Recep Tayyip Erdoğan Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, 6(1) 220-244 2025
Recep Tayyip Erdogan University Journal of Science and Engineering, 6(1) 220-244 2025
ISSN: 2687-2315 / E-ISSN: 2757-7686

Improved Gray Wolf Optimization Algorithm for Tuning Non-integer Order Proportional Integral Derivative Controller Design

Kadri DOĞAN*[1], Hasan BAŞAK[2]

[1]Artvin Çoruh University, Faculty of Engineering, Basic Science, Artvin, Türkiye
[1]Artvin Çoruh University, Faculty of Engineering, Electrical-Electronics Engineering, Artvin, Türkiye

Sorumlu Yazar/Corresponding Author
E-mail: dogankadri@artvin.edu.tr

Araştırma Makalesi/Research Article
Geliş Tarihi/Received: 09.09.2024
Kabul Tarihi/Accepted: 03.04.2025

Abstract

In this study, a noninteger-order proportional–integral–derivative (NIOPID) controller was used for controlling the speed of the direct current (DC) motor. The controller parameters have optimally been adjusted using the GWOJOS algorithm formed by combining the Grey Wolf Optimization (GWO) algorithm and the recently defined the Joint Opposite Selection (JOS) feature. The JOS brings a mutual reinforcement by a joint of the two opposition strategies Dynamic Opposite (DO) and Selective Leading Opposition (SLO). The DO and SLO improve the balance of exploration and exploitation, respectively, in a given search space. During the optimization phase, JOS helps GWO attack the target quickly by employing SLO. DO help GWO find more opportunities to find the most suitable prey. The GWO is able to improve its performance with JOS. This combination helps accelerating the convergence rate of GWO. We assessed GWOJOS's performance using the benchmark functions from the IEEE Congress on Evolutionary Computation 2017 (CEC2017). The benchmark covers composition, hybrid, multimodal, and unimodal functions. The NIOPID-based speed control system for DC-motor using the GWOJOS algorithm has been designed using a time domain objective function that takes into account the performance criteria (maximum overshoot, steady-state error, rising time, and settling time). Some analyses, including robustness, time and frequency domain simulations, have been used to evaluate the performance of the proposed novel approach. The evaluation results have shown that the performance of GWOJOS was better than the performance of GWO, Slime Mould Algorithm (SMA), Atom Search Optimization (ASO), Simulated Annealing (SA) and the hybrid optimization algorithm created by opposition-based learning (OBL) strategy of SA and SMA algorithms (OBLSMASA).

Anahtar Kelimeler: NIOPID, DC Motor, GWO JOS, Metaheuristic Optimization

Geliştirilmiş Gri Kurt Optimizasyon Algoritmasına Dayalı Kesirli Mertebeden Oransal İntegral Türevsel PID Denetleyici Tasarımı

Öz

Bu çalışmada, doğru akım (DC) motorunun hızını kontrol etmek için tam sayı olmayan mertebeden oransal-integral-türevsel (NIOPID) kontrolör kullanılmıştır. Kontrolör parametreleri, GWO algoritması ve yeni tanımlanan JOS özelliğinin birleşiminden oluşan GWOJOS algoritması kullanılarak optimum şekilde ayarlanmıştır. JOS, Dinamik Karşıtlık (DO) ve Seçici Lider Karşıtlık (SLO) olmak üzere iki karşıtlık stratejisinin bir araya getirilmesiyle karşılıklı bir güçlendirme sağlar. DO ve SLO, belirli bir arama uzayında sırasıyla keşif ve sömürü dengesini iyileştirir. Optimizasyon aşamasında JOS, SLO'yu kullanarak GWO'nun hedefe hızlı bir şekilde saldırmasına yardımcı olur. DO, GWO'nun en uygun avı bulmak için daha fazla fırsat bulmasına yardımcı olur. GWO, JOS ile performansını artırabilmektedir. Bu birleşim, GWO'nun yakınsama oranını hızlandırmaya yardımcı olur. GWOJOS'un performansını CEC2017'deki kıyaslama fonksiyonlarını kullanarak değerlendirdik. Kıyaslama bileşim, hibrit, multimodal ve unimodal fonksiyonları kapsamaktadır. GWOJOS algoritmasını kullanan DC-motor için NIOPID tabanlı hız kontrol sistemi, performans kriterlerini (maksimum aşım, kararlı durum hatası, yükselme süresi ve yerleşme süresi) dikkate alan bir zaman alanı amaç fonksiyonu kullanılarak tasarlanmıştır. Önerilen yeni yaklaşımın performansını değerlendirmek için sağlamlık, zaman ve frekans alanı simülasyonları dahil olmak üzere bazı analizler kullanılmıştır. Değerlendirme sonuçları GWOJOS'un performansının GWO, SMA, ASO, SA ve OBLSMASA algoritmaların performansından daha iyi olduğunu göstermiştir.

Keywords: NIOPID, DC Motor, GWOJOS, Metaheuristic Optimizasyon

Cite as;
Doğan, K. & Başak, H. (2025). Improved Gray Wolf Optimization Algorithm for Tuning Non-integer PID Controller Design. *Recep Tayyip Erdogan University Journal of Science and Engineering,* 6(1), 220-244. Doi: 10.53501/rteufemud.1545913

1. Introduction

Optimization is not only a collection of techniques and tools or a way of thinking; it is also a fundamental requirement in the industry (Xu et al., 2021; Zhao et al., 2021, Zhang et al., 2022), and it falls under a large category of methods such as fuzzy logic optimization (Chen et al., 2019), robust optimization (Qu et al., 2021), multi-branch learning methods (He et al., 2020), multiobjective optimization (Cao et al, 2022, Lin et al., 2022), large-scale tasks (Liu et al., 2021), constraint optimization, multi-dimensional optimizer (Meng et al., 2018), Wang et al., 2021). Large-Scale Tasks grow in complexity and dimensionality, specialized optimization methods are necessary and constraint optimization ensures feasible solutions while optimizing the objective functions. Memetic methods combine global search (e.g., genetic algorithms) with local search (e.g., gradient-based methods). Over the last few decades, numerous deterministic approaches have been created and developed to solve optimization problems However, deterministic models require knowledge of the properties of the optimization problem and the gradient information (Meng et al., 2019, Gursoy and Gunnec, 2018). The simplicity of metaheuristic optimization algorithms, reaching absolute optimum, applicability and derivative-free have made them significantly widespread. The sources of inspiration for swarm intelligence techniques are mostly natural colonies, flocks, herds, and schools. In this field, some popular MOAs include Evolutionary Programming (EP) (Sebald and Fogel, 1994), Genetic Algorithms (GA) (Holland, 1992), Ant Colony Optimization (ACO) (Dorigo and Socha, 2007), Evolutionary Strategy (ES) (Alavi and Henderson, 1981), Differential Evolution (DE) (Lampinen and Storn, 2004), GWO (Mirjalili et al., 2014), and Particle Swarm Optimization (PSO) (Kennedy and Eberhat, 1995).

