International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8161
***Article Received:** 28 June 2023 **Revised:** 24 August 2023 **Accepted:** 07 September 2023*

An Improved Chaotic Grey Wolf Optimization Algorithm (CGWO)

B. V. Ramana[1], K. Kavitha[2], Smita Rani Sahu[3], B. Manideep[4], T. Ravi Kumar[5], Nibedan Panda[6]
[1] Dept. of IT, Aditya Institute of Technology and Management, Tekkali, Srikakulam, A.P, India-532201, ramana.bendi@gmail.com
[2] Dept. of IT, Aditya Institute of Technology and Management, Tekkali, Srikakulam, A.P, India-532201, kavitha.akki@adityatekkali.edu.in
[3] Dept. of IT, Aditya Institute of Technology and Management, Tekkali, Srikakulam, A.P, India-532201, smitharanisahu.it@adityatekkali.edu.in
[4] Dept. of CSE, Aditya Institute of Technology and Management, Tekkali, Srikakulam, A.P, India-532201, bendimanideep@gmail.com
[5] Dept. of CSE, Aditya Institute of Technology and Management, Tekkali, Srikakulam, A.P, India-532201, ravi.4u.kumar@gmail.com
[6]School of Computer Engineering KIIT, Deemed to be University, Bhubaneswar, Odisha, India-751024, nibedan.pandafcs@kiit.ac.in

Abstract— Grey Wolf Optimization (GWO) is a new type of swarm-based technique for dealing with realistic engineering design constraints and unconstrained problems in the field of metaheuristic research. Swarm-based techniques are a type of population-based algorithm inspired by nature that can produce low-cost, quick, and dependable solutions to a wider variety of complications. It is the best choice when it can achieve faster convergence by avoiding local optima trapping. This work incorporates chaos theory with the standard GWO to improve the algorithm's performance due to the ergodicity of chaos. The proposed methodology is referred to as Chaos-GWO (CGWO). The CGWO improves the search space's exploration and exploitation abilities while avoiding local optima trapping. Using different benchmark functions, five distinct chaotic map functions are examined, and the best chaotic map is considered to have great mobility and ergodicity characteristics. The results demonstrated that the best performance comes from using the suitable chaotic map function, and that CGWO can clearly outperform standard GWO.

Keywords- Swarm Intelligence, Grey Wolf Optimization, Optimization, Chaotic theory, Benchmark functions.

I. INTRODUCTION

Optimization is an effective method for making decisions and evaluating physical systems. An optimization problem in mathematics is the task of choosing the optimum solution from a set of all possible options. The goal of global optimization is to determine the maximum and least values of a single objective function. To handle such optimization challenges, a variety of optimization algorithms have been used [1-2]. Nature provides some of the most dynamic techniques to deal with certain difficult challenges. Nature inspired approaches are inspired by natural forms observing from the environment itself. Swarm Intelligence and Evolutionary Computing are two major components of nature inspired algorithms [3-4]. Through the use of optimization methodologies and models, these nature-inspired algorithms play a vital role in resolving many real-world challenges [5-6]. It's worth mentioning the No Free Lunch (NFL) theorem. This theorem logically proves that there is no single best meta-heuristic for all optimization problems [7]. In other words, a particular meta-heuristic may perform commendably on one set of problems while failing miserably on another. Clearly, the NFL stimulates this field of research, resulting in the improvement of current approaches and the introduction of new metaheuristics each year. This also inspires us to continue working on a novel meta-heuristic based on grey wolves. Exploration and exploitation are the two major steps of every meta-heuristic method. The programme seeks to study the whole search space of the issue during the exploration phase, and search agent motions are long and unpredictable. During the exploitation phase, the search agents, on the other hand, take only a few steps around the prospective solutions. Establishing a balance between exploration and exploitation is a crucial task in evolving a new meta-heuristic algorithm [8-9]. Nature-inspired metaheuristics have recently demonstrated exceptional performance in solving optimization problems and are widely used [10-11].

