Efficacy of GWO Algorithm by Varying One Algorithm-Specific Parameter

[1] P. D. Dewangan, [2] N. Patnana, [3] Lalbihari Barik, [4] P. J. Krishna, [5] V. P. Meena, and [6] V. P. Singh

[1]*Dept. of Electrical Engineering NIT Raipur, Raipur, India,* pddewangan.ele@nitrr.ac.in

[2]*Dept. of Electrical, Electronics and Comm. Engg., GITAM University, Vizag, India,* npatnana@gitam.edu

[3]*Dept. of Information Systems, King Abdulaziz University, Kingdom of Saudi Arabia,* lalbihari@gmail.com

[4]*Department of Electrical Engineering, MNIT Jaipur, Jaipur, India,* puthi.jayakrishna@yahoo.com

[5]*Department of Electrical Engineering, MNIT Jaipur, Jaipur, India,* vmeena1@ee.iitr.ac.in

[6]*Department of Electrical Engineering, MNIT Jaipur, Jaipur, India,* vinay.ee@mnit.ac.in

Abstract.

This paper compares the performance of the grey wolf optimization (GWO) algorithm for four distinct exploration approaches. These exploration strategies are used throughout the iterative procedure, with the co-efficient vector being modified for different dimensions of each solution. The co-efficient vector, $\overline{X}$ is linearly changed in the first exploration strategy from 1 to 0. While, in second, third, and fourth strategies, the co-efficient vector, $\overline{X}$ is varied linearly from 2 to 0, 10 to 0, and 20 to 0 respectively. Thus, modified GWO algorithm is applied on five different unimodel benchmarks functions for performance comparison. The performance comparison is done on the basis of results obtained for the value of objective function (i.e. figure of merit), standard deviation, mean, minimum and maximum values of the figure of merit.

Keywords. Grey Wolf Optimization, Exploration; Meta-heuristics, Swarm Intelligence, Unimodal Benchmark Functions.

1. INTRODUCTION

Over the last few decades, meta-heuristic optimization approaches have been increasingly popular. These optimization techniques are used in a wide range of research fields as well as in a large number of applications. This is because of advantages such as simplicity, flexibility, avoiding local optima, no requirement of derivation. Meta-heuristics optimization techniques are easy to learn and apply to the existing optimization problems. These optimization techniques are implemented by numerous scholars because of ease of use in replicating many natural behaviours. Also, new meta-heuristics by merging two or more meta-heuristics or strengthening the current meta-heuristics with an improvement are suggested in literature. Furthermore, the simplicity of the meta-heuristics helps researchers from inter-disciplinary branches in quickly understanding and applying them to their own optimization problems.

The next most important advantage of using these meta-heuristics to optimize the problems is their flexibility. Meta-heuristics are easily adaptable to wide range of issues as they presume the problems to be black boxes. In simple context, meta-heuristics only examine the system's input(s) and output(s). A designer just has to know how to articulate his or her problem by understanding these meta-heuristics.

The vast majority of meta-heuristics include procedures that do not need derivation. Meta-heuristics, in contrast to gradient-based optimization methods, approach issues in a stochastic way. The optimization process starts with random solution(s) to find the best, and there is no need to compute the derivative of such search spaces. As a result, meta-heuristics are well-suited for real-world scenarios where derivative information is not known.

Meta-heuristics are better than conventional optimization techniques in avoiding local optima. This is because meta-heuristics are stochastic in nature. This stochastic nature allows them to avoid trappings in local solution and thoroughly search the whole search space.

Swarm intelligence (SI) is one of the most important branch of population-based meta-heuristics. Beni and Wang [1] proposed SI for the first time in 1993. Natural colonies are a primary source of SI based methods [2]. These SI optimization algorithms generally emulate the social behaviour of swarms, herds, flocks, or schools of organisms in nature. SI optimization techniques provide a number of benefits, including:

- These optimization techniques are simple to implement.
- These optimization methods contain fewer operators than evolutionary strategies.
- There are usually fewer parameters to configure with these optimization techniques.
- These optimization methods maintain track of information about the search space during the course of iterations.
- These optimization methods usually require memory to preserve the best solution identified.

Along withstanding the benefits indicated above, SI optimization strategies have certain disadvantages too. The search agent activity seems to be noisy because the decision action is stochastic. Without understanding how the search agent works, it is impossible to know the functions of colony. Anticipating behaviour based on a set of specified rules is tough. Even little modifications to the basic rules have a significant influence on group behaviour.

