Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 5, pp. 1401–1424
DOI : 10.47974/JIOS-1361

StableGWO : A grey wolf optimizer with von Neumann stability criteria

Shitu Singh ‡
Anshul Gopal †
Department of Mathematics
South Asian University
New Delhi 110068
India

Susheel Kumar Joshi §
Department of Computational Science and Humanities
Indian Institute of Information Technology
Kottayam 686635
Kerala
India

Jagdish Chand Bansal*
Department of Mathematics
South Asian University
New Delhi 110068
India

Abstract

The Grey wolf optimizer (GWO) is a recent swarm intelligence-based algorithm. The performance of GWO highly depends on the choice of the controlling parameters. Finding the most suitable values of these parameters becomes a challenging task due to the stochastic nature of the position update process. Mathematical analysis of the position update mechanism can guide in finding the most appropriate range of these parameters. This paper attempts to use von Neumann stability criteria to find the most suitable value of these parameters. The objective of this study is also to check whether the original parameter setting is in line with the recommendation of mathematical analysis. Recommendations are further examined using numerical experiments over 23 classical benchmark functions. It is

‡ *E-mail:* **singh.shitu93@gmail.com**
† *E-mail:* **anshul.gopal@gmail.com**
§ *E-mail:* **sushil4843@gmail.com**
* *E-mail:* **jcbansal@sau.ac.in (Corresponding Author)**

found that the parameters of GWO are within the stable range. It is also verified that GWO performs poorly if the parameters are set beyond the stable range.

Subject Classification: *90C31, 90C30, 90C26.*

Keywords: *Swarm intelligence, Optimization, Grey wolf optimizer, Von Neumann stability criteria, Stability analysis.*

1. Introduction

In the last few years, swarm intelligence-based algorithms proved to be effective tools to solve many real-world optimization problems. The source of inspiration for such algorithms is the collective and self-organized system that emerged among the social creatures in a swarm for foraging or predator avoidance. Some of the popular swarm intelligence-based algorithms are particle swarm optimization (PSO) [8], artificial bee colony (ABC) [19], spider monkey optimization (SMO), [5], cuckoo search optimization (CSO) [38], grey wolf optimizer (GWO) [26] and gravitational search algorithm (GSA) [29]. GWO is a well-known algorithm in this class, which is inspired by the leadership and social hierarchy behavior of the wolves. GWO has been successfully applied to solve various optimization problems, including but not limited to proportional integral derivative (PID)[24], load frequency control (LFC) [16], scheduling problem [20], feature selection [9], power dispatch problem [34], feature selection [12] etc.

Researchers have also modified the original version of the GWO algorithm in various ways to make it more efficient and accurate. In [32, 6], to improve the exploitation and exploration phases, an improved version of GWO (IGWO) is proposed. In [23], a modified variant of GWO algorithm, the exploration-enhanced GWO (EEGWO) is proposed. In EEGWO, to improve the search ability of wolves in terms of exploration and exploitation, a novel position-updated process has been proposed with the help of a random individual in the population. In addition, a nonlinear control parameter strategy has been applied in various ways to improve the exploration and exploitation phases of the GWO algorithm. In [30], a weighted grey wolf optimizer (wdGWO) has been proposed, in which a weighted average of leader wolves are computed instead of a simple arithmetic average. In [28], an improved grey wolf optimizer (I-GWO) has been proposed. In this paper, dimension learning-based hunting (DLH) is incorporated in the GWO algorithm. The DLH search strategy has increased the global search by multi-neighbor learning. The linear rank-based grey wolf optimizer (LGWO) [2] is a new variant of GWO. In LGWO, different selection methods have been chosen to find the optimal of the optimization problem. In [25], the stopping condition of GWO was tuned in an adaptive manner based on fitness improvement over the course of optimization to decrease its computational cost. In [10], a hybrid method between the variational iteration method and GWO is applied for solving nonlinear differential equations.

All the swarm intelligence-based algorithms have gone through many modifications and applications to various real-world domains. However the theoretical analysis of these algorithms is not that expensive. Few attempts have been made in this direction. In [4, 3, 14], authors have carried out the stability analysis of ABC. In this paper, von Neumann stability criterion is used and the selection of the parameters has been recommended. In [1], based on stability analysis, authors have used differential equations in place of difference equations. This approach helped to determine the range of parameters to confirm stability in pheromone trails. In [11], the stability analysis of the gravitational search algorithm (GSA) has been investigated using the Lyapunov stability theorem and utilized for adapting parameters. In [13], the stability analysis for differential evolution (DE) has been found using the von Neumann stability criterion. In this paper, the stability analysis of the GWO algorithm has been carried out using the von Neumann stability criteria for a two-level finite difference scheme. Based on the von Neumann stability criteria, the selection of parameters is suggested. The GWO algorithm with suggested parameters is tested on 23 classical benchmark functions using statistical and convergence analysis.

The remaining paper is organized as follows: section 2 describes the brief concept of the original GWO. In section 3, the basic concept of the von Neumann stability criterion and then GWO stability analysis is performed using von Neumann stability criterion. Numerical results and analysis are presented in section 4. Finally, section 5 concludes the work.

2. Grey Wolf Optimizer (GWO) algorithm

In this section, the inspiration and mathematical models of the GWO algorithm are presented.

2.1 *Inspiration*

GWO is a well-known swarm intelligence-based optimization algorithm [26]. The main inspiration behind the development of the GWO algorithm is the leadership and hunting strategy of the wolves. Grey wolves always prefer to live in a group (pack) of size 5 – 12 wolves on average. According to the hierarchy mechanism of GWO, the grey wolves are divided into four categories.

The first category wolf is alpha (α) which is a dominant wolf. The main responsibility of the α wolf is to make better decisions for all the issues in the pack. The second category wolf is the beta (β) wolf who works as a subordinate to the α wolf and makes the decisions for the entire pack when the α wolf is absence. The third category of wolves is delta (δ) wolves. These are the lowest ranking members of the pack with permission to eat at last. The α, β, and δ are known as leader wolves. The wolves which is not α, β or δ are called omega (ω) wolves. δ wolves obey orders from α and β wolves. They are the third rank wolves but higher than ω wolves. δ wolves usually take control of the pack when β and/or α wolves are busy to do so. Another important feature of the group is group hunting. Muro et al. [27] have explained grey wolves' hunting behavior in the following three steps: chasing, encircling, and attacking the prey.