The GWO excels in resolving real-world issues and engineering designs with unknown search spaces. Author Mirjalili et al. demonstrated GWO's supremacy over several other familiar optimization algorithms, which is due to the GWO's fast search speed and precision, as well as its simplicity [8, 11]. Parameters A and C in GWO influence the balance between exploitation and exploration. In other words, it controls exploitation and exploration by linearly reducing the settings. It is unnecessary to preserve linear decrements in GWO, though, as a chaotic value could weaken the local optimum. The parameter affects exploration, but because it is a pure random walk, it slows convergence slower. As a result, GWO's convergence feature is highly dependent on its parameter values. To overcome the aforementioned issues, the use of chaos in this metaheuristic can be used to substitute specific algorithm-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8161
***Article Received:** 28 June 2023 **Revised:** 24 August 2023 **Accepted:** 07 September 2023*

dependent parameters [16-17]. Nonlinear systems exhibit the property of chaos. Because of chaos' ergodicity, chaotic-based map functions can aid in population diversification. To address the problem of premature convergence, chaos theory has been incorporated into several metaheuristics in state-of-the-art literature. When compared to traditional metaheuristics, chaotic-based metaheuristics outperform because they only require one or a few chaotic maps to embed within the metaheuristics. As a result, knowing how to select an appropriate chaotic map is a challenging task for improving the performance of an optimization strategy [18]. In this suggested approach we have integrated chaotic function with standard GWO and the hybrid form is termed as chaotic-based GWO (CGWO), which is based on the successful chaotic-based technique coupled with metaheuristic. The paper aims to present the CGWO algorithm, which can be used to interchange the crucial constraints of the GWO algorithm, which aids in switching GWO's local and global searching abilities. It also proposes a mathematical model of grey wolf leadership hierarchy and chasing mechanisms in nature.

II. TYPE GWO FRAMEWORK

GWO is a novel approach to swarm intelligence developed by S. Mirjalilli in 2014 [11]. This algorithm imitates grey wolves distinct hunting and prey-finding behaviours. The entire process of GWO such as their societal behaviour, chasing, surrounding, stalking, attacking, and searching for prey is presented in this section. When constructing GWO in order to quantitatively simulate the social structure of wolves, we deliberate the proper solution to be the alpha (α). As a result, the letters beta (β) and delta (δ) denote the second and third best solutions, respectively [19]. All the remaining solutions are thought to be omega (ω). The GWO algorithm's optimization process is led by α, β, and δ, and the letter X represents the current location of the wolf in the search space. During the hunt, grey wolves enclose their quarry. The encircling behaviour is represented mathematically in Eq. (1) and Eq. (2).

$$\vec{B} = \left|\vec{C} * \vec{Y}_p(t) - \vec{Y}(t)\right| \quad (1)$$

$$\vec{Y}(t+1) = \vec{Y}_p(t) - \vec{A}_i * \vec{B} \quad (2)$$

Where t represents ongoing reiteration, A and C are coefficient vectors, Y_p denotes the prey's position vector, and (Y) denotes a grey wolf's position vector [20].

$$\overrightarrow{A_\iota} = 2\vec{a} * \overrightarrow{rm_1} - \vec{a} \quad (3)$$

$$\vec{C} = 2 * \overrightarrow{rm_2} \quad (4)$$

Where $\overrightarrow{rm_1}$ and $\overrightarrow{rm_2}$ are random vectors between 0 to 1 and value of a ⃗ is decreased from 2 to 0 during the entire reiterations.

Grey wolves [20] are capable of detecting and encircling prey. The alpha usually leads the hunt. On rare occasions, the beta and delta may also join in on the hunt. We keep the first three best solutions we've discovered so far, and the other search agents (including the omegas) must adjust their locations to match the finest search agents' positions. The hunting behaviour is represented mathematically in Eq. (5) to Eq. (7).

$$B_\propto = C_1 * Y_\propto - Y, B\beta = C_2 * Y_\beta - Y, B\delta = C_3 * Y_\delta - Y \quad (5)$$

$$Y_1 = Y_\propto - A_1 * B_\propto, Y_2 = Y_\beta - A_2 * B_\beta, Y_3 = Y_\delta - A_3 * B_\delta \quad (6)$$

$$Y(t+1) = (Y_1 + Y_2 + Y_3)/3 \quad (7)$$

As previously stated, grey wolves complete the hunt process by attacking the prey when it stops moving [6]. Grey wolves mostly search as per the location of alpha, beta, and delta. They move around and look for prey, and when they find it, they work together to attack it. The C vector in GWO contains random values between 0 and 2. The random behaviour throughout optimization can be seen in GWO because of this vector C.