Drives that convert electrical energy into mechanical energy are called DC motors. DC motors are extensively employed as the primary drive in a wide range of industrial applications because of their comparatively low cost, excellent durability, and simple controllability (Ali, 2015, Rodriguez-Molina et al., 2017, Potnuru et al., 2019). Meta-heuristic algorithms have been employed in studies pertaining to the speed control of DC motors. Conventional controllers, such as linear proportional-integral (PI) and proportional–integral–derivative (PID) controllers, have been extensively employed in the literature to regulate DC motors (Griffin, 2003). NIOPID control is a non-conventional control method used in this study to operate the DC motor. The use of non-integer calculus to control theory has been explained in numerous studies (Podlubny, 1999, Petras, 1999), and its benefits have also been demonstrated. The traditional PID controller, which is based on non-integer calculus, has been expanded into the NIOPID controller. To achieve the desired controller, it is necessary to employ a highly effective tuning method in place of time-consuming, low-performing classical tuning methods (Çelik and Öztürk, 2018).

In this study, our aim is to enhance the GWO algorithm by using Joint Opposite Selection (JOS) (Arini et al., 2022b) strategy for finding the right balance between two stages exploration and exploitation. As shown in Table 1, the superiority of the improved GWO (GWOJOS) has been demonstrated using the benchmark CEC2017 (Awad et al., 2016) functions. This superiority has been proven by statistical analysis. Furthermore, it has been demonstrated that the NIOPID parameters are effectively tuned for DC speed control using the GWOJOS algorithm

2. Overview of the Gray Wolf Algorithm (GWO)

The Grey Wolf (Canis lupus) is a member of the Canidae family. Grey wolves are regarded as apex predators at the top of the food chain. Figure 1 illustrates their rigid social hierarchy, which is very intriguing. The pack is subject to the alpha's judgments. However, there has also been evidence of a democratic tendency when an alpha wolf follows the other wolves in the pack (Mirjalili et al., 2014),

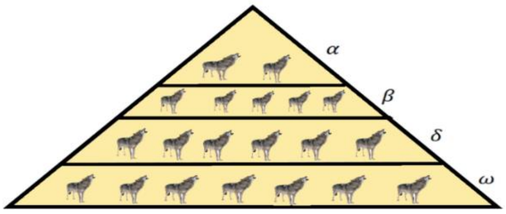

Figure 1. Social hierarchy of the Gray Wolves

The social hierarchy and group hunting of Grey Wolves are other interesting social behaviors, shown in Figure 2. The phases of gray wolf hunting are (Mirjalili et al., 2014):

i) Tracking, pursuing, and getting close to the prey,

ii) chasing, encircling, and intimidating the victim until it gives up,

iii) Assault the target.

Figure 2. The behavioral phases of the gray wolf in its natural life are shown as three main phases. *Phase 1*, searching for prey; *Phase 2*, attacking prey and *Phase 3*, catching prey.

To design the social hierarchy of Gray Wolves and perform optimization, the mathematical model was established as follows:

Encircling prey: $\vec{E} = |\vec{U}\vec{X}_p(t) - \vec{X}(t)|$, $\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\vec{E}$, $t =$ current iteration, $\vec{U}$ *and* $\vec{A}$, coefficient vectors $\vec{X}_p =$ the position of prey, and $\vec{X} =$ Possition of predator. Also, it calculated as $\vec{A} = 2\vec{a}\vec{r}_1 - \vec{a}$ *and* $\vec{D} = 2\vec{r}_2$ such that $\vec{a} \in [0,2]$ is linearly decreasing, $\vec{r}_1, \vec{r}_2 \in [0,1]$ are random vectors.

Hunting: Gray wolves have the ability to recognize the location of prey and surround them. The prey is usually guided by the alpha. Beta and delta may also occasionally participate in hunting. However, nothing can be determined about the location of the prey in a search space. It is assumed that alpha, beta, and delta wolves possess superior knowledge regarding the possible location of prey to statistically describe the hunting behavior of wolves. This saves the top three best solutions obtained so far and forces other search agents to update their positions based on the position of their best search agent. These narratives are formulated as follows:

$\vec{E}_\alpha = |\vec{U}_1\vec{X}_\alpha(t) - \vec{X}(t)|$, $\vec{E}_\beta = |\vec{U}_2\vec{X}_\beta(t) - \vec{X}(t)|$, *and* $\vec{E}_\delta = |\vec{U}_3\vec{X}_\delta(t) - \vec{X}(t)|$,

and

$\vec{X}_1(t) = \vec{X}_\alpha(t) - \vec{A}_1$, $\vec{X}_2(t) = \vec{X}_\beta(t) - \vec{A}_2$, *and* $\vec{X}_3(t) = \vec{X}_\delta(t) - \vec{A}_3$.

Thus, it was obtained

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

Attacking prey (exploitation): Gray wolves complete the hunt by attacking when the prey remains motionless, that is, when the prey's energy is exhausted, as shown in Figure 3. $|A| < 1$ is forcing the wolves to attack the prey.

The flow chart of the Gray Wolf's hunting strategy is given in Figure 4. As can be seen from this flow chart, the three main search positions used by gray wolves are alpha, beta, and delta. They split up to look for prey, then come together and attack it. When |A| >1, gray wolves are forced to leave their packs to find better prey.

Search for prey (exploration): The three main search positions used by gray wolves are alpha, beta, and delta. They split up to look for prey, then come together and attack it. When $|A| > 1$, gray wolves are forced to leave their packs to find better prey (Mirjalili et al., 2014).

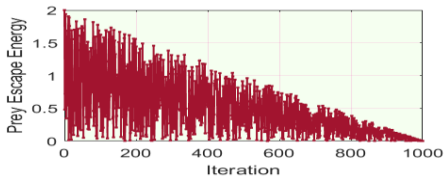

Figure 3. The evolution of a prey's absolute escape energy ran for 1000 iterations around.