2.2 *Mathematical model of GWO*

The mathematical model of the GWO is inspired by the hunting strategy of grey wolves. The hunting strategy of grey wolves includes chasing, encircling prey, and attacking the prey.

The social behavior is modeled by considering the best solution as α wolf, the second best solutions as β, and the third best as δ wolf, respectively, while the rest of the solutions are termed as ω wolves. During the search, ω wolves follow the α, β and δ wolves.

Initially, grey wolves encircle the prey. The mathematical model of encircling prey is given by the following equation:

$$X^t = X_p^t - A \times D \tag{1}$$

where, X_p^t is the position of the leader wolves at t^{th} iteration in wolf group. A and D are calculated as:

$$A = 2 \times a \times rand_1 - a \tag{2}$$

where, a is decreases linearly from 2 to 0 over the iterations and $rand_1$ is random value in the range (0,1). Clearly, A lies in the range $(-a, a)$ or we can say A is a uniformly distributed random number between $(-a, a)$. It can be defined in Figure 2.

$$a = 2 - 2 \times \left(\frac{t}{t_{max}} \right) \tag{3}$$

$$D = | C \times X_p^t - X^t | \tag{4}$$

where
$$C = 2 \times rand_2 \tag{5}$$

In equation (3), t_{max} is the maximum iterations.

In the mathematical model of hunting strategy, the prey position is approximated by the α, β and δ wolves as follows:

$$X^{(t+1)} = \frac{X_1^t + X_2^t + X_3^t}{3} \tag{6}$$

where,

$$X_1^t = X_\alpha^t - A_1 \times D_1 \tag{7}$$

$$X_2^t = X_\beta^t - A_2 \times D_2 \tag{8}$$

$$X_3^t = X_\delta^t - A_3 \times D_3 \tag{9}$$

and, A_1, A_2 and A_3 can be calculated from equation (2). D_1, D_2 and D_3 are calculated as:

$$D_1 = | C_1 \times X_\alpha^t - X^t | \tag{10}$$

$$D_2 = | C_2 \times X_\beta^t - X^t | \tag{11}$$

$$D_3 = | C_3 \times X_\delta^t - X^t | \tag{12}$$

The parameters C_1, C_2 and C_3 are lie randomly in the range (0, 2).

From the mathematical model of GWO, it is clear that the parameters A and C are the most significant parameters. They are responsible for exploration and exploitation during the search process. The flowchart of the GWO algorithm is presented in Figure 1.

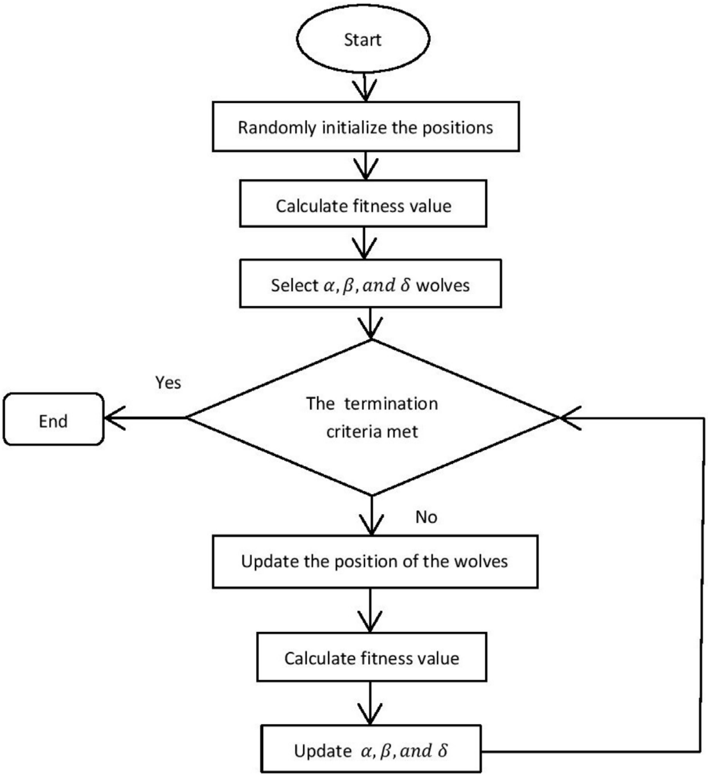

Figure 1

Flowchart of the grey wolf optimizer (GWO)

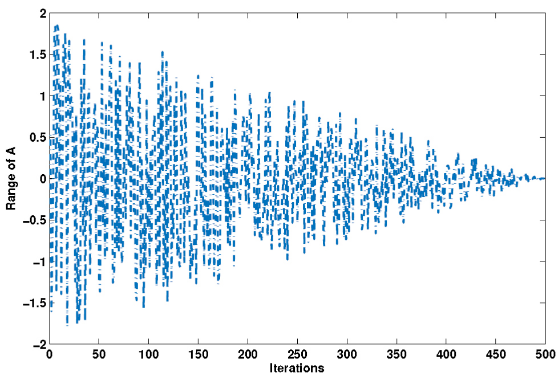

Figure 2

Decreasing behavior of the parameter A

3. Stability Analysis

Swarm Intelligence methods belong to the family of stochastic optimization algorithms. In order to get a stable solution, the error due to the algorithm should be bounded. Therefore, we should find appropriate conditions so that the error remains bounded. These conditions depend upon many parameters of the considered optimization algorithms. In the GWO algorithm, parameters A and C play important roles during the search process. Therefore, it is required to derive the conditions under which the parameters A and C lie within the *stable range*. A stable range of parameters is the one which is obtained by performing a stability analysis of the algorithm. In this paper, the von Neumann stability criteria [36] is used to find the stable range of the parameters A and C of the GWO algorithm. In the following subsection, the von Neumann stability criteria is given and applied to the GWO algorithm:

3.1 *von Neumann stability criteria*

The von Neumann stability criteria is used for stability analysis for one or multidimensional finite difference schemes. This method is based on the Fourier series solution when boundary conditions are assumed to be periodic.