III. CHAOS TECHNIQUE

The chaos procedure is a mathematical technique used to analyse chaotic states of systems that are governed by deterministic principles and subject to initial constraints [12-13]. Chaos theory holds that there are patterns beneath the apparent randomness of chaotic systems. These patterns include interconnectedness, constant feedback loops, reiteration, and self-similarity. Chaos is a type of behaviour explained by Edward Lorenz that can be seen in a lot of natural systems, including things like fluid flow, heartbeat irregularities, and weather and climate [16]. The chaotic-optimization algorithm is a metaheuristic that uses chaotic literals to improve the performance of a solution [21]. Chaos exhibits properties that make it more likely to find solutions to problems than random search, which is based on probabilities. This makes chaos more efficient at finding solutions. Chaotic maps are used in these algorithms to speed up convergence and avoid local optima [22-24]. A large range of chaotic maps generated by physicians, researchers, and mathematicians is now available for use in optimization. Many chaotic maps have been used to create algorithms that can be used in real world situations. As a result, the current study utilized the most important unidimensional chaotic maps to solve CGWO, according to the literature [14]. The chaotic map function is used in the multidimensional solution searching space to substitute one-dimensional GWO parameters for decision variables. As a result, the parameters of GWO are modified using a one-dimensional chaotic map function. Table-1 lists five notable chaotic maps, where k denotes the chaotic sequence's index and zk denotes the chaotic sequence's kth number. These sample chaotic maps are used to evaluate the performance of different chaotic maps integrated to

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8161
***Article Received:** 28 June 2023 **Revised:** 24 August 2023 **Accepted:** 07 September 2023*

GWO. To enhance the capability of GWO in terms of exploitation and exploration we have hybridized five chaotic maps.

TABLE I. CHAOTIC MAP FUNCTIONS

S. No.	Map Name	Map Equation
1	Logistic map	$Z_{k+1} = az_k (1 - z_k)$, a = 4
2	Chebyshev map	$Z_{k+1} = \cos(a \cdot \cos{-1}(z_k))$, a = 4
3	Singer map	$Z_{k+1} = \mu (7.86z_k - 23.31z^2{}_k + 28.75z^3{}_k - 13.30287z^4{}_k)$
4	Sinusoidal map	$Z_{k+1} = az2\ k \sin(\pi z_k)$
5	Sine map	$Z_{k+1} = a\ 4 \sin(\pi z_k)$, $0 < a \leq 4$

IV. PROPOSED HYBRID CGWO TECHNIQUE

In this section, a chaotic and GWO algorithm named CGWO is introduced, which employs chaotic maps for initialization and parameter adjustment. As chaotic maps create chaos, which is unpredictable for a longer time and predictable for a shorter time in the feasible region, they boost the GWO algorithm's convergence rate. Because of the dynamic nature of chaotic maps [25-26] and their non-repetition properties, they are mostly used for initialising and modifying the parameters of metaheuristic algorithms to improve convergence speed and escape from local optima [15, 16]. We have incorporated a technique known as chaos into GWO in order to improve its ability to find global optima and speed up convergence. Chaos helps to improve the effectiveness of any chosen metaheuristic approach when searching for a solution in a given area. We have selected five map functions out of 12 chaos map function for use in realistic applications which can be used to add chaos to optimization methods. In our proposed work, we have cast-off five of the chaotic functions which called logistic function, Sine map, Sinusoidal map, Singer map and Chebyshev map. The first phase in this method is the stochastic initialization of a grey wolf population. After that, the algorithm selects a chaotic map to map and sets up its first chaotic number and variable [15]. There is no need to keep linear decrements in the standard GWO algorithm, and a chaotic variation of the value can mitigate the local optimum due to the non-repetition and ergodicity property of chaos [16]. Algorithm-1 depicts the pseudo code for the suggested CGWO approach for tackling optimization difficulties [24].

Algorithm-1: Chaotic Grey Wolf Optimization Algorithm [12]