Within the due course of time, many other SI optimization techniques are proposed in the literature providing good results. Some of them are particle swarm optimization [3], [4], grey wolf optimization [5], [6], cat and mouse based optimizer [7], elephant herding optimization [8], [9], jaya algorithm [10], [11], rock hyraxes swarm optimization [12], teacher-learner-based optimization [13], symbiotic organisms search optimization [14], [15], differential evolution algorithm [16], honey badger algorithm [17], sine cosine algorithm [18], [19], whale optimization algorithm [20], etc. These algorithms have also been modified and published in the literature. Furthermore, hybridising two or more algorithms improves the performance of these algorithms.

In this article, the efficacy of grey wolf optimization (GWO) algorithm is investigated by varying one algorithm-specific parameter. This algorithm-specific parameter is required in exploration. In parent GWO algorithm [5], this algorithm is varied linearly from 2 to zero as the number of iterations increase. However, in this article, three other ranges are taken for this parameter. Then, efficacy of GWO algorithm is tested for four variations (one parent and other three suggested in this article). The performance is tested for five benchmarks functions.

The organization of this contribution is as follows: In section II, GWO is described in detail and modification in GWO algorithm is proposed. Section III deals with the simulation results and discussion. Finally, the whole work is concluded in section IV.

2. Grey Wolf Optimization Algorithm

Grey wolf optimization (GWO) is a revolutionary swarm intelligence system based on grey wolf leadership hierarchy and prey hunting. Mirjalili et al. [21] produced GWO in 2014. Canis lupus is the scientific name for grey wolves, and they belong to the canidae family. These grey wolves are top-tier predators in the food chain. Grey wolves like to live in packs, which have an average of 5 to 12 members.

2.1. Description of GWO algorithm

The pack's social order, surrounding prey, hunting, attacking prey, and looking for prey are all examples of grey wolves' behaviour. These grey wolves in the pack have an extremely strict social hierarchy. Grey wolves are divided into four groups depending on social hierarchical dominance α, β, γ, and ω. The leader is at the α level and is in charge of making decisions for the pack. The decisions taken by α are dictated to the other wolves in the pack, and the entire pack must obey them. The wolf in this α level is the greatest at sustaining pack discipline and structure, demonstrating that group organisation is more essential than collective strength. The next level in the grey wolf social hierarchy is β, and the wolves in this level are in charge of aiding the α level in making decisions and other collective interests. In the absence of a α level, the wolves in the β level function as the leader.

The next level in the grey wolf social hierarchy is gamma, which includes scouts, sentinels, hunters, and caretakers. Scouts are in responsible of keeping an eye on the territory's boundaries and informing the pack if any threat persists, while sentinels are in charge of defending and ensuring the pack's safety. Hunters help α and β level wolves by hunting animals and providing food for the pack. The caretakers must look after the pack's weak, injured, and wounded wolves. The pack's lowest level is ω. At all times, the wolves at this rank must surrender to all other dominant wolves. The next most important social behaviour among grey wolves is that they always engage in groups while hunting. This encircling behaviour of grey wolves is given by

$$\overline{C} = \left|\overline{Y} \cdot \overline{J}_p - \overline{J}\right| \tag{1}$$

$$\overline{J}i + 1 = \overline{J}p - \overline{X} \cdot \overline{C} \tag{2}$$

where $\overline{J}p$ represents the position vector of prey, $\overline{J}$ indicates position vector of wolves in current iteration, $\overline{J}i + 1$ illustrates the position vector of wolves in next iteration, $\overline{X}$ and $\overline{Y}$ are co-efficient vectors which are calculated as

$$\overline{X} = 2\overline{x} \cdot \overline{r}_1 - \overline{x} \tag{3}$$

$$\overline{Y} = 2 \cdot \overline{r}_2 \tag{4}$$

where, $\overline{X}$ decreases linearly from 2 to 0 as the number of iterations increase, $\overline{r}_1$ and $\overline{r}_2$ are random vectors in the range of [0,1].