Consider the following partial differential equation with linear spatial differential operator $C'_x(v)$:

$$\frac{\partial v}{\partial t} + C'_x(v) = 0 \tag{13}$$

where v is a dependent variable. The finite difference scheme corresponding to equation (13) can be written in the following form [36]:

$$\sum_{q=-\mu_l}^{\mu_r} R_q v_{j+q}^{n+1} = \sum_{q=-\eta_l}^{\eta_r} S_q v_{j+q}^{n+1} \tag{14}$$

Where, μ_l, μ_r, η_l, $\eta_r \in \mathbb{Z}^*$; $\mathbb{Z}^*$ denotes the set of non-negative numbers. j and n are mesh points in the time and space domain, respectively. In this method, each term (v_j^n) of equation (14) is replaced by the m^{th} fourier coefficient of a harmonic decomposition of v_j^n, i.e., by $w^n(m)e^{\iota(mj\Delta x)}$, where $w^n(m)$ represents m^{th} fourier coefficient at time level t and $\iota = \sqrt{-1}$.

Fourier coefficient of harmonic decomposition of v_j^n at n^{th} and $(n+1)^{th}$ are given by the following formula:

$$: w^{n+1}(m) = G(m) w^n(m) \tag{15}$$

where $G(m)$ denotes the amplification factor. The finite difference scheme (14) is stable if and only if $|G(m)| \leq 1 \ \forall m$. The scheme is unstable if $|G(m)| > 1$ for some m. If $|G(m)| = 1 \ \forall m$ finite difference scheme (14) is nondissipative or marginally stable.

In the literature, there are many methods for the stability analysis: Z-transformation [22, 21, 31], Lyapunovâ€™s stability theorem [35, 17], eigenvalue method [7, 15], Crank-Nicolson scheme [18] and von Neumann stability criteria [36]. These methods have also been used to check the stability analysis of the metaheuristic algorithms, namely differential evolution (DE), artificial bee colony (ABC), particle swarm optimization (PSO), gravitational search algorithm (GSA), Bacterial foraging optimization (BFO), etc. In the GWO algorithm, the update equation can easily be converted to a finite difference scheme which can further be analyzed for stability by von Neumann stability criteria.

The next subsection explains the stability analysis of the GWO algorithm.

3.2 *Stability Analysis of GWO algorithm*

The exploration and exploitation are two important phases for the swarm intelligence-based algorithms. In the GWO algorithm, the first half of the iterations are used for exploration and the second half of the iterations are used for exploitation. The parameters A and C are very helpful in both phases of exploration and exploitation.

In the GWO algorithm, the encircling behavior of prey is given by the following equation:

$$X^{(t+1)} = X_p^t - A \times D$$

where, X_p^t is the position of the leader wolves at t^{th} iteration, and the parameter A, and the distance vector D are given in equations (2) and (4).

From equation (4),

$$D = |(C \times X_p^t - X^t)| = \begin{cases} (C \times X_p^t - X^t), & (C \times X_p^t - X^t) > 0 \\ -(C \times X_p^t - X^t), & \text{otherwise} \end{cases}$$

Case 1: When $(C \times X_p^t - X^t) > 0$,

$$X^{(t+1)} = X_p^t - A \times (C \times X_p^t - X^t)$$

$$X^{(t+1)} = (1 - A \times C) \times X_p^t + A \times X^t$$

In the whole process, we take $B_1 = (1 - A \times C)$.

The above equation can be written in the *i* and *j* space as follows:

$$X_{i,j}^{(t+1)} = B_1 X_{p,j}^t + A \times X_{i,j}^t \tag{16}$$

where, $X_{i,j}^{(t+1)}$ and $X_{i,j}^t$ is the position of the i^{th} grey wolf of the j^{th} dimension at $(t+1)^{th}$ and t^{th} iterations, respectively. Without loss of generality, equation (16) is written as follows:

$$X_i^{(t+1)} = B_1 \times X_p^t + A \times X_i^t \tag{17}$$

where *i* denotes the index of grey wolf and without loss of generality, *p* can be written as $p = i \pm b$, b is a random integer in the range $[1, SN]$, SN is the number of grey wolves.

Then,

$$X_i^{(t+1)} = B_1 \times X_{i \pm b}^t + A \times X_i^t \tag{18}$$

Consider that the exact (optimal) solution of the optimization problem is $X = X(i,t)$ at time *t* and uniform mesh grid point *i*, and $X_{l,n} = X(i_l, t_n)$ is the approximate solution of the problem. After that equation (18) can be written in terms of node points as:

$$X_l^{(n+1)} = B_1 \times X_{l \pm b}^n + A \times X_l^n \tag{19}$$

According to the von Neumann stability criteria, in the equation (19), each term (X_l^n) is replaced by the Fourier coefficient i.e., $w^n(m)e^{\iota(ml\Delta i)}$. After putting the value of each term in equation (19), the value of the amplification factor $G(m)$ can be calculated [Appendix A] and given as:

$$|G(m)| = \sqrt{(A + B_1^2 - 4 \times A \times \sin^2\left(\frac{\theta}{2}\right) B_1} \tag{20}$$

As discussed in subsection (3.1) the condition of stability is:

$$|G(m)| \leq 1$$

or

$$\sqrt{(A + B_1)^2 - 4 \times A \times \sin^2\left(\frac{\theta}{2}\right) B_1} \leq 1$$

squaring both sides,

$$(A + B_1)^2 - 4 \times A \times \sin^2\left(\frac{\theta}{2}\right) B_1 \leq 1$$

$$(A+B_1)^2 - 1 \le 4 \times A \times \sin^2\left(\frac{\theta}{2}\right) B_1 \tag{21}$$

Where A is uniformly distributed random value between -2 and 2 i.e., $A \in rand(-2,2)$ and C is also a uniformly distributed random value in [0, 2] i.e., $C \in rand(0,2)$. The stability condition will be different for different sets of range of values of the parameters A and C. Here, the following cases need to be discussed.

Case 1(A): when $A = rand(0,2)$ and $C = rand(0,2)$ then B_1 may be positive or negative which leads the following subcases:

Subcase (a): when B_1 is positive.

$$\left[\frac{(A+B_1)^2-1}{4\times A\times B_1}\right] \le \sin^2\left(\frac{\theta}{2}\right)$$

$$\left[\frac{(A+B_1)^2-1}{4\times A\times B_1}\right] \le 1 \qquad \because \quad \sin^2\left(\frac{\theta}{2}\right) \le 1$$

$$(A-B_1)^2 - 1 \le 0$$

$$(A-B_1)^2 \le 1 \tag{22}$$

By using condition $B_1 \ge 0$ and obtained inequality (22), it can be concluded that the parameters A and C are random numbers in the range (0, 1) i.e., A, $C \in rand(0,1)$.