1. Initialize: Set the generation counter t to 0.
2. Random Initialization: Create a population of grey wolves, where each wolf represents a potential solution to the optimization problem. Let i range from 1 to n.
3. Chaotic Map Initialization: Generate an initial value for a chaotic map, providing randomness to the algorithm.
4. Parameter Initialization: Set the values of parameters a, A, and C. These parameters are essential for controlling the exploration and exploitation behavior of the algorithm.
5. Fitness Evaluation: Calculate the fitness of each wolf in the population. The fitness function assesses the quality of each wolf's solution.
6. Elite Wolves Identification: Identify the alpha ($X\alpha$), beta ($X\beta$), and delta ($X\delta$) wolves. These are the best, second-best, and third-best solutions in the current population.
7. Optimization Loop: Execute the following steps while the generation counter t is less than the maximum number of iterations (Max iterations):
   a. Update Wolf Positions: Wolves adjust their positions based on the positions of the alpha, beta, and delta wolves, utilizing the chaotic map and the algorithm's parameters. This step guides the exploration and exploitation of the solution space.
   b. Fitness Re-evaluation: Recalculate the fitness of each wolf after the position updates.
   c. Elite Wolves Update: Check if any of the updated wolves outperform the current elite wolves ($X\alpha$, $X\beta$, $X\delta$) in terms of fitness. If so, update the elite wolves accordingly.
   d. Increment Generation Counter: Increment the generation counter t by 1 to keep track of the current iteration.
8. Convergence Check: After reaching the maximum number of iterations, or when a termination condition is met, exit the optimization loop.
9. Output: The best solution found during the algorithm's execution can be obtained from $X\alpha$, and its fitness value represents the optimized objective function value.
10. Termination: End the algorithm.

V. RESULTS

Each trial consists of 30 distinct runs on each benchmark for each variant with different map functions, with a sample size of 100. Table 3 records the acquired findings (e.g. mean and standard deviation values) for different variations of five chaotic map functions across 30 separate runs. Table 4 records the acquired findings (e.g. mean and standard deviation values) for different variations of five chaotic map functions across 30 separate runs.

A collection of commonly used constrained benchmark functions was utilised to examine the capabilities of the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8161
***Article Received:** 28 June 2023 **Revised:** 24 August 2023 **Accepted:** 07 September 2023*

suggested CGWO for resolving restricted difficulties, and it was applied to all of the maps presented in Table 1. Table 2 shows the results of applying all selected maps to all constrained benchmark functions to find the best possible map for all restricted optimization challenges. As per the observations, the chebyshev map outperformed the majority of the benchmark functions when compared to other maps, and it was thus chosen for future CGWO exploration on constrained optimization issues. All versions are tested using the following common parameter values to ensure a fair comparison.

A graphical analysis was also performed in order to evaluate the performance of the chaotic maps using the GWO method. Furthermore, in Figures 2-7, the algorithm optimization process is depicted using five chaotic maps for benchmark functions chosen at random. The representative unimodal (F1, F4), multimodal (F9, F13), and composite functions are among the benchmarks chosen (F16, F17). In these figures, the fitness curves represent the average optimum values produced by each method over 30 runs. The chebyshev map clearly outperforms the other chaotic maps, as shown in Figs. 2-7.

TABLE 2. RESULTS OBTAINED BY DIFFERENT CHAOTIC MAPS WITH BENCHMARKS FUNCTIONS

Functions	Logistic map	Chebyshev map	Singer map	Sinusoidal map	Sine map
F1	71736.667873	53771.165996	80604.643344	67752.576523	63782.108480
F2	4.587704e+42	7.140219e+39	2.435295e+43	6.321204e+43	1.167081e+45
F3	212977.958369	295957.976948	168305.494074	1.291228e+05	324643.272836
F4	91.723341	73.957567	86.993898	95.033326	90.782648
F5	3.848902e+10	2.431260e+10	4.111925e+10	2.747636e+10	3.317402e+10
F6	84601.326522	40134.173865	66405.507107	82663.420603	67886.415312
F7	5.135685e+09	2.883909e+09	4.495799e+09	5.463795e+09	4.219577e+09
F8	-1573.790104	-943.350115	-1516.202094	-1571.764150	-1573.391518
F9	77243.095566	51327.416168	72986.291353	75194.753025	66762.133352
F10	20.061096	20.758202	20.120565	20.073806	20.064314
F11	19.084936	15.855376	18.515177	18.440192	19.584989
F12	2.478543e+10	9.695379e+09	2.043805e+10	1.780219e+10	2.238963e+10
F13	3.228408e+10	1.840050e+10	2.930260e+10	2.400493e+10	2.214842e+10
F14	9.980251e-01	9.982013e-01	9.980041e-01	9.68476e+00	9.980041e-01
F15	6.386224e+00	0.301076	14.668159	5.934747	11.678166
F16	1.295598e+08	1.888475e+07	2.400625e+06	4.538428e+06	6.384610e+07
F17	920.565662	8.983515	45.744946	1590.658830	1329.419672
F18	2.076415e+12	8.435047e+06	6.191657e+10	1.211011e+12	3.573261e+07