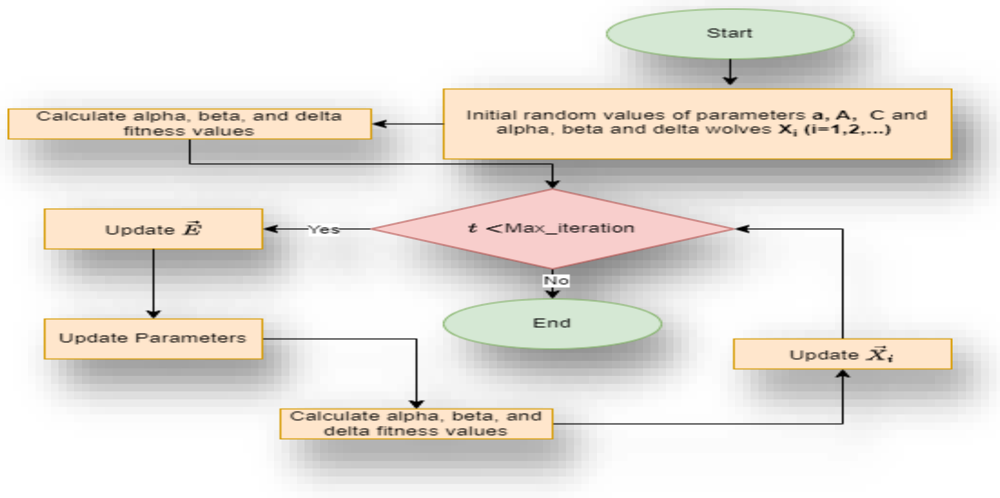

Figure 4. Flow Chart of GWO

In addition to the advantages of the Gray Wolf Optimization Algorithm such as simple applicability, high exploration capacity and flexible parameter settings, it also has disadvantages such as slow convergence, parameter sensitivity and local optimum problems. Therefore, the advantages and disadvantages of the algorithm should be taken into account depending on the problem and the requirements to be applied. Thus, an extra feature added to GWO can correct the disadvantages of GWO's hunting strategy and turn it into an advantage.

In order to improve metaheuristic optimization algorithms, many techniques have been proposed. The use of OBL, chaos, hybridization and so on are some of these techniques. One of the most popular and effective techniques is OBL (Tizhoosh, 2005). In the last few years, OBL algorithms have attracted the interest of a large number of computer scientists. Many of the well-known ones like GWO, GSA, SA, GA, PSO, ACO, ABC, DE, etc. Were improved using OBL. Opposite Centre Learning (OCL) was established by Liu et al. (2014), which defines the opposite point as the optimal solution among pairwise samples of the search space given a random starting point. Population-based search algorithms converge faster using this method. Rahnamayan et al. (2014) have proposed an OTDS (Oppositional Target-Domain Estimation) by dividing the search space into grids, which speeds up the estimation of the target domain. However, if the solution is close to a grid boundary, the computational complexity increases. There are few papers where researchers have used OBL in addition to GWO, according to a review of the literature. A simplex-based opposition is performed on all wolves in Elite Opposition-based Learning (EOGWO) (Zhang et al., 2017). The wolves are divided into two subgroups and the best wolves in one subgroup replace the worst wolves in the other subgroup in the Improved Grey Wolf Optimizer (IGWO) (Nasrabadi et al., 2016). In the two parts, one part is partially counterbalanced (in one dimension), and the remaining part is opposed in all the dimensions. With each iteration, the number of completely opposing wolves decreases. Pradhan et al. (2017) have combined GWO and OBL named Oppositional Grey Wolf Optimization (OGWO) algorithm for solving the optimal operating strategy of the Economic Load Dispatch (ELD) problem. The algorithm combines two fundamental approaches: the hunting behavior and social hierarchy and the accelerate to convergence rate of the traditional GWO algorithm (mahdavi et al., 2018). Dhargupta et al. (2020) have combined OBL with GWO (SOGWO) to enhance its exploratory behavior while maintaining a fast convergence rate.

2.1. Joint Opposite Selection (JOS)

Selective Leading Opposition and Dynamic Opposite are two opposition learning methods that are combined to create the Joint Opposite Selection operator. In a particular search space, the DO and SLO enhance the balance of exploration to exploitation, respectively. SLO calculates the search agents' close distance dimension using a threshold value that decreases linearly. DO gives search agents opportunities to develop their capabilities in the search domain (Xu et al., 2014). Dynamic opposite consists of the combination of two different opposites: Quasi-opposite and reflection-opposite.

Definition 1. (Dhargupta et al., 2020) Let $x \in [a, b]$ be a real number. The opposite number of x $(\hat{x})$ is defined as follows:

$$\hat{x} = a + b - x.$$

Definition 2. (Xu et al., 2005) Let $x \in [a, b]$ be a real number. The quasi-opposite point of x $\left(\hat{x}_q\right)$ is defined as follows:

$$\hat{x}_q = rand(m, \hat{x}\,).$$

Where $m = \frac{a+b}{2}$, and $rand(m, \hat{x}\,)$ is a random number uniformly distributed between $m \; and \; \hat{x}$.

Definition 3. (Xu et al., 2014) Let $x \in [a, b]$ be a real number. The quasi-opposite point of x $(\hat{x}_r)$ is defined as follows:

$$\hat{x}_r = rand(m, x\,).$$

Where $m = \frac{a+b}{2}$, and $rand(m, x)$ is a random number uniformly distributed between $m \; and \; x$.

Definition 4. (Xu et al., 2014) Let $P(x_1, x_2, x_3, \dots, x_n)$ be a point such that $x_1, x_2, x_3, \dots, x_n \in \mathbb{R}$ and $x_i \in [a_i, b_i]$. The opposite point of P $(\hat{P})$ is defined as follows:

$$\hat{x}_i = a_i + b_i - x_i.$$

Algorithm 1: Pseudocode of Selective Opposition (SO) (Dhargupta et al., 2020)

Input: initial generation t, maximum generation T, population size N, dimension d