Subcase (b): when B_1 is negative.

$$\left[\frac{(A+B_1)^2-1}{4\times A\times B_1}\right] \ge \sin^2\left(\frac{\theta}{2}\right)$$

$$\left[\frac{(A+B_1)^2-1}{4\times A\times B_1}\right] \ge 0 \qquad \because \quad \sin^2\left(\frac{\theta}{2}\right) \ge 0$$

$$(A+B_1)^2 - 1 \le 0$$

$$(A+B_1)^2 \le 1 \tag{23}$$

By using condition $B_1 < 0$ and above inequality (23), it can be concluded that the value of A and C are random numbers in the interval (1, 2) i.e., A, $C \in rand(1,2)$.

Case 1(B): when $A = rand(-2,0)$ and $C = rand(0,2)$ then B_1 is always positive. From equation (21),

$$\left[\frac{(A+B_1)^2-1}{4\times A\times B_1}\right] \geq \sin^2\left(\frac{\theta}{2}\right)$$

$$\left[\frac{(A+B_1)^2-1}{4\times A\times B_1}\right] \geq 0 \qquad \because \quad \sin^2\left(\frac{\theta}{2}\right) \geq 0$$

$$(A+B_1)^2 - 1 \leq 0$$

$$(A+B_1)^2 \leq 1 \tag{24}$$

By using condition $B_1 > 0$ and obtained inequality (24), it can be concluded that A takes random value between –2 and 0 i.e., $A \in rand(-2,0)$ and $C \in rand(0,2)$

Case 2: when, $(C \times X_p^t - X^t) \leq 0$,

Again, from equation (1),

$$X^{(t+1)} = (1 + A \times C) \times X_p^t - A \times X^t \tag{25}$$

In the whole process, we take $B_2 = (1 + A \times C)$.

As an earlier calculation, the amplification factor is calculated [Appendix A] for equation (25) same as equation (19).

$$G(k) = B_2 \times e^{\iota\theta} - A \tag{26}$$

By similar analysis and calculation as done earlier,

$$1 - (A - B_2)^2 \geq 4 \times A \times B_2 \times \sin^2\left(\frac{\theta}{2}\right)$$

Again, A is random value in (–2,2) i.e., $A \in rand(-2,2)$ and C is a random value between 0 and 2 i.e., $C \in rand(0,2)$.

Case 2(A): when $A = rand(0,2)$ and $C = rand(0,2)$ then B_2 is positive.

$$\left[\frac{1-(A-B_2)^2}{4\times A\times B_2}\right] \geq \sin^2\left(\frac{\theta}{2}\right)$$

$$\left[\frac{1-(A-B_2)^2}{4\times A\times B_2}\right] \geq 0 \qquad \because \quad \sin^2\left(\frac{\theta}{2}\right) \geq 0$$

$$1 - (A - B_2)^2 \geq 0$$

$$(A - B_2)^2 \leq 1 \tag{27}$$

By using condition $B_2 > 0$ and obtained inequality (27), it can be concluded that A is random value between 0 and 2 i.e., $A \in rand(0,2)$ and C takes random value between 0 and 2 i.e., $C \in rand(0,2)$.

Case 2(B): when $A = rand(-2,0)$ and $C = rand(0,2)$ the B_2 may be positive or negative.

Subcase (a): when B_2 is positive.

$$\left[\frac{1-(A-B_2)^2}{4\times A\times B_2}\right] \le \sin^2\left(\frac{\theta}{2}\right)$$

$$\left[\frac{1-(A-B_2)^2}{4\times A\times B_2}\right] \le 1 \qquad \because \quad \sin^2\left(\frac{\theta}{2}\right) \le 1$$

$$1-(A+B_2))^2 \ge 0$$

$$(A+B_2)^2 \le 1 \tag{28}$$

In this case, by using condition $B_2 > 0$ and obtained inequality (28), it can be concluded that A is random value between –1 and 0 i.e., $A \in rand(-1,0)$ and C takes a random value between 0 and 1 i.e., $C \in rand(0,1)$.

Subcase (b): when B_2 is negative.

$$\left[\frac{1-(A-B_2)^2}{4\times A\times B_2}\right] \ge \sin^2\left(\frac{\theta}{2}\right)$$

$$\left[\frac{1-(A-B_2)^2}{4\times A\times B_2}\right] \ge 0 \qquad \because \quad \sin^2\left(\frac{\theta}{2}\right) \ge 0$$

$$1-(A-B_2)^2 \ge 0$$

$$(A-B_2)^2 \le 1 \tag{29}$$

By using condition $B_2 < 0$ and obtained inequality (29), it can be concluded that A is random value between –2 and 1 i.e., $A \in rand(-2,1)$ and C takes random value between 1 and 2 i.e., $C \in rand(1,2)$.

The above discussed cases and the respective stable conditions (or stable range of parameters, A and C) are outlined in Table 1.

Table 1

The values of parameters A and C using von Neumann stability criteria

Cases	Considered Parameter value	Relation between A and C	Conclusion
1. $(C \times X_p^t - X^t) > 0$	$A \in rand(0,2)$ and $C \in rand(0,2)$	$B_1 > 0$ $(A - B_1)^2 \leq 1$	$A \in rand(0,1)$ $C \in rand(0,1)$
		$B_1 < 0$ $(A + B_1)^2 \leq 1$	$A \in rand(1,2)$ $C \in rand(1,2)$
	$A \in rand(-2,0)$ and $C \in rand(0,2)$	$B_1 > 0$ $(A + B_1)^2 \leq 1$	$A \in rand(-2,0)$ $C \in rand(0,2)$
2. $(C \times X_p^t - X^t) \leq 0$	$A \in rand(0,2)$ and $C \in rand(0,2)$	$B_2 > 0$ $(A - B_2)^2 \leq 1$	$A \in rand(0,2)$ $C \in rand(0,2)$
	$A \in rand(-2,0)$ and $C \in rand(0,2)$	$B_2 > 0$ $(A + B_2)^2 \leq 1$	$A \in rand(-1,0)$ $C \in rand(0,1)$
		$B_2 < 0$ $(A - B_2)^2 \leq 1$	$A \in rand(-2,1)$ $C \in rand(1,2)$

This table has discussed two cases, e.g., case 1 and case 2. In these cases, von Neumann stability criteria have been applied. After applying stability criteria, many cases have arisen, which are discussed in subsection 3.2. Using von Neumann stability criteria, we determine the best values for the parameters A and C. From the above discussion, it can be concluded that the values of parameters A and C lie in the range that was proposed for these parameters in [26].