Grey wolves have the ability to track down and encircle their prey. In most cases, the α level is in charge of hunting. Hunting is a skill that β and δ level wolves can exhibit occasionally. However, in search space, we are unable to determine where the best prey position is located. We suppose that the α level is the best candidate, although the $beta$ and $delta$ levels have a better understanding of the prey's likely position. We preserve the top three best solutions identified so far and need additional search components to update their positions in line with the best search component. This behavioural aspect is considered in order to mathematically simulate the hunting behaviour of grey wolves. The following formulae are stated in this regard:

$$\bar{J}i + 1 = \frac{\bar{J}\alpha + \bar{J}\beta + \bar{J}\gamma}{3} \quad (5)$$

$$\bar{J}\alpha = \bar{J}\alpha, i - \bar{X}1 \cdot \bar{C}\alpha \quad (6)$$

$$\bar{J}\beta = \bar{J}\beta, i - \bar{X}2 \cdot \bar{C}\beta \quad (7)$$

$$\bar{J}\gamma = \bar{J}\gamma, i - \bar{X}3 \cdot \bar{C}\gamma \quad (8)$$

$$\bar{C}\alpha = \left|\bar{Y}_1 \cdot \bar{J}\alpha, i - \bar{J}\right| \quad (9)$$

$$\bar{C}\beta = \left|\bar{Y}_2 \cdot \bar{J}\beta, i - \bar{J}\right| \quad (10)$$

$$\bar{C}\gamma = \left|\bar{Y}_3 \cdot \bar{J}\gamma, i - \bar{J}\right| \quad (11)$$

where, $\bar{J}i + 1$ interprets position of wolves in next iteration, $\bar{J}\alpha$, $\bar{J}\beta$ and $\bar{J}\gamma$ represent new position of α level, β level and γ level wolves respectively; $\bar{J}\alpha, i$, $\bar{J}\beta, i$ and $\bar{J}\gamma, i$ indicate the current position of α level, β level and γ level wolves respectively; $\bar{C}\alpha$, $\bar{C}\beta$ and $\bar{C}\gamma$ refer to the encircling behaviour of α level, β level and γ level wolves respectively.

2.2. Amendment to GWO algorithm

In parent GWO algorithm, the co-efficient vector, $\bar{X}$ is presumed to be decreasing linearly from 2 to 0 as the iterations increase in number. The modification in the value of this co-efficient vector, $\bar{X}$ tends to modify the searching capability of the GWO algorithm. In this contribution, the efficacy of GWO algorithm is tested by modifying the exploration of algorithm. The exploration is modified by varying the value of $\bar{X}$ considering the following four cases.

1. By varying the value of $\bar{X}$ from 1 to 0 linearly.
2. By varying the value of $\bar{X}$ from 2 to 0 linearly.
3. By varying the value of $\bar{X}$ from 10 to 0 linearly.
4. By varying the value of $\bar{X}$ from 20 to 0 linearly.

3. RESULT AND DISCUSSION

Four different exploration strategies are proposed in previous section and are used with GWO algorithm one by one. For presenting the results, the modified GWO is used for minimizing following five benchmarks functions.

A. Sphere Function

$$F_1(x) = \sum_{i=1}^{n} x_i^2 \quad (12)$$

B. Schwefel 2.22 Function

$$F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \quad (13)$$

C. Schwefel 1.2 Function

$$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2 \quad (14)$$

D. Schwefel 2.21 Function

$$F_4(x) = max\{|x_i|, 1 \le i \le n\} \quad (15)$$

E. Rosenbrock Function

$$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (16)$$

The following four cases are considered for testing efficacy of the GWO algorithm:

1. Case 1: When $\overline{X}$ is varied from 1 to 0 linearly.
2. Case 2: When $\overline{X}$ is varied from 2 to 0 linearly.
3. Case 3: When $\overline{X}$ is varied from 10 to 0 linearly.
4. Case 4: When $\overline{X}$ is varied from 20 to 0 linearly.

A total of 100 solutions are explored while minimizing the five benchmark functions considered. The number of iterations, on the other hand, is taken to be 100. The algorithm is repeated five times in a row. The statistical analysis is based on the outcomes of five successive runs.