Output: $\overline{\mathrm{X}}_{\mathrm{dc}}$: new position based on selective opposition

$$\text{Threshold: } 2 \times \left(1 - \frac{\mathrm{t}}{\mathrm{T}}\right)$$

for $\mathrm{i} = 1{:}\,\mathrm{N}$

if $\mathrm{X_i} \neq \mathrm{X_{ibest}}$

for $\mathrm{j} = 1{:}\,\mathrm{d}$

$\mathrm{dd_j} = \left|\mathrm{X_{ibest,j}} - \mathrm{X}\right|$ {$\mathrm{dd_j} =$ difference distance for each dimension}

if $\mathrm{dd_j} >$ threshold

Identify the close distance dimensions $\mathrm{d_f}$

Count the number of close distance dimensions $\mathrm{d_f}$

else

Identify the faraway distance dimensions $\mathrm{d_c}$

Count the number of far distance dimensions $(\mathrm{d_c})$

end

end

sum all $\mathrm{dd_j}$

$\mathrm{src} = 1 - \frac{6 * \sum_{\mathrm{j}=1}\left(\mathrm{dd_j}\right)^2}{\mathrm{dd_j} * \left(\mathrm{dd_j}^2 - 1\right)^2}$, {src=Spearman's Rank Correlation Coefficient}

if src $<= 0$ and $\mathrm{d_f} > \mathrm{d_c}$

Perform $\overline{\mathrm{X}}_{\mathrm{d_f}} = \mathrm{Lb}_{\mathrm{d_f}} + \mathrm{Ub}_{\mathrm{d_f}} - \mathrm{X}_{\mathrm{d_f}}$

end

end

end

SLO is an expansion of the SO idea embedded in GWO. Processes embedded in the GWO indicate the positions of search agents based on competence (optimal value) classified according to their social hierarchy. This strategy is to produce a faster convergence rate. However, its fast convergence rate can quickly lead to getting stuck in the local optimum, causing its performance to be unstable.

Algorithm 2: Pseudocode of SLO (Dhargupta et al., 2020)

Input: initial generation t, maximum generation T, population size N, dimension d

Output: $\overline{X}_{dc}$: new position based on selective opposition

$$\text{Threshold}: \ 2 \times \left(1 - \frac{t}{T}\right)$$

for $i = 1{:}N$

if $X_i \neq X_{ibest}$

for $j = 1{:}d$

$dd_j = |X_{ibest,j} - X|$ { $dd_j =$ difference distance for each dimension}

if $dd_j <$ threshold

Identify d_c (the close distance dimensions)

Count the number of d_c $(d_c = d_c + 1)$

else

Identify d_f (the faraway distance dimensions)

Count the number d_f $(d_f = d_f + 1)$

end

end

sum all dd_j

$src = 1 - \frac{6 * \sum_{j=1}(dd_j)^2}{dd_j * (dd_j^2 - 1)^2}$, {src=Spearman's Rank Correlation Coefficient}

if src <= 0 and $d_c > d_f$

perform $\overline{X}_{d_c} = Lb_{d_c} + Ub_{d_c} - X_{d_c}$

end

end

end

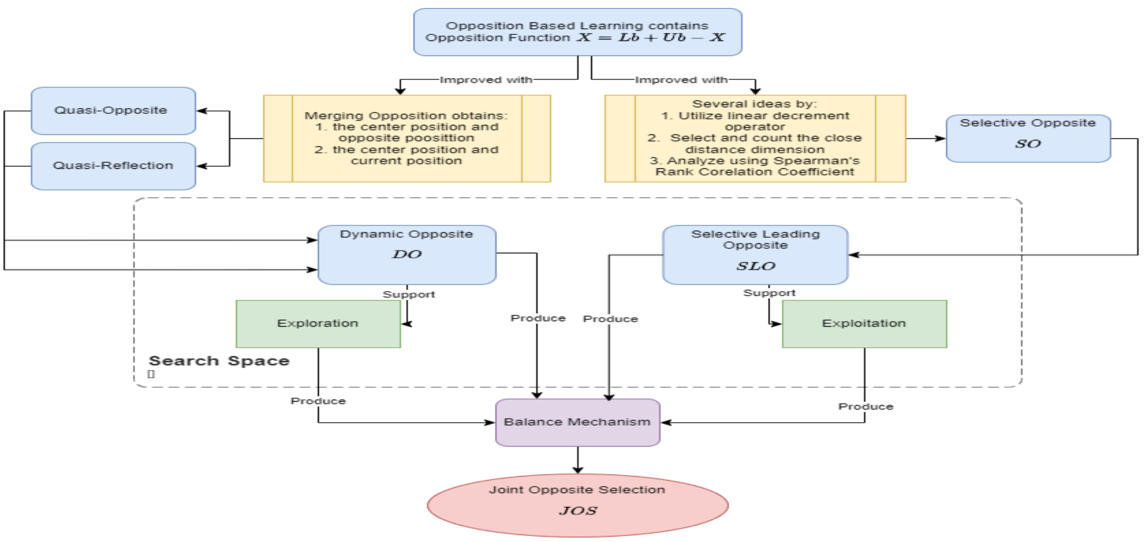

Figure 5. Hierarchical Development of Joint Opposite Selection

Algorithm 3: Pseudo Code of DO (Xu et al., 2014)

Input: Jumping rate Jr, Population size N and Position X

Output:

Stage 1: Population Initialization

Initialize position X

$D_{OBL} = Lb + Ub - X$ (D_{OBL} is the OBL strategy)

$D_{ROP} = rand \times D_{ROP}$ (D_{ROP} is the reflection opposition position)

$D_{DO} = X + rand \times (D_{DO} - X)$ (D_{DO} is the dynamic opposite)

$$D_{DO} \rightarrow X$$

Stage 2: Population Generation utilizes the Jumping rate Jr.

while nFE $<$ maxFE do

if rand $<$ Jr

$$D_{OBL} = Lb + Ub - X$$

$$D_{ROP} = rand \times D_{ROP}$$

$$D_{DO} = X + rand \times (D_{DO} - X)$$

$$D_{DO} \rightarrow X$$

end

end

Rahnamayan et al. (2008) pointed out that opposite numbers are more likely to produce a better result than normal random numbers. Supporting evidence for this theory from scientific studies (Al-Qunaieer et al., 2010, Mahdavi et al., 2018, Arini et al., 2022a) have also confirmed that the opposite strategy produces remarkable results. Rahnamayan et al. (2008), established the concept of quasi-opposition based learning, or QOBL. It uses a jumping rate and determines the midpoint of opposite points to increase the likelihood of being near the solution. Quasi-reflection was introduced by Ergezer et al. (2009) and improves the BBO success rate while requiring less fitness computation. Xu et al. (2020) suggested DO in order to enhance exploration skills and generate diversity through asymmetric search behavior by combining quasi-opposition and quasi-reflection. In order to solve multi-task optimization problems, the mutation technique is combined with the dynamic opposite to provide mutual learning (Li et al., 2021). Gonzalez (2007), proved that the balance between exploration and exploitation in the search space must be maintained for basic optimization processes. Many researchers have stated that there is no definitive formula to define the balance of exploration and exploitation in the search space and no Nature-inspired optimization algorithm to calculate this balance (Črepinšek et al., 2013, Yang et al., 2014; Morales-Castaneda et al., 2020). Additionally, Wolpert et al. (1997) pointed out that no algorithm can solve all optimization problems. Afterwards, Wang et al. (2019) also investigated whether two opposites are better than one.