The next section explains the numerical results of the stability analysis of GWO (StabeGWO). It shows that the values of the parameters are in the stable range and the stability analysis of GWO (StabeGWO) has the same behavior as GWO.

4. Numerical results

In this section, the validation of the GWO with parameters which satisfy the stability conditions, call it- StableGWO (StableGWO) algorithm is performed by numerical experiments on 23 classical benchmark functions which are commonly used by many researchers [26, 39, 33, 37]. These functions can be divided into three types of functions namely, unimodal, multimodal, and multimodal functions with fixed dimensional. Since the unimodal functions contain only one optima called global optima, they can be utilized to evaluate the local search ability of the algorithm and the multimodal functions contain more than one optimal solution, so that these can be used to evaluate global exploration ability. In order to ensure the comparability of the simulation experiments, the experimental parameters which are used in the StableGWO are set as:

Number of wolves $(SN) = 30$,

Maximum iteration $(t_{max}) = 500$,

The number of independent runs = 30.

4.1 *Analysis of the results*

In this subsection, the performance evaluation of the StableGWO is performed through various metrics such as mean, minimum, maximum, median, and standard deviation (STD). The results of StableGWO are compared to the results obtained with GWO and GWO1. For a fair comparison, the experimental setting is the same for GWO and GWO1. GWO indicates the range of A in $(-2,2)$ and the range of C in $(0,2)$ and GWO1 indicates the range of A in $(-3,3)$ and the range of C in $(0,2)$. The obtained results are reported in Table 2. In Table 2, for F1-F7 functions, StableGWO is significantly better than GWO1 and performs the same as the GWO algorithm. For F8-F13 functions, StableGWO is better than GWO and GWO1 in terms of mean value, and for F9 and F11 functions, StableGWO, GWO, and GWO1 obtained optimal value. In F10, StableGWO is performed better in F12 and F13. For F14-F23 functions, the GWO1 algorithm performed better than StableGWO and GWO. In F16, F17, F20, and F21, StableGWO is better and in F21-F23, StableGWO, GWO, and GWO1 obtained optimal value.

To demonstrate the significant difference between the StableGWO, GWO, and GWO1 algorithms, a Wilcoxon-rank sum test at a 5% level of significance has been carried out. The p-value and corresponding conclusion have also been represented in Table 3. In Table 3, "+" denotes significantly better, "–" refers significantly worse, and "≈" indicates similar. NaN (not a Numerical) indicates that the test is not applicable to similar solution vectors. The StableGWO algorithm is the same as the GWO algorithm for F1-F23 except for F11. In function F11, StableGWO does not perform better. StableGWO is performed better in comparison to the GWO1 algorithm in functions F1-F4, F10, F16, F17, F20, F21, and F22.

Furthermore, the convergence behavior of some selected benchmark functions F1, F3, F6, F7, F9, F11, F14, F17, F20 and F23 are shown in Figure 3. In such figures, on horizontal axis denotes the number of iterations, and on vertical axis indicates the best score (α score) in the intermediate iterations for a single run. In these figures, the convergence rate of StableGWO same as the GWO algorithm.

Table 2

Comparison of Results obtained by StableGWO, GWO, and GWO1 on 23 classical benchmark functions

Function	Algorithm	Mean	Minimum	Maximum	Median	STD
	StableGWO	**1.34E-31**	**6.74E-34**	**5.62E-31**	**8.76E-32**	**1.62E-31**
F1	**GWO**	3.00E-31	3.28E-33	1.70E-30	1.56E-31	4.16E-31
	GWO1	6.35E-25	7.46E-27	4.65E-24	1.93E-25	1.07E-24
	StableGWO	3.38E-19	**1.74E-20**	1.35E-18	2.45E-19	3.08E-19
F2	**GWO**	2.72E-19	3.26E-20	1.00E-18	1.70E-19	2.53E-19
	GWO1	9.81E-16	6.97E-17	6.93E-15	5.95E-16	1.23E-15
	StableGWO	**3.36E-06**	**2.60E-10**	**3.55E-05**	3.72E-07	**7.68E-06**
F3	**GWO**	1.63E-05	3.69E-10	0.000204541	2.79E-07	5.13E-05
	GWO1	0.001599545	5.26E-07	0.028659993	0.000132327	0.005335064
	StableGWO	2.71E-06	1.43E-07	3.37E-05	**7.70E-07**	7.12E-06
F4	**GWO**	**2.01E-06**	**9.14E-08**	**1.03E-05**	1.12E-06	**2.52E-06**
	GWO1	0.000150888	4.67E-06	0.000891406	6.42E-05	0.000201049
	StableGWO	27.12355125	**25.93090586**	**29.49135076**	**27.11885485**	0.913807199
F5	**GWO**	27.34395156	26.05559344	28.77939111	27.17847622	0.869403974
	GWO1	**27.11730427**	25.9458105	28.75701076	27.13230818	**0.769944863**
	StableGWO	0.918660352	0.000165775	1.775965121	0.882292536	0.473006744
F6	**GWO**	0.782782433	**8.64E-05**	**1.510812255**	0.757370174	**0.380594874**
	GWO1	**0.737866424**	0.000385655	1.519159871	**0.730523999**	0.424489966

Contd...