TABLE 3. UNIMODAL BENCHMARKS FUNCTIONS

Functions	Statistics	Logistic map	Chebyshev map	Singer map	Sinusoidal map	Sine map
F1	Mean	71736.667873	53771.165996	80604.643344	67752.576523	63782.108480
	Std	3428.165680	8065.511122	1029.414844	1705.791487	1957.329675
F2	Mean	4.587704e+42	7.140219e+39	2.435295e+43	6.321204e+43	1.167081e+45
	Std	1.259103e+27	2.364588e+40	2.325706e+43	3.888580e+43	8.219882e+44
F3	Mean	212977.958369	295957.976948	168305.494074	1.291228e+05	324643.272836
	Std	2655.697233	62509.887434	698.951706	5.920274e-11	8314.199979
F4	Mean	91.723341	73.957567	86.993898	95.033326	90.782648
	Std	6.763480	6.769598	3.840994	0.273111	2.387849
F5	Mean	3.848902e+10	2.431260e+10	4.111925e+10	2.747636e+10	3.317402e+10
	Std	4.824734e+09	4.093145e+09	7.196652e+09	2.974564e+08	1.163973e-05

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8161
***Article Received:** 28 June 2023 **Revised:** 24 August 2023 **Accepted:** 07 September 2023*

F6	Mean	84601.326522	40134.173865	66405.507107	82663.420603	67886.415312
	Std	2424.603835	6596.243239	4091.231873	2264.400685	2160.745265
F7	Mean	5.135685e+09	2.883909e+09	4.495799e+09	5.463795e+09	4.219577e+09
	Std	1.352339e+08	9.305532e+08	5.572161e+08	1.758548e+08	8.246025e+07

TABLE 4. MULTIMODAL BENCHMARKS FUNCTIONS

Functions	**Statistics**	**Logistic map**	**Chebyshev map**	**Singer map**	**Sinusoidal map**	**Sine map**
F8	Mean	-1573.790104	-943.350115	-1516.202094	-1571.764150	-1573.391518
	Std	263.014657	229.910483	317.193606	261.419796	246.303269
F9	Mean	77243.095566	51327.416168	72986.291353	75194.753025	66762.133352
	Std	1365.856113	6671.304458	405.080214	4474.605806	6201.373763
F10	Mean	20.061096	20.758202	20.120565	20.073806	20.064314
	Std	0.266086	0.289605	0.358978	0.298651	0.281994
F11	Mean	19.084936	15.855376	18.515177	18.440192	19.584989
	Std	0.596490	1.137251	0.810060	1.487324	0.433995
F12	Mean	2.478543e+10	9.695379e+09	2.043805e+10	1.780219e+10	2.238963e+10
	Std	6.511196e+08	3.138676e+09	4.568832e+09	1.674272e+09	3.332056e+09
F13	Mean	3.228408e+10	1.840050e+10	2.930260e+10	2.400493e+10	2.214842e+10
	Std	3.863349e+08	5.430459e+09	1.665790e+09	8.489924e+08	1.551964e-05

TABLE 5. COMPOSITE BENCHMARKS FUNCTIONS

Functions	**Statistics**	**Logistic map**	**Chebyshev map**	**Singer map**	**Sinusoidal map**	**Sine map**
F14	Mean	9.980251e-01	9.982013e-01	9.980041e-01	9.68476e+00	9.980041e-01
	Std	4.375667e-15	1.826917e-10	1.475821e-12	4.753281e+01	1.410123e-15
F15	Mean	6.386224e+00	0.301076	14.668159	5.934747	11.678166
	Std	9.033621e-16	1.589285	26.775500	20.226733	35.516473
F16	Mean	1.295598e+08	1.888475e+07	2.400625e+06	4.538428e+06	6.384610e+07
	Std	4.546770e-08	4.850863e+07	2.902423e+06	0.000000e+00	3.031180e-08
F17	Mean	920.565662	8.983515	45.744946	1590.658830	1329.419672
	Std	605.056917	16.415241	72.292548	508.418034	0.043638
F18	Mean	2.076415e+12	8.435047e+06	6.191657e+10	1.211011e+12	3.573261e+07
	Std	2.272061e+12	2.534951e+07	2.126045e+11	3.264070e+12	1.515590e-08

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8161
***Article Received:** 28 June 2023 **Revised:** 24 August 2023 **Accepted:** 07 September 2023*