Accordingly, it has been associated that DO is a subpart of exploration, SLO is a subpart of exploitation, and DO is the opposite of SLO. According to Gonzalez (2007), the opposing acts of exploration and exploitation reinforce each other. Therefore, the combination of DO and SLO were established the balance of mutual strengthening and has been named JOS (Arini, 2022a).

2.2. The proposed GWOJOS

The Matlab code of the GWOJOS algorithm has been shared on GitHub by Florentina. You can find it in reference (GitHub, 2024). The main GWOJOS is performed even though the number of functional evaluations (nFE) is less than the maximum functional evaluation ($maxFE$). After the wolves have checked their boundaries and evaluated their fitness, the JOS strategy is used. Each time the wolves evaluate their fitness, the number of functional evaluations is updated. The best value of the fitness of the wolves is the position of the prey.

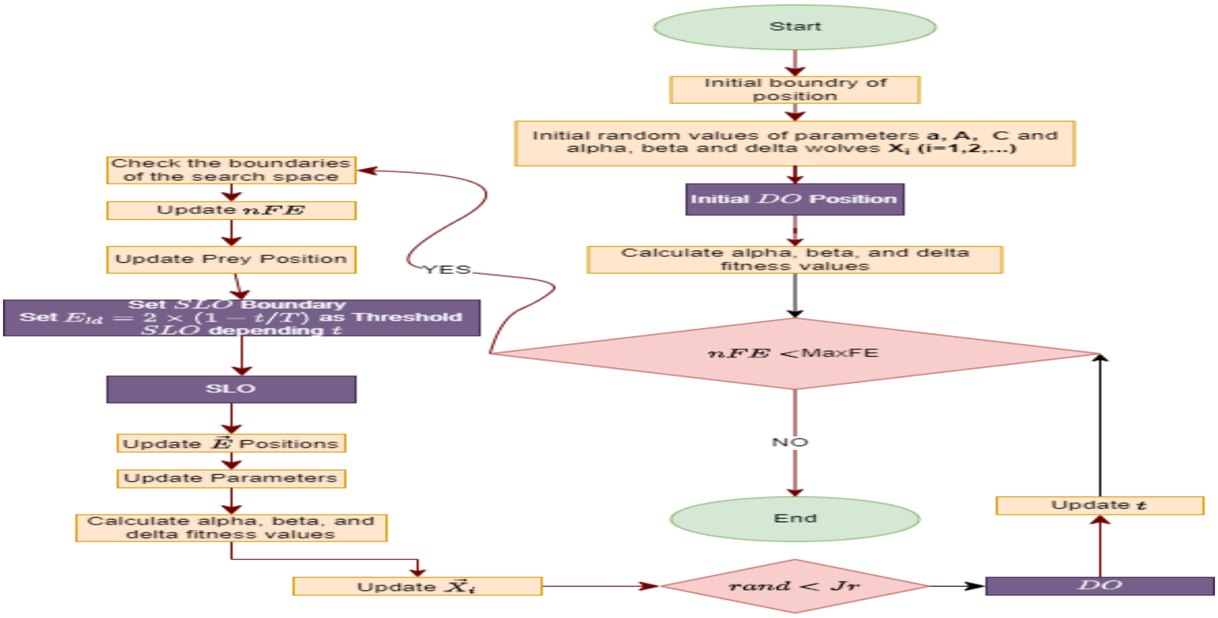

Figure 6: Flow Chart of GWOJOS

Experimental setup and experimental results: The experimental results include required comprehensive statistical analysis such as Wilcoxon and comparison assessment with other algorithms, globally.

The experiments were exhibited to solve single-objective real parameter numerical optimization of CEC 2017. The CECs are the preferable standard benchmark problem set on single-objective real parameters and required a specific standard value of the parameters to run the experiment as follows:

1. The population size (*N*) equals to 30 (*N* = 30). Note: The population size is fixed. 2. On each experiment, the maximum number of runs (*maxRun*) is 30 runs. 3. We tested on 29 benchmark functions of CEC 2017. It is noted that only the F2 function on CEC 2017 was deleted due to an unconfirmed result on the experiment. 4. Each benchmark function was tested on four numbers of variables (dimension = D): 10D, 30D, 50D and 100D, but only shown for 30D. 5. The maximum number of function evaluations (*maxFE*) is set up based on 10,000 multiply by the dimensions of 10D, 30D, 50D, and 100D. 6. For each run, the maximum number of iteration is defined by dividing *maxFE* with *N*. 7. The searching space is in the range of $[-100, 100]^D$, where the lower bound (*lb*) is -100, the upper bound (*ub*) is 100.

It is essential to examine the divergence of the proposed GWO-JOS compared to its original GWO from the population diversity. If the population is highly diverse, it means the population has difficulty converging. However, if the population has low diversity, then premature convergence might occur. The proposed JOS embedded on GWO shows the proportional balance on the diversity to avoid those occurrences.

Table 1: CEC2017 competition on single objective real parameter numerical optimization