	StableGWO	0.003033099	0.00082825	0.008128232	0.002585117	0.001673333
F7	**GWO**	**0.002616563**	0.000905647	**0.00534812**	**0.002456694**	**0.000977994**
	GWO1	0.003724349	**0.000620238**	0.009728167	0.003402236	0.002029109
	StableGWO	**-6408.746544**	-8054.43197	-3193.147726	-6418.948487	**937.8221955**
F8	**GWO**	-6356.388823	-7712.524775	-3595.24404	**-6563.333436**	1004.806
	GWO1	-6056.957638	**-8516.459683**	**-3690.781754**	-6481.564073	1666.878842
	StableGWO	8.521519434	0	31.00115006	**5.755922473**	8.891314718
F9	**GWO**	10.04039697	0	35.69714859	7.287431798	8.431554069
	GWO1	**7.49885094**	**0**	**18.92573346**	7.159423884	**5.879173404**
	StableGWO	**3.37E-14**	**2.93E-14**	**4.35E-14**	**3.29E-14**	4.14E-15
F10	**GWO**	3.42E-14	2.93E-14	4.35E-14	3.29E-14	**4.12E-15**
	GWO1	2.725827406	6.48E-14	20.51133004	2.19E-13	7.068374576
	StableGWO	0.007355793	0	0.037026274	0	0.011577079
F11	**GWO**	0.002351183	0	0.018693359	0	0.005482864
	GWO1	0.003864345	0	0.023478213	0	0.008178915
	StableGWO	0.056904662	**0.01569415**	0.131798181	0.050120182	0.029025729
F12	**GWO**	0.062379459	0.018753801	0.155444685	0.04815991	0.034540774
	GWO1	**0.04992272**	0.018141645	**0.099393363**	**0.046172614**	**0.019708632**
	StableGWO	0.801301665	0.30322729	1.24749455	0.796442876	**0.223130771**
F13	**GWO**	0.817685241	0.28835001	1.234320261	0.827182874	0.24454802
	GWO1	**0.699691468**	**0.223648188**	**1.123658046**	**0.689298899**	0.231672149
	StableGWO	5.425549884	0.998003838	12.67050581	2.982105157	4.9017482
F14	**GWO**	4.291433922	0.998003838	12.67050581	2.982105157	4.361775889
	GWO1	**2.437478495**	**0.998003838**	**10.76318067**	**0.998003838**	**2.944720718**
	StableGWO	0.003142417	**0.000307488**	**0.020363341**	0.000321173	0.006878567
F15	**GWO**	0.003137654	0.000307544	0.020363348	0.000368376	0.006878574
	GWO1	**0.001820985**	0.000307688	0.020363342	**0.000384232**	**0.005047424**
	StableGWO	**-1.031628444**	**-1.031628453**	**-1.031628406**	**-1.031628446**	**1.02E-08**
F16	**GWO**	-1.031628446	-1.031628453	-1.031628412	-1.03162845	1.03E-08
	GWO1	-1.031628421	-1.031628453	-1.031628304	-1.031628432	3.61E-08
	StableGWO	**0.397887579**	**0.39788738**	**0.397888098**	**0.397887521**	**2.04E-07**
F17	**GWO**	0.397887642	0.397887373	0.397888331	0.397887577	2.70E-07
	GWO1	0.397888226	0.397887372	0.397891813	0.397887974	9.69E-07
	StableGWO	5.700028459	3.000003696	84.00001411	3.000017712	14.78850634
F18	**GWO**	5.700020677	3.000000041	84.00002639	3.00001265	14.78851013
	GWO1	**3.000021995**	**3.000000003**	**3.000178944**	**3.000006756**	**3.75E-05**
	StableGWO	**-0.300478907**	**-0.300478907**	**-0.300478907**	**-0.300478907**	**2.26E-16**

F19	GWO	**-0.300478907**	**-0.300478907**	**-0.300478907**	**-0.300478907**	**2.26E-16**
	GWO1	**-0.300478907**	**-0.300478907**	**-0.300478907**	**-0.300478907**	**2.26E-16**
	StableGWO	**-3.263294493**	**-3.321993744**	**-3.018892349**	**-3.321978468**	**0.08520441**
F20	GWO	-3.233464878	-3.321992027	-3.018250864	-3.321946772	0.104296686
	GWO1	-3.151803248	-3.321989166	-1.840911133	-3.179983574	0.26194494
	StableGWO	**-8.970772402**	**-10.15310337**	**-5.055194793**	**-10.15257032**	**2.179079979**
F21	GWO	-8.551745587	-10.15313216	-2.682767049	-10.15242419	2.522414307
	GWO1	-8.547761634	-10.15308093	-2.682759378	-10.15078677	2.526367658
	StableGWO	-10.04927558	-10.40281835	-5.087656775	-10.40235323	1.34321338
F22	GWO	**-10.22666271**	**-10.40293709**	**-5.128678173**	**-10.40255385**	**0.962855627**
	GWO1	-9.693986204	-10.40234468	-5.087621363	-10.40110949	1.833579939
	StableGWO	-9.275134907	-10.53626742	-2.421683783	-10.53571509	2.903117491
F23	GWO	-9.797287082	-10.53620533	-1.859420858	-10.53559152	2.300645742
	GWO1	**-9.815718129**	**-10.53634714**	**-5.12844822**	**-10.53503997**	**1.865381438**

Table 3

Results of Wilcoxon signed-rank sum Test for 23 classical benchmark functions

Function	Algorithm	GWO	GWO1
F1	**p-value**	NaN	1.73E-06
	conclusion	≈	+
F2	**p-value**	NaN	1.73E-06
	conclusion	≈	+
F3	**p-value**	NaN	3.52E-06
	conclusion	≈	+
F4	**p-value**	NaN	2.60E-06
	conclusion	≈	+
F5	**p-value**	NaN	NaN
	conclusion	≈	≈
F6	**p-value**	NaN	NaN
	conclusion	≈	≈
F7	**p-value**	NaN	NaN
	conclusion	≈	≈
F8	**p-value**	NaN	NaN
	conclusion	≈	≈

Contd...