Table 1: Figure of merit (FOM) for dimension of $D = 10$

$D = 10$				
	Case-1	Case-2	Case-3	Case-4
F_1	**3.6954e-113**	5.6107e-97	7.7927	806.4701
	2.3269e-114	5.1516e-20	9.3672	581.2886
	1.4317e-115	4.5816e-95	13.9832	230.8661
	3.8480e-115	2.1863e-98	24.0044	194.6897
	8.2018e-115	5.6902e-97	6.1811	360.3082
F_2	**2.4177e-59**	6.0033e-50	1131.5887	6.3369e+6
	3.6649e-59	3.3865e-49	49.2205	3405.4965
	1.4559e-58	9.3874e-51	34.8568	1.1747e+7
	8.9506e-59	1.5391e-50	23.8319	1.9239e+5
	7.1150e-59	9.4559e-50	36.6730	1.9504e+5
F_3	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
F_4	**1.4726e-55**	4.0766e-47	5.2664	19.7313
	4.2422e-55	3.5396e-46	5.4806	25.6629
	3.3699e-54	4.6897e-48	5.6047	0.8569
	5.9530e-56	1.0689e-46	9.0666	0.6890
	5.7683e-55	6.4228e-46	6.1606	25.2770
F_5	8.7541	**8.1003**	1.1269e+5	6.7109e+7
	8.5925	**8.1014**	4.7527e+4	2.2580e+7
	8.7685	**8.7292**	8566.5078	1.6725e+7
	8.0738	8.1002	8.1725e+4	4.2854e+6
	8.7003	**8.0723**	5.4835e+5	5.0827e+6

Table 2: Figure of merit (FOM) for dimension of $D = 20$

$D = 20$				
	Case-1	Case-2	Case-3	Case-4
F_1	**1.1333e-103**	1.1491e-86	2019.8435	11011.4692
	1.9705e-104	2.3098e-87	2024.3171	7629.1472
	5.9303e-105	2.6469e-88	1523.5633	8539.1056
	3.1142e-104	2.6261e-89	1977.4169	8483.1347
	1.1380e-103	6.9748e-87	2403.7282	38627.4326
F_2	**8.2755e-54**	3.8488e-45	1.0319e+14	6.9821e+16
	2.3608e-53	1.1079e-45	6.9370e+13	3.9841e+16
	2.5641e-53	1.6536e-45	2.2913e+12	5.5195e+15
	4.6818e-53	1.7741e-46	2.3862e+9	1.6333e+21
	2.6256e-53	1.8730e-45	3.8336e+12	3.4570e+17
F_3	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
F_4	**2.2649e-51**	3.9997e-42	0.3277	12.6963
	8.6926e-50	2.1907e-42	0.1535	7.3062
	4.4330e-51	5.0165e-41	25.1857	53.4712
	1.8159e-50	3.9018e-42	19.6380	2.9966
	4.4769e-51	3.1600e-41	36.0258	10.7821
F_5	18.9093	**18.0925**	1.3002e+8	5.6713e+8
	18.8884	**18.6166**	5.1006e+7	9.5274e+8
	18.6499	**18.0848**	2.3117e+7	3.7050e+8
	18.8924	**18.6972**	8.4099e+8	8.0334e+8
	18.6815	**18.0722**	2.5234e+7	1.5402e+9

In this contribution, all benchmark functions are minimized with dimension of $D = 10$ and $D = 20$. The values of objective functions or figure of merit of five benchmark functions are represented in Table 1 for each of the four scenarios independently for dimension $D = 10$. Similarly, Table 2 presents the figure of merits of five benchmark functions for dimension $D = 20$ when all four cases are considered independently.

The statistical analysis of all four cases evaluated for dimension $D = 10$ is shown in Table 3. The values of standard deviation (SD), mean, minimum, and maximum of figure of merit (FOM) are tabulated in this table for each case. Table 4 shows similar findings for all four cases when the dimension is set to $D = 20$.

Table 3: Statistical Analysis for dimension of $D = 10$

		Case-1	Case-2	Case-3	Case-4
		$D = 10$			
F1	SD	**1.6137e-113**	2.3038e-20	7.1794	257.0936
	Mean	**8.1258e-114**	1.0303e-20	12.2657	434.7245
	Min	**1.4317e-115**	2.1863e-98	6.1811	194.6897
	Max	3.6954e-113	**5.1516e-20**	24.0044	806.4701
F2	SD	**4.8086e-59**	1.3591e-49	489.9798	5.2434e+6
	Mean	**7.3414e-59**	1.0360e-49	255.2342	3.6951e+6
	Min	**2.4177e-59**	9.3874e-51	23.8319	3405.4965
	Max	1.4559e-58	**3.3865e-49**	1131.5887	1.1747e+7
F3	SD	0	0	0	0
	Mean	0	0	0	0
	Min	0	0	0	0
	Max	0	0	0	0
F4	SD	**1.3877e-54**	2.6789e-46	1.5728	12.6982
	Mean	**9.1555e-55**	2.2972e-46	6.3158	14.4434
	Min	**5.9530e-56**	4.6897e-48	5.2664	0.6890
	Max	3.3699e-54	**6.4228e-46**	9.0666	25.6629
F5	SD	0.2901	**0.2845**	2.2066e+5	2.5769e+7
	Mean	8.5778	**8.2207**	1.5977e+5	2.3156e+7
	Min	8.0738	**8.07239**	8566.5078	4.2854e+6
	Max	**8.7685**	8.7292	5.4835e+5	6.7109e+7