Categories	Number of Functions	Functions	Optima, F_i^*
Unimodal Functions	f_1	Shifted and Rotated Bent Cigar Function	100
	f_3	Shifted and Rotated Zakhrov Function	300
Simple Multimodal Functions	f_4	Shifted and Rotated Rosenbrock's Function	400
	f_5	Shifted and Rotated Rastrigin's Function	500
	f_6	Shifted and Rotated Expanded Scaffer's F6 Function	600
	f_7	Shifted and Rotated Lunacek Bi_Rastringin's Function	700
	f_8	Shifted and Rotated Non-Continuous Rastringin's Function	800
	f_9	Shifted and Rotated Levy Function	900
	f_{10}	Shifted and Rotated Schwefel's Function	1000
Hybrid Functions	f_{11}	Hybrid Function 1 ($N = 3$) Zakhrov, Rosenbrock's, Rastrigin's	1100
	f_{12}	Hybrid Function 2 ($N = 3$) High-Conditioned Elliptic, Modified Schwefel's, Ben Cigar	1200
	f_{13}	Hybrid Function 3 ($N = 3$) Ben Cigar, Rosenbrock's, Lunacek Bi_Rastringin's	1300
	f_{14}	Hybrid Function 4 ($N = 4$) High-Conditioned Elliptic, Ackley, Schaffer's $F7$, Rastringin's	1400
	f_{15}	Hybrid Function 5 ($N = 4$) Ben Cigar, HGBat, Rastringin's, Rosenbrock's	1500
	f_{16}	Hybrid Function 6 ($N = 4$) Expanded Schaffer's $F6$, HGBat, Rosenbrock's, Modified Schwefel's	1600
	f_{17}	Hybrid Function 6 ($N = 5$) Katsuura, Ackley, Expanded Griewank's plus, Rosenbrock's, Schwefel's, Rastringin's	1700
	f_{18}	Hybrid Function 6 ($N = 5$) High-Conditioned Elliptic, Ackley, Rastringin's, HGBat, Discus	1800
	f_{19}	Hybrid Function 6 ($N = 5$) Bent Cigar, Rastringin's, Griewank's plus Rosenbrock's, Weierstrass, Expanded Schaffer's $F6$	1900
	f_{20}	Hybrid Function 6 ($N = 5$) HappyCat, Katsuura, Ackley, Rastringin's, Modified Schwefel's, Schaffer $F7$	2000
Composition Functions	f_{21}	Composition Function 1 ($N = 3$) Rosenbrock's, High Conditioned Elliptic, Rastringin's	2100
	f_{22}	Composition Function 1 ($N = 3$) Rastringin's, Griewank's, Modified Schwefel's	2200
	f_{23}	Composition Function 1 ($N = 4$) Rosenbrock's, Ackley, Modified Schwefel's, Rastringin's	2300
	f_{24}	Composition Function 2 ($N = 4$) Ackley, High-Conditioned Elliptic, Griewank's, Rastringin's	2400
	f_{25}	Composition Function 3 ($N = 5$) Rastringin's, HappyCat, Ackley Discus, Rosenbrock's	2500
	f_{26}	Composition Function 4 ($N = 5$) Expanded Schaffer's $F6$, Modified Schwefel's, Griewank's, Rosenbrock's, Rastringin's	2600
	f_{27}	Composition Function 5 ($N = 6$) HGBat, Rastringin's, Modified Schewel's, Bent Cigar, High-Conditioned Elliptic, Expanded Schaffer's $F6$	2700
	f_{28}	Composition Function 6 ($N = 6$) Ackley, Griewank, Discus, Rosenbrock, HappyCat, Expanded Schaffer's $F6$	2800
	f_{29}	Composition Function 7 ($N = 3$) $F15, F16, F17$	2900
	f_{30}	Composition Function 8 ($N = 3$) $F15, F18, F19$	3000

Algorithm 4: Pseudocodes of GWOJOS

Initialize N Gray Wolves Populations

Initialize the parameters a, A, C, alpha, beta and delta wolves positions and

Randomly initialize individual in DO (Algorithm 3) population of size N

Modify the $X_{\alpha,\beta,\delta}$ to X_{DO}

Initialize $nFE = 0, t = 0$ and $T =$ Maximum iteration

While $\mathrm{nFE} < \mathrm{maxFE}$ do

for $\mathrm{i} = 1{:}\,\mathrm{N}$

Return back the search agents that go beyond the boundaries of the search space

Calculate objective function for each search agent

Update nFE

Update $X_{\alpha,\beta,\delta}$

Updating boundary for opposition after every iteration

(Threshold for SLO) $\mathrm{a} = 2 \times 1 - \frac{\mathrm{t}}{\mathrm{T}})$

Continue the process by applying Algorithm 2 (Selective Leading Opposite)

Update the Position of search agents including omegas (ω)

if $\mathrm{rand} < \mathrm{Jr}$

if $\mathrm{nFE} + \mathrm{SearchAgents\_no} < \mathrm{maxFE}$

Perform Algorithm 3 (Dynamic Opposite)

Assign X_{DO} to X_{wolf}

end

end

end

The experiments were conducted considering the following points:

1. To demonstrate the effectiveness of the GWOJOS algorithm, a comparison with GWO was made below using CEC2017.

2. The candidacy of the Random Jump Strategy of the jump ratio in GWOJOS, Jr=0.25.

3. It demonstrated the successful JOS behavior and exploration ability.

4. The statistical analysis of the Wilcoxon Signed Rank Test performance of GWOJOS compared to its competitors' algorithms was presented.

3 Main Result

3.1. Experimental Results of Benchmark CEC2017

Table 2. Comparison of GWO and GWOJOS performance using CEC2017 Benchmark for 30 dimensions.

F	Ist. Anl	GWO	GWOJOS	F	GWO	GWOJOS	F	GWO	GWOJOS
	Min	9.86E+07	8.17E+05		5.80E+05	1.10E+06		2.40E+03	2.30E+03
	Max	5.65E+09	3.05E+07		4.58E+08	4.82E+07		9.38E+03	2.35E+03
F_1	Std	1.33E+09	7.66E+06	F_{12}	1.12E+08	1.11E+07	F_{22}	2.05E+03	9.74E+00
	Media	1.54E+09	1.03E+07		2.91E+07	1.09E+07		5.32E+03	2.32E+03
	Mean	1.97E+09	1.17E+07		7.06E+07	1.35E+07		5.00E+03	2.32E+03
	Min	1.38E+04	8.61E+02		4.49E+04	1.89E+04		2.70E+03	2.69E+03
	Max	5.24E+04	4.46E+03		4.34E+08	7.82E+05		2.89E+03	2.88E+03
F_3	Std	1.03E+04	9.68E+02	F_{13}	1.03E+08	1.83E+05	F_{23}	3.54E+01	4.42E+01
	Media	3.03E+04	2.04E+03		7.58E+04	8.27E+04		2.75E+03	2.73E+03