F9	**p-value** **conclusion**	NaN ≈	NaN ≈
F10	**p-value** **conclusion**	NaN ≈	1.73E-06 +
F11	**p-value** **conclusion**	3.53E-02 –	NaN ≈
F12	**p-value** **conclusion**	NaN ≈	NaN ≈
F13	**p-value** **conclusion**	NaN ≈	NaN ≈
F14	**p-value** **conclusion**	NaN ≈	2.18E-02 –
F15	**p-value** **conclusion**	NaN ≈	NaN ≈
F16	**p-value** **conclusion**	NaN ≈	1.48E-03 +
F17	**p-value** **conclusion**	NaN ≈	7.51E-05 +
F18	**p-value** **conclusion**	NaN ≈	1.32E-02 –
F19	**p-value** **conclusion**	NaN ≈	NaN ≈
F20	**p-value** **conclusion**	NaN ≈	1.25E-04 +
F21	**p-value** **conclusion**	NaN ≈	7.27E-03 +
F22	**p-value** **conclusion**	NaN ≈	5.29E-04 +
F23	**p-value** **conclusion**	NaN ≈	NaN ≈

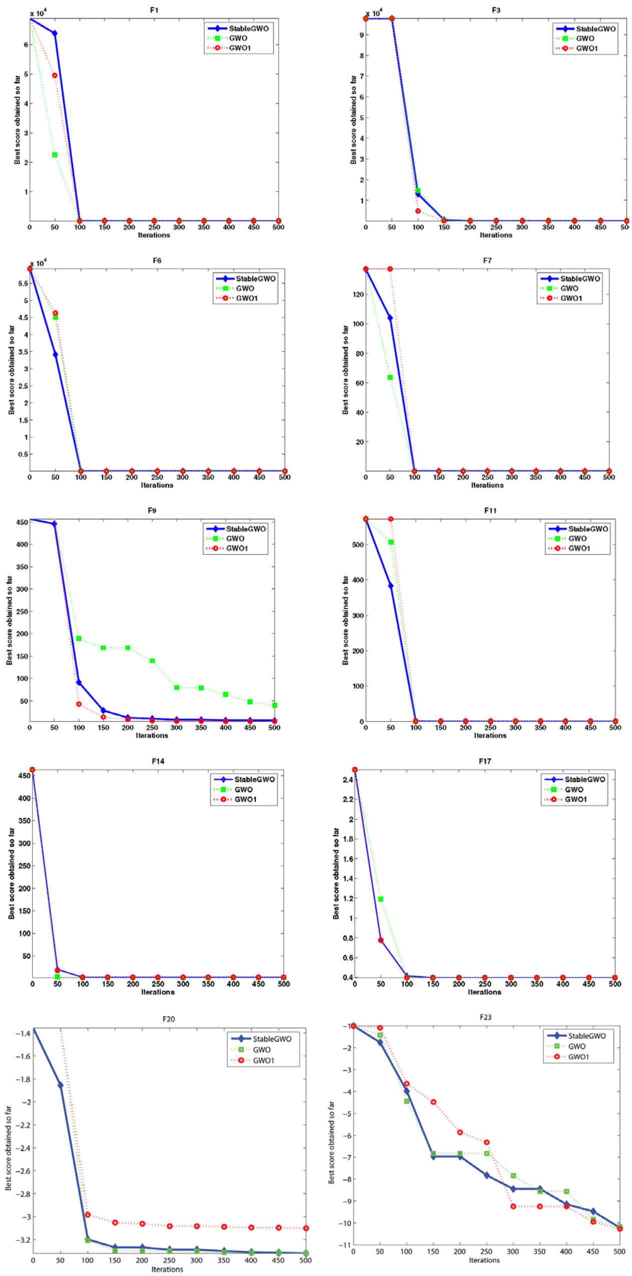

Figure 3

Convergence Curves

From the numerical results, statistical analyses, and convergence rate, it can be concluded that the StableGWO performs similarly to the Standard GWO algorithm on these benchmark problems. So from this analysis, it can also be concluded that StableGWO (or Standard GWO) is a reliable algorithm and its parameters are in stable range. Since it is now found that Standard GWO and StableGWO both perform the same on the test problems or in other words StableGWO performance does not deteriorate as compared to GWO. Thus for real-world problems, StableGWO is recommended as compared to GWO with any other parameter settings, because if unstable GWO is applied to solve real-world problems, an unbounded error may occur.

5. Conclusion and future work

For nature-inspired algorithms, finding a stable range of user parameters is a crucial and difficult task. This paper presents the stability analysis of GWO using von Neumann stability criteria and the conditions on parameters A and C are obtained. It is found that the range of the parameters for StableGWO is different than that of the GWO. The GWO with parameters in the stable range (StableGWO) has been tested and compared with standard GWO on 23 classical benchmark problems. The performance of GWO and StableGWO over test problems is found to be the same. This concludes that StableGWO with a stable range of parameters will not provide inferior results than GWO while providing the guarantee of stable solutions and therefore the bounded error. Thus for real-world problems, StableGWO is recommended over GWO.

Because of the ease of implementation of the von Neumann stability criterion, it can be used to find the stability condition for a variety of other swarm intelligence-based optimization algorithms and it can also further be extended for the convergence analysis of the GWO algorithm.

Conflict of interest: The authors declare no conflict of interest.

Data Availability: No data were used to support the findings of the study.

Appendix A

According to the discussion in the subsection (3.2), to calculate the amplification factor, each term X_l^n of the equation (19) replace by $w^n(m) \times e^{iml\Delta i}$. After substituting the value of X_l^n, equation (19) is modified in this way:

$$w^{n+1}(m)\times e^{\imath(ml\Delta i)} = B_1 \times w^n(m)\times e^{\imath m(l\pm b)\Delta i} + A\times w^n(m)\times e^{\imath(ml\Delta i)}$$

$$w^{n+1}(m) = (B_1 \times e^{\imath m(\pm b)\Delta i} + A)\times w^n(m)$$

By comparing from equation (15),

$$G(m) = B_1 \times e^{\imath\theta} + A \tag{30}$$

where, $\theta = m(\pm b)\Delta i$

Similarly, the amplification factor can be calculated for equation (25). For equation (30), the amplification factor is calculated as:

$$G(m) = B_1 \times e^{\imath\theta} + A$$

$$G(m) = B_1 \times (\cos\theta + \imath\sin\theta) + A$$

$$G(m) = (B_1 \times \cos\theta + A) + \imath\sin\theta \times B_1$$

By taking the modulus on both sides, we get,

$$|G(m)| = \sqrt{(B_1 \times \cos\theta + A)^2 + \sin^2\theta \times B_1^2}$$

$$|G(m)| = \sqrt{B_1^2 + A^2 + 2\times A\times B_1 \times \cos\theta}$$

$$|G(m)| = \sqrt{B_1^2 + A^2 + 2\times A\times B_1 \times (1 - 2\sin^2\theta/2)}$$

$$|G(m)| = \sqrt{B_1^2 + A^2 + 2\times AB_1 - 4\times A\times \sin^2\left(\frac{\theta}{2}\right)B_1}$$

$$|G(m)| = \sqrt{(A+B_1)^2 - 4\times A\times \sin^2\left(\frac{\theta}{2}\right)B_1}$$

References

[1] A. Abraham, A. Konar, N. R. Samal, and S. Das, "Stability analysis of the ant system dynamics with non-uniform pheromone deposition rules," in 2007 IEEE Congress on Evolutionary Computation, pp. 1103–1108 (2007).