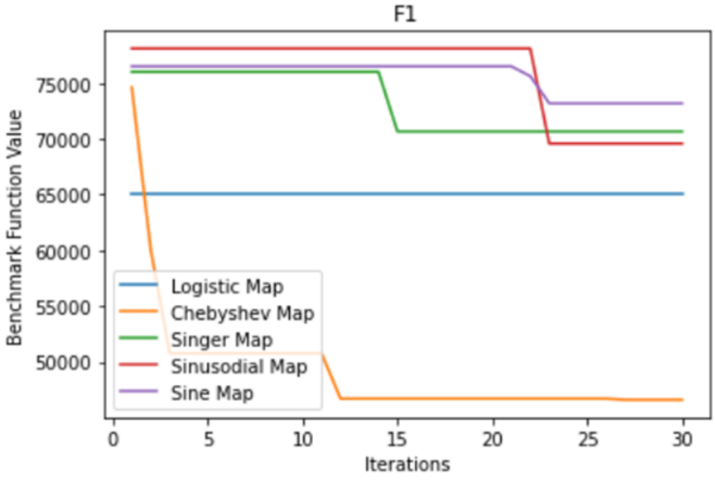

Fig. 2 Chaotic maps performance comparison on F1

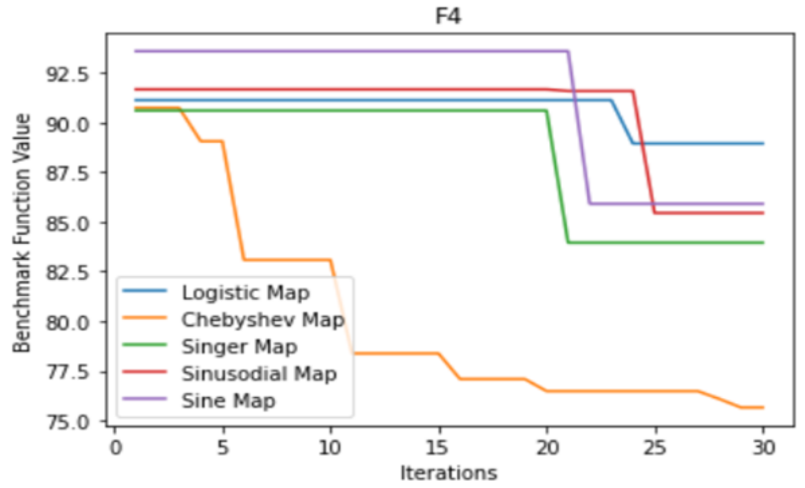

Fig. 3 Chaotic maps performance comparison on F4

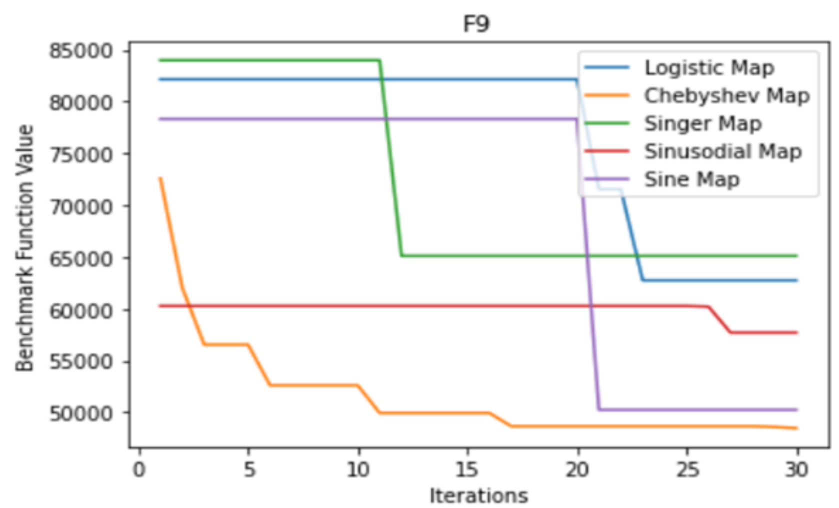

Fig. 4 Chaotic maps performance comparison on F9

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8161
***Article Received:** 28 June 2023 **Revised:** 24 August 2023 **Accepted:** 07 September 2023*