	Mean	3.01E+04	2.17E+03		3.49E+07	1.47E+05		2.76E+03	2.74E+03
	Min	5.00E+02	4.76E+02		3.38E+03	2.49E+03		2.86E+03	2.85E+03
	Max	8.10E+02	5.31E+02		5.02E+05	1.07E+05		3.05E+03	3.02E+03
F_4	Std	7.21E+01	1.59E+01	F_{14}	1.71E+05	2.84E+04	F_{24}	3.44E+01	4.33E+01
	Media	5.78E+02	5.10E+02		5.56E+04	2.35E+04		2.93E+03	2.89E+03
	Mean	6.00E+02	5.09E+02		1.36E+05	3.07E+04		2.93E+03	2.90E+03
	Min	5.55E+02	5.37E+02		1.08E+04	8.19E+03		2.92E+03	2.88E+03
	Max	6.43E+02	6.52E+02		3.42E+06	5.13E+04		3.08E+03	2.96E+03
F_5	Std	1.96E+01	2.06E+01	F_{15}	1.06E+06	1.03E+04	F_{25}	3.89E+01	2.05E+01
	Media	6.06E+02	5.76E+02		3.58E+04	2.21E+04		2.98E+03	2.91E+03
	Mean	6.05E+02	5.76E+02		4.80E+05	2.40E+04		2.99E+03	2.91E+03
	Min	6.02E+02	6.00E+02		2.10E+03	1.82E+03		3.26E+03	2.81E+03
	Max	6.21E+02	6.06E+02		2.98E+03	2.82E+03		5.72E+03	4.65E+03
F_6	Std	4.00E+00	1.78E+00	F_{16}	2.39E+02	2.51E+02	F_{27}	4.51E+02	5.98E+02
	Media	6.08E+02	6.02E+02		2.51E+03	2.34E+03		4.72E+03	2.92E+03
	Mean	6.09E+02	6.02E+02		2.48E+03	2.29E+03		4.70E+03	3.20E+03
	Min	7.95E+02	7.66E+02		1.79E+03	1.77E+03		3.21E+03	3.20E+03
	Max	1.02E+03	9.72E+02		2.34E+03	2.43E+03		3.32E+03	3.25E+03
F_7	Std	5.44E+01	6.28E+01	F_{17}	1.32E+02	1.45E+02	F_{27}	2.73E+01	1.23E+01
	Media	8.62E+02	8.00E+02		1.95E+03	1.90E+03		3.24E+03	3.23E+03
	Mean	8.78E+02	8.24E+02		1.97E+03	1.94E+03		3.25E+03	3.22E+03
	Min	8.53E+02	8.43E+02		3.91E+04	3.41E+04		3.27E+03	3.21E+03
	Max	9.61E+02	8.89E+02		9.84E+06	6.45E+05		4.02E+03	3.30E+03
F_8	Std	2.28E+01	1.20E+01	F_{18}	1.76E+06	1.85E+05	F_{28}	1.34E+02	2.42E+01
	Media	8.96E+02	8.71E+02		4.48E+05	1.21E+05		3.40E+03	3.23E+03
	Mean	8.97E+02	8.69E+02		8.19E+05	1.96E+05		3.42E+03	3.24E+03
	Min	1.14E+03	9.11E+02		7.96E+03	1.55E+04		3.49E+03	3.46E+03
	Max	3.97E+03	4.30E+03		3.34E+06	2.46E+06		4.13E+03	3.95E+03
F_9	Std	6.89E+02	6.56E+02	F_{19}	7.13E+05	5.52E+05	F_{29}	1.58E+02	1.18E+02
	Media	1.73E+03	1.01E+03		1.75E+05	3.12E+05		3.70E+03	3.59E+03
	Mean	1.90E+03	1.20E+03		4.76E+05	5.16E+05		3.73E+03	3.62E+03
	Min	3.01E+03	2.66E+03		2.15E+03	2.17E+03		4.33E+05	1.63E+05
	Max	5.94E+03	8.15E+03		2.55E+03	2.61E+03		4.09E+07	1.17E+07
F_{10}	Std	6.23E+02	1.14E+03	F_{20}	1.16E+02	1.21E+02	F_{30}	9.70E+06	2.81E+06
	Media	4.18E+03	3.68E+03		2.35E+03	2.30E+03		4.87E+06	2.53E+06
	Mean	4.24E+03	3.88E+03		2.36E+03	2.33E+03		7.72E+06	3.35E+06
	Min	1.29E+03	1.15E+03		2.36E+03	2.34E+03			
	Max	4.23E+03	1.32E+03		2.44E+03	2.48E+03			
F_{11}	Std	8.26E+02	4.06E+01	F_{21}	1.93E+01	2.53E+01			
	Media	1.62E+03	1.23E+03		2.39E+03	2.36E+03			
	Mean	1.91E+03	1.23E+03		2.39E+03	2.36E+03			

The experimental results in the Table 2 show a comparison among the proposed algorithm (GWO-JOS) and GWOfor 30 dimension. The total functions exhibited in Table 3 are evaluated in 10, 30, 50 and 100 dimensions on 29 benchmark functions of CEC 2017. The experiment concludes from the mean of 30 runs which were experimented successively according to each function. The sign (+) at the bottom of Table 3 indicates that the development of opposition is better than the original and this positive indication are also highlighted. The sign (-) denotes the value of objective function is slightly higher than the original GWO.

The Wilcoxon signed-rank test was used to compare the performance of GWOJOS vs. GWO across multiple benchmark functions (F1–F30) at different dimensional settings (10, 30, 50, and 100). The test provides insight into whether the two algorithms have statistically significant performance differences. The following can be mentioned as key observations.

- Statistically Significant Results ($p < 0.05$) → GWOJOS is Better: In most cases, the p-value is below 0.05, indicating that the difference between the two algorithms is statistically significant.. The positive ranks (Rank +) are significantly higher than the negative ranks (Rank -) in these cases, favoring GWOJOS over GWO. This suggests that GWOJOS consistently outperforms GWO in optimizing the given benchmark functions.

- No Significant Difference ($p \geq 0.05$) → Similar Performance: Some test cases (e.g., F4, F5, F13, F16, F17, F19, and F20) show p-values greater than 0.05, meaning there is no statistically significant difference between GWOJOS and GWO. In these cases, both algorithms exhibit comparable performance, and the improvements by GWOJOS are not large enough to be considered statistically meaningful.

- Rare Cases Where GWO Outperforms GWOJOS: F9 (50D, 100D) shows a negative winner, indicating that GWO outperforms GWOJOS in these specific settings. This suggests that for certain high-dimensional functions, GWO may still be a viable or even better option.

GWOJOS is generally superior to GWO in terms of optimization performance, as it wins in most test cases. However, there are a few benchmark functions where GWO still performs comparably or better, especially in higher-dimensional settings. The Wilcoxon test results provide strong statistical evidence supporting the advantages of GWOJOS.