Table 4: Statistical Analysis for dimension of $D = 20$

		Case-1	Case-2	Case-3	Case-4
		$D = 20$			
F1	SD	**5.2601e-104**	4.9332e-87	312.6342	1269.1422
	Mean	**5.6783e-104**	4.2134e-87	1989.7738	8858.0579
	Min	**5.9303e-105**	2.6261e-89	1523.5633	7629.1472
	Max	1.1380e-103	**1.1491e-86**	2403.7282	11011.4692
F2	SD	**1.3724e-53**	1.3521e-45	4.7683e+13	7.3042e+20
	Mean	**2.6119e-53**	1.7321e-45	3.5737e+13	3.2677e+20
	Min	**8.2755e-54**	1.7741e-46	2.3862e+9	5.5195e+15
	Max	4.6818e-53	**3.8488e-45**	1.0319e+15	1.6333e+21
F3	SD	0	0	0	0
	Mean	0	0	0	0
	Min	0	0	0	0
	Max	0	0	0	0
F4	SD	**3.6150e-50**	2.1584e-41	15.7720	20.4718
	Mean	**2.3252e-50**	1.8371e-41	16.2661	17.4505
	Min	**2.2649e-51**	2.1907e-42	0.1535	2.9966
	Max	8.6926e-50	**5.0165e-41**	36.0258	53.4712
F5	SD	**0.1272**	0.3156	4.3985e+7	4.4695e+8
	Mean	18.8043	**18.3127**	5.4074e+7	8.4680e+8
	Min	18.6499	**18.0722**	2.3117e+7	3.7050e+8
	Max	**18.9093**	18.6972	1.3002e+8	1.5402e+9

When comparing data in Tables 1-4, bold face data represents the better value than the equivalent normal face data. From Tables 1-2, it is clear that thirty-one times results are better in case 1. However, better results are obtained nine times in case 2. In cases 3 and 4, none result is better in comparision to case 1 and case 2. This proves that case 1, where $\overline{X}$ is varied from 1 to 0 linearly, is better than case 2, case 3 and case 4.

The same is also proved with the statistical analysis presented in Tables 3-4. In Tables 3-4, minimum value of FOM is obtained nineteen times in case 1. However, it is obtained only five times in case 2. Also, the maximum value of FOM is obtained only six times in case 2 in comparison to two times as obtained in case 1. So, with the results presented in 1-4, it is clear that by varying $\overline{X}$ from 1 to 0 linearly is better than varying $\overline{X}$ from 2 to 0 linearly, 10 to 0 linearly, or 20 to 0 linearly.

4. CONCLUSION

This article provides the comparison of performance of grey wolf optimization (GWO) algorithm with four different exploration strategies. This experimentation of initializing the co-efficient vector, $\overline{X}$ from different values simulate the divergence towards or away from the best solution. It is clear from results that the divergence of function is towards the best solution when the co-efficient vector, $\overline{X}$ declines linearly from 1 to 0. The performance of modified GWO algorithm with four different exploration strategies is utilized further for comparison of five unimodel benchmark functions. The results are tabulated in terms of value of objective function, standard deviation, mean, minimum and maximum values of objective function. Further, it can be concluded with results that we obtain the best solution when the co-efficient vector, $\overline{X}$ declines linearly from 1 to 0 instead of $\overline{X}$ declining linearly from 2 to 0, linearly from 10 to 0, or linearly from 20 to 0.

The study in this contribution should be extended to other optimization techniques as well. Furthermore, the same approach could be extended to different benchmark functions.

5. REFERENCES

[1] G. Beni and J. Wang, “Swarm intelligence in cellular robotic systems,” in *Robots and biological systems: towards a new bionics?*, Springer, 1993, pp. 703–712.