[2] M. A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, "Natural selection methods for grey wolf optimizer," *Expert Systems with Applications*, vol. 113, pp. 481–498 (2018).

[3] J. C. Bansal, A. Gopal, and A. K. Nagar, "Analysing convergence, consistency, and trajectory of artificial bee colony algorithm," *IEEE Access*, vol. 6, pp. 73593–73602 (2018).

[4] J. C. Bansal, A. Gopal, and A. K. Nagar, "Stability analysis of artificial bee colony optimization algorithm," *Swarm and Evolutionary Computation*, vol. 41, pp. 9–19 (2018).

[5] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic Computing* vol. 6, no. 1, pp. 31–47 (2014).

[6] J. C. Bansal and S. Singh, "A better exploration strategy in grey wolf optimizer," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–20 (2020).

[7] A. Biswas, S. Das, A. Abraham, and S. Dasgupta, "Stability analysis of the reproduction operator in bacterial foraging optimization," *Theoretical Computer Science*, vol. 411, no. 21, pp. 2127–2139 (2010).

[8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995).

[9] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381 (2016).

[10] A. Entesar, H. S. Mahmood, and O. S. Qasim, "Solving nonlinear differential equations using a hybrid method between the variational iteration method and grey wolf optimization algorithm," *Journal of Interdisciplinary Mathematics*, vol. 25, no. 2, pp. 573–585 (2022).

[11] F. Farivar and M. A. Shoorehdeli, "Stability analysis of particle dynamics in gravitational search optimization algorithm," *Information Sciences*, vol. 337, pp. 25–43 (2016).

[12] R. Ghosh, N. Sinha, S. K. Biswas, and S. Phadikar, "A modified grey wolf optimization based feature selection method from EEG for silent speech classification," *Journal of Information and Optimization Sciences*, vol. 40, no. 8, pp. 1639–1652 (2019).

[13] A. Gopal and J. C. Bansal, "Stability analysis of differential evolution," in 2016 International Workshop on Computational Intelligence (IWCI), pp. 221–223 (2016).

[14] A. Gopal, M. M. Sultani, and J. C. Bansal, "On stability analysis of particle swarm optimization algorithm," *Arabian Journal for Science and Engineering*, pp. 1–10 (2019).

[15] M. Gopal, Digital Control & State Variable Methods, Tata McGraw-Hill Education (2012).

[16] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of interconnected power system using grey wolf optimization," *Swarm and Evolutionary Computation*, vol. 27, pp. 97–115 (2016).

[17] W. Hahn, H. H. Hosenthien, and H. Lehnigk, *Theory and Application of Liapunov's Direct Method*, vol. 3, Prentice-Hall Englewood Cliffs (1963).

[18] Y. He and W. Sun, "Stability and convergence of the Crank–Nicolson/Adams–Bashforth scheme for the time-dependent Navier–Stokes equations," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 837–869 (2007).

[19] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471 (2007).

[20] G. M. Komaki and V. Kayvanfar, "Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *Journal of Computational Science*, vol. 8, pp. 109–120 (2015).

[21] N. Li, D.-B. Sun, T. Zou, Y.-Q. Qin, and Y. Wei, "An analysis for a particle's trajectory of PSO based on difference equation," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 29, no. 11, pp. 2052–2061 (2006).

[22] Q. Liu, "Order-2 stability analysis of particle swarm optimization," *Evolutionary Computation*, vol. 23, no. 2, pp. 187–216 (2015).

[23] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 63–80 (2018).

[24] A. Madadi and M. M. Motlagh, "Optimal control of DC motor using grey wolf optimizer algorithm," *Tech J Eng Appl Sci*, vol. 4, no. 4, pp. 373–379 (2014).

[25] K. Meidani, A. H. Hemmasian, S. Mirjalili, and A. B. Farimani, "Adaptive grey wolf optimizer," *Neural Computing and Applications*, pp. 1–21 (2022).

[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software* vol. 69, pp. 46–61 (2014).

[27] C. Muro, R. Escobedo, L. Spector, and R. P. Coppinger, "Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behavioural Processes*, vol. 88, no. 3, pp. 192–197 (2011).

[28] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Systems with Applications*, vol. 166, p. 113917 (2021).

[29] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248 (2009).

[30] L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, C. I. Gonzalez, G. E. Martinez, and J. Soto, "A fuzzy hierarchical operator in the grey wolf optimizer algorithm," *Applied Soft Computing*, vol. 57, pp. 315–328 (2017).

[31] N. R. Samal, A. Konar, S. Das, and A. Abraham, "A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence," in 2007 IEEE Congress on Evolutionary Computation, pp. 1769–1776 (2007).

[32] S. Singh and J. C. Bansal, "Grey wolf optimizer with crossover and opposition-based learning," in International Conference on Harmony Search Algorithm, pp. 401–410 (2020).

[33] S. Singh and J. C. Bansal, "Mutation-driven grey wolf optimizer with modified search mechanism," *Expert Systems with Applications*, p. 116450 (2022).

[34] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman, "Using the grey wolf optimizer for solving optimal reactive power dispatch problem," *Applied Soft Computing*, vol. 32, pp. 286–292 (2015).

[35] M. Vidyasagar, *Nonlinear Systems Analysis*, SIAM (2002).

[36] R. F. Warming and B. J. Hyett, "The modified equation approach to the stability and accuracy analysis of finite-difference methods," *Journal of Computational Physics*, vol. 14, no. 2, pp. 159–179 (1974).

[37] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-inspired Computation*, vol. 2, no. 2, pp. 78–84 (2010).

[38] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–214 (2009).

[39] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102 (1999).

Received December, 2022