algorithm. Advances in Intelligent Systems and Computing, vol 862. Springer, https://doi.org/10.1007/978-981-13-3329-3_12.
4. Kaya, E., Uymaz, S.A. & Kocer, B. (2019). Boosting galactic swarm optimization with ABC. *Int. J. Mach. Learn. & Cyber.* **10**, 2401–2419, <https://doi.org/10.1007/s13042-018-0878-6>.
5. Zhang, Y., Jin, Z., & Chen, Y. (2020). Hybridizing grey wolf optimization with neural network algorithm for global numerical optimization problems. *Neural Comput & Applic* **32**, 10451–10470, <https://doi.org/10.1007/s00521-019-04580-4>.
6. Geraldine Bessie Amali, D., Dinakaran, M. (2018). A new quantum tunneling particle swarm optimization algorithm for training feedforward neural networks. *Int. J. Intell. Syst. Appl. (IJISA)* **10**(11), 64–75. <https://doi.org/10.5815/ijisa.2018.11.07>.
7. Hardi M. Mohammed, Shahla U. Umar, Tarik A. Rashid. (2019). A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm. *Computational Intelligence and Neuroscience*, Article ID 8718571, <https://doi.org/10.1155/2019/8718571>.

8. Bhardwaj, Shubham & Amali, Geraldine & Phadke, Amrut & Umadevi, K.S., & Balakrishnan, Ponnuram. (2020). A new parallel galactic swarm optimization algorithm for training artificial neural networks. *Journal of Intelligent & Fuzzy Systems*. 38. 1-11. 10.3233/JIFS-179747.
9. Gabriel Villarrubia, Juan, F., De Paz, Pablo Chamoso, Fernando De la Prieta. (2018). Artificial neural networks used in optimization problems. *Neurocomputing*, Volume 272, ISSN 0925-2312, <https://doi.org/10.1016/j.neucom.2017.04.075>.
10. Fadi N. Sibai, Hafsa I. Hosani, Raja M. Naqbi, Salima Dhanhani, Shaikha Shehhi. (2011). Iris recognition using artificial neural networks. *Expert Systems with Applications*, Volume 38, Issue 5, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2010.11.029>.
11. Geraldine Bessie Amali, D., Dinakaran, M. (2017). A review of heuristic global optimization based artificial neural network training approaches. *IAES International Journal of Artificial Intelligence (IJ-AI)* Vol. 6, No. 1, pp. 26~32, ISSN: 2252-8938, DOI: 10.11591/ijai.v6.i1.pp26-32.
12. Venkataraman Muthiah-Nakarajan, Mathew Mithra Noel. (2016). Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. *Applied Soft Computing*, Volume 38, ISSN 1568-4946, <https://doi.org/10.1016/j.asoc.2015.10.034>.
13. Ersin Kaya, Halife Kodaz and İsmail Babaoğlu. (2017). Galactic Swarm Optimization using Artificial Bee Colony Algorithm. *Fifteenth International Conference on ICT and Knowledge Engineering*, 978-1-5386-2117-2/17/\$31.00 ©2017 IEEE, <http://dx.doi.org/10.1109/ICTKE.2017.8259616>.
14. Kaya, E., Umaz, O., Korkmaz, S., İramkaya, Kiran, M.S. (2018). Performance analysis of Galactic Swarm Optimization with Tree Seed Algorithm. *International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES'18)*, 2018 Safranbolu, Turkey.
15. Mathew M. Noel. (2012). A new gradient based particle swarm optimization algorithm for accurate computation of global minimum. *Applied Soft Computing*, Volume 12, ISSN 1568-4946, <https://doi.org/10.1016/j.asoc.2011.08.037>.
16. Nguyen, B. M., Tran, T., Nguyen, T., & Nguyen, G. (2020). Hybridization of Galactic Swarm and Evolutionary Whale Optimization for Global Search Problem. *IEEE Access*, vol. 8, pp. 74991-75010, DOI: 10.1109/ACCESS.2020.2988717.
17. Chen, Shih-Chieh & Lin, Shih-Wei & Tseng, Thomas & Lin, H.-C. (2006). Optimization of Back-Propagation Network Using Simulated Annealing Approach. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. 4. 2819 - 2824. 10.1109/ICSMC.2006.385301.
18. Xin, Bin & Wang, Yipeng & Chen, Lu & Cai, Tao & Chen, Wenjie. (2017). A Review on Hybridization of Particle Swarm Optimization with Artificial Bee Colony. 242-249. 10.1007/978-3-319-61833-3_25.
19. J. Kennedy and R. Eberhart. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, pp. 1942-1948 vol.4, DOI: 10.1109/ICNN.1995.488968.
20. Mirjalili, S. (2015). How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *ApplIntell* **43**, 150–161. <https://doi.org/10.1007/s10489-014-0645-7>.
21. Slowik, A., Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Comput & Applic* **32**, 12363–12379. <https://doi.org/10.1007/s00521-020-04832-8>.
22. Bernal, E., Castillo, O., Soria, J. (2020). Fuzzy Galactic Swarm Optimization with Dynamic Adjustment of Parameters Based on Fuzzy Logic. *SN COMPUT. SCI*. **1**, 59 <https://doi.org/10.1007/s42979-020-0062-4>.
23. Narinder Singh, Singh, S. B. (2017). Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance. *Journal of Applied Mathematics*, vol. 2017, ArticleID 2030489, <https://doi.org/10.1155/2017/2030>.