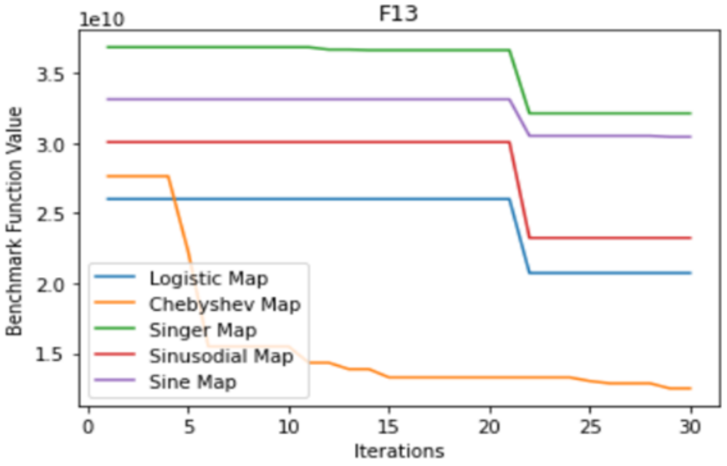

Fig. 5 Chaotic maps performance comparison on F13

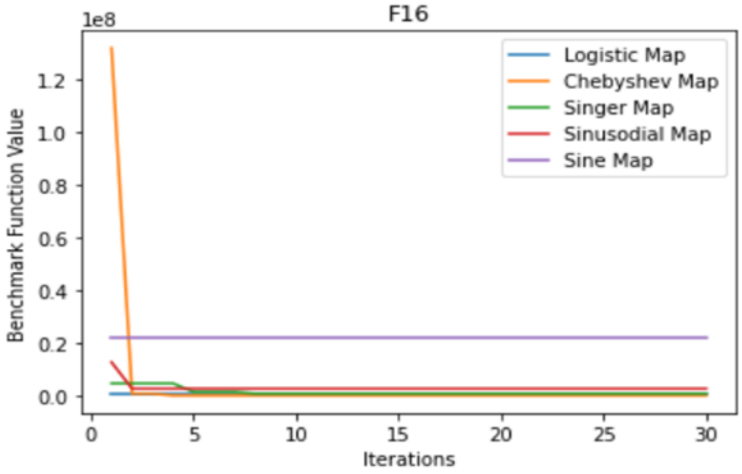

Fig. 6 Chaotic maps performance comparison on F16

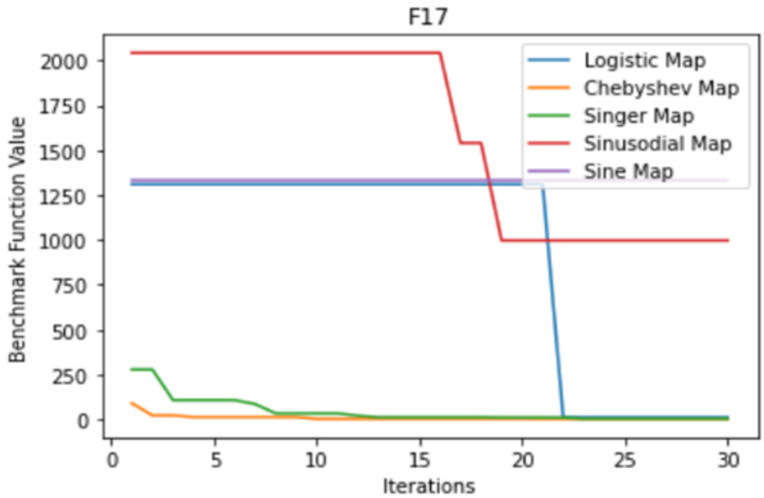

Fig. 7 Chaotic maps performance comparison on F17

VI. CONCLUSION

In this paper, by merging chaos theory and the Grey Wolf Optimizer, a unique GWO chaotic technique was provided for limited optimization problems. Due to the low solution accuracy, and easy falling into the local optimum, GWO often encounters the problem of being stuck in the local optimum and the convergence speed is slower. The proposed CGWO integrates chaos technique into the standard GWO to effectively achieve more balanced exploration and exploitation, improved population diversity, and modified convergence rate. To mitigate local optima, several chaotic map functions are used to govern the crucial parameters. The proposed CGWO has been validated through several constrained unimodal, multimodal, and composite benchmark functions. The Chebyshev map is chosen as it's by evaluating many chaotic GWO variations in order to develop the optimum CGWO. This chaos aids the controlling parameter in finding the optimal solution faster, which improves the algorithm's convergence rate. Further, the suggested approach will be utilized to resolve various realistic complications.

REFERENCES

[1] Kennedy, J., 2006. Swarm intelligence. In Handbook of nature-inspired and innovative computing (pp. 187-219). Springer, Boston, MA.