[2] E. Bonabeau, G. Theraulaz, M. Dorigo, G. Theraulaz, and D. de R. D. F. Marco, *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999.

[3] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, “Particle Swarm Optimization: A Comprehensive Survey,” *IEEE Access*, 2022.

[4] D. Sedighizadeh, E. Masehian, M. Sedighizadeh, and H. Akbaripour, “GEPSO: A new generalized particle swarm optimization algorithm,” *Math. Comput. Simul.*, vol. 179, pp. 194–212, 2021.

[5] N. S. Rathore, V. P. Singh, and B. Kumar, “Controller design for doha water treatment plant using grey wolf optimization,” *J. Intell. Fuzzy Syst.*, vol. 35, no. 5, pp. 5329–5336, 2018.

[6] G. Vashishtha and R. Kumar, “An amended grey wolf optimization with mutation strategy to diagnose bucket defects in Pelton wheel,” *Measurement*, vol. 187, p. 110272, 2022.

[7] M. Dehghani, Š. Hubálovskỳ, and P. Trojovskỳ, “Cat and mouse based optimizer: a new nature-inspired optimization algorithm,” *Sensors*, vol. 21, no. 15, p. 5214, 2021.

[8] S. Gupta, V. P. Singh, S. P. Singh, T. Prakash, and N. S. Rathore, “Elephant herding optimization based PID controller tuning,” *Int. J. Adv. Technol. Eng. Explor.*, vol. 3, no. 24, p. 194, 2016.

[9] G.-G. Wang, S. Deb, and L. dos S. Coelho, “Elephant herding optimization,” in *2015 3rd international symposium on computational and business intelligence (ISCBI)*, 2015, pp. 1–5.

[10] S. P. Singh, T. Prakash, V. P. Singh, and M. G. Babu, “Analytic hierarchy process based automatic generation control of multi-area interconnected power system using Jaya algorithm,” *Eng. Appl. Artif. Intell.*, vol. 60, pp. 35–44, 2017.

[11] W. Warid, H. Hizam, N. Mariun, and N. I. Abdul-Wahab, “Optimal power flow using the Jaya algorithm,” *Energies*, vol. 9, no. 9, p. 678, 2016.

[12] B. Al-Khateeb, K. Ahmed, M. Mahmood, and D.-N. Le, “Rock hyraxes swarm optimization: A new nature-inspired metaheuristic optimization algorithm,” *Comput. Mater. Contin.*, vol. 68, no. 1, pp. 643–654, 2021.

[13] V. P. Singh, T. Prakash, N. S. Rathore, D. P. Singh Chauhan, and S. P. Singh, “Multilevel thresholding with membrane computing inspired TLBO,” *Int. J. Artif. Intell. Tools*, vol. 25, no. 06, p. 1650030, 2016.

[14] N. S. Rathore, V. P. Singh, and B. D. H. Phuc, “A modified controller design based on symbiotic organisms search optimization for desalination system,” *J. Water Supply Res. Technol.-Aqua*, vol. 68, no. 5, pp. 337–345, 2019.

[15] A. E. Ezugwu and D. Prayogo, “Symbiotic organisms search algorithm: theory, recent advances and applications,” *Expert Syst. Appl.*, vol. 119, pp. 184–209, 2019.

[16] V. P. Meena, N. Kumar, R. K. Lenka, R. K. Barik, and V. P. Singh, “DE Based Reduced-Order Modeling for Interval Modeling Doha Water Treatment Plant,” in *2021 19th OITS International Conference on Information Technology (OCIT)*, 2021, pp. 428–432.

[17] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, “Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems,” *Math. Comput. Simul.*, vol. 192, pp. 84–110, 2022.

[18] V. P. Singh, “Sine cosine algorithm based reduction of higher order continuous systems,” in *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, 2017, pp. 649–653.

[19] V. P. Meena, S. Singh, M. Kandpal, R. K. Barik, and V. P. Singh, “SCA Assisted Reduced-Order Modelling of Interval Modelled Doha Water Treatment Plant,” in *2021 IEEE 2nd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC)*, 2021, pp. 1–6.

[20] N. S. Rathore and V. P. Singh, “Whale optimisation algorithm-based controller design for reverse osmosis desalination plants,” *Int. J. Intell. Eng. Inform.*, vol. 7, no. 1, pp. 77–88, 2019.

[21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.