[2] Yegnanarayana, B., 2009. Artificial neural networks. PHI Learning Pvt. Ltd.

[3] Fogel, D.B., 1998. Evolutionary computation: the fossil record. Wiley-IEEE Press.

[4] Kennedy, J., 2006. Swarm intelligence. In Handbook of nature-inspired and innovative computing (pp. 187-219). Springer, Boston, MA.

[5] Wolpert, D.H. and Macready, W.G., 1997. No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), pp.67-82.

[6] Alba, E., Dorronsoro, B.: The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans. Evol. Comput. 9(2), 126–142 (2005). https://doi.org/10.1109/TEVC.2005.843751

[7] Song, H., Triguero, I., Özcan, E.: A review on the self and dual interactions between machine learning and optimisation. Progress Artifcial Intell. 8(2), 143–165 (2019). https://doi.org/10.1007/s13748-019-00185-z

[8] Mirjalili, S., Lewis, A.: The Whale optimization algorithm. Adv. Eng. Softw. 95, 51–67 (2016). https://doi.org/10.1016/j.advengsoft.2016.01.008

[9] Lu C, Li XY, Gao L, Liao W, Yi J (2017) An effective multiobjective discrete virus optimization algorithm for flexible job shop scheduling problem with controllable processing times. Comput Indust Eng 104:156–174

[10] Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. Evolut Comput IEEE Trans 13:945–958

[11] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007

[12] Kohli, M. and Arora, S., 2018. Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of computational design and engineering, 5(4), pp.458-472.

[13] He, D., He, C., Jiang, L.-G., Zhu, H.-W., & Hu, G.-R. (2001). Chaotic characteristics of a one-dimensional iterative map with infinite collapses. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 48(7), 900–906

[14] Gandomi, A. H., & Yang, X.-S. (2014). Chaotic bat algorithm. Journal of Computational Science, 5(2), 224–232.

[15] Kellert, S.H.: In the wake of chaos: Unpredictable order in dynamical systems. University of Chicago press, (1994)

[16] Yang, D., Li, G., Cheng, G.: On the efficiency of chaos optimization algorithms for global optimization. Chaos, Solitons Fractals 34(4), 1366–1375 (2007). https://doi.org/10.1016/j.chaos.2006. 04.057.

[17] Dewangan, P.D., Patnana, N., Barik, L., Krishna, P.J., Meena, V.P. and Singh, V.P., 2022, March. Performance Evaluation of GWO Algorithm Using Different Initialization Strategies. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 714-718). IEEE.

[18] Lu, C., Gao, L., Li, X., Hu, C., Yan, X. and Gong, W., 2020. Chaotic-based grey wolf optimizer for numerical and engineering optimization problems. Memetic Computing, 12(4), pp.371-398.

[19] Mosavi, M.R., Khishe, M. and Ghamgosar, A., 2016. Classification of sonar data set using neural network trained by gray wolf optimization. Neural Network World, 26(4), p.393.

[20] Escobar, H. and Cuevas, E., 2021. Implementation of Metaheuristics with Extreme Learning Machines. In Metaheuristics in Machine Learning: Theory and Applications (pp. 125-147). Springer, Cham.

[21] Wang, G.G., Deb, S., Gandomi, A.H., Zhang, Z. and Alavi, A.H., 2014, September. A novel cuckoo search with chaos theory and elitism scheme. In 2014 International Conference on Soft Computing and Machine Intelligence (pp. 64-69). IEEE.

[22] Asghari, K., Masdari, M., Gharehchopogh, F.S. and Saneifard, R., 2021. A chaotic and hybrid gray wolf-whale algorithm for solving continuous optimization problems. Progress in Artificial Intelligence, 10(3), pp.349-374.

[23] Yang, Z. and Wei, C., 2020. Prediction of equipment performance index based on improved chaotic lion swarm optimization–LSTM. Soft Computing, 24(13), pp.9441-9465.

[24] Kohli, M. and Arora, S., 2018. Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of computational design and engineering, 5(4), pp.458-472.

[25] Qin, X., Qiao, Y. and Hu, G., 2020. Degree reduction of SG-Bézier surfaces based on grey wolf optimizer. Mathematical Methods in the Applied Sciences, 43(10), pp.6416-6429.

[26] Varol Altay, E. and Alatas, B., 2020. Bird swarm algorithms with chaotic mapping. Artificial Intelligence Review, 53(2), pp.1373-1414.