Table 3: Scalability Analysis

F	GWOJOS vs.GWO						
	Dim	**P-Value**	**z-Value**	**Rank (-)**	**Rank (+)**	**Statistically significant**	**Winner**
F_1	10	0.135908	-1.4912	305	160	0 (p≥0.05)	=
	30	0.000002	-4.7821	465	0	1 (p < 0.05)	+
	50	0.000002	-4.7821	465	0	1 (p < 0.05)	+
	100	0.000002	-4.7821	465	0	1 (p < 0.05)	+
F_3	10	0.000002	-4.7821	465	0	1 (p < 0.05)	+
	30	0.000002	-4.7821	465	0	1 (p < 0.05)	+
	50	0.000002	-4.7821	465	0	1 (p < 0.05)	+
	100	0.000002	-4.7821	465	0	1 (p < 0.05)	+
F_4	10	0.628843	-0.4834	256	209	0 (p≥0.05)	=
	30	0.000004	-4.6176	457	8	1 (p < 0.05)	+
	50	0.000002	-4.7821	465	0	1 (p < 0.05)	+
	100	0.000002	-4.7821	465	0	1 (p < 0.05)	+
F_5	10	0.530440	-0.6273	263	202	0 (p≥0.05)	=
	30	0.000053	-4.0417	429	36	1 (p < 0.05)	+
	50	0.000009	-4.4325	448	17	1 (p < 0.05)	+
	100	0.000002	-4.7821	465	0	1 (p < 0.05)	+
F_6	10	0.013194	-2.4785	253	112	1 (p < 0.05)	+
	30	0.000004	-4.6382	458	7	1 (p < 0.05)	+
	50	0.000002	-4.7410	463	2	1 (p < 0.05)	+
	100	0.000005	-4.5765	455	10	1 (p < 0.05)	+
F_7	10	0.001036	-3.2807	392	73	1 (p < 0.05)	+
	30	0.001484	-3.1778	387	78	1 (p < 0.05)	+

	50	0.000009	-4.4325	448	17	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_8	10	0.097772	-1.6558	313	152	0 ($p \geq 0.05$)	=
	30	0.000024	-4.2268	438	27	1 ($p < 0.05$)	+
	50	0.000010	-4.4119	447	18	1 ($p < 0.05$)	+
	100	0.000002	-4.7204	462	3	1 ($p < 0.05$)	+
F_9	10	0.000003	-4.6587	459	6	1 ($p < 0.05$)	+
	30	0.000148	-3.7949	417	48	1 ($p < 0.05$)	+
	50	0.909931	0.1131	227	238	0 ($p \geq 0.05$)	=
	100	0.000716	0.000716	68	397	0 ($p \geq 0.05$)	-
F_{10}	10	0.042767	-2.0260	331	134	1 ($p < 0.05$)	+
	30	0.003379	-2.9310	375	90	1 ($p < 0.05$)	+
	50	0.000106	-3.8771	421	44	1 ($p < 0.05$)	+
	100	0.000031	-4.1651	435	30	1 ($p < 0.05$)	+
F_{11}	10	0.001382	-3.1984	388	77	1 ($p < 0.05$)	+
	30	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	50	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_{12}	10	0.198610	-1.2855	295	170	0 ($p \geq 0.05$)	=
	30	0.005667	-2.7664	367	98	1 ($p < 0.05$)	+
	50	0.000004	-4.6176	457	8	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_{13}	10	0.477947	-0.7096	267	198	0 ($p \geq 0.05$)	=
	30	0.428430	-0.7919	271	194	0 ($p \geq 0.05$)	=
	50	0.000004	-4.6382	458	7	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_{14}	10	0.130592	-1.5118	306	159	0 ($p \geq 0.05$)	=
	30	0.002957	-2.9721	377	88	1 ($p < 0.05$)	+
	50	0.000053	-4.0417	429	36	1 ($p < 0.05$)	+
	100	0.000002	-4.7204	462	3	1 ($p < 0.05$)	+
F_{15}	10	0.000205	-3.7126	413	52	1 ($p < 0.05$)	+
	30	0.000160	-3.7743	416	49	1 ($p < 0.05$)	+
	50	0.000420	-3.5275	404	61	1 ($p < 0.05$)	+
	100	0.000002	-4.7204	462	3	1 ($p < 0.05$)	+
F_{16}	10	0.349346	-0.9359	278	187	0 ($p \geq 0.05$)	=
	30	0.027029	-2.2111	340	125	1 ($p < 0.05$)	+
	50	0.007731	-2.6636	362	103	1 ($p < 0.05$)	+
	100	0.000003	-4.6793	460	5	1 ($p < 0.05$)	+
F_{17}	10	0.599936	-0.5245	258	207	0 ($p \geq 0.05$)	=
	30	0.298944	-1.0387	283	182	0 ($p \geq 0.05$)	=
	50	0.040702	-2.0465	332	133	1 ($p < 0.05$)	+
	100	0.000420	-3.5275	404	61	1 ($p < 0.05$)	+
F_{18}	10	0.125438	-1.5323	307	158	0 ($p \geq 0.05$)	=
	30	0.000664	-3.4041	398	67	1 ($p < 0.05$)	+
	50	0.002585	-3.0133	379	86	1 ($p < 0.05$)	+
	100	0.000022	-4.2474	439	26	1 ($p < 0.05$)	+
F_{19}	10	0.416534	-0.8124	272	193	0 ($p \geq 0.05$)	=
	30	0.557743	0.5862	204	261	0 ($p \geq 0.05$)	=
	50	0.003162	-2.9516	376	89	1 ($p < 0.05$)	+
	100	0.000002	-4.7410	463	2	1 ($p < 0.05$)	+
F_{20}	10	0.465283	0.7302	197	268	0 ($p \geq 0.05$)	=
	30	0.530440	-0.6273	263	202	0 ($p \geq 0.05$)	=
	50	0.044919	-2.0054	330	135	1 ($p < 0.05$)	+
	100	0.097772	-1.6558	313	152	0 ($p \geq 0.05$)	=
F_{21}	10	0.000049	-4.0622	430	35	1 ($p < 0.05$)	+

	30	0.000082	-3.9388	424	41	1 ($p < 0.05$)	+
	50	0.000002	-4.7616	464	1	1 ($p < 0.05$)	+
	100	0.000002	-4.7616	464	1	1 ($p < 0.05$)	+
F_{22}	10	0.280214	-1.0798	285	180	0 ($p \geq 0.05$)	=
	30	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	50	0.000012	-4.3708	445	20	1 ($p < 0.05$)	+
	100	0.000016	-4.3091	442	23	1 ($p < 0.05$)	+
F_{23}	10	0.236936	-1.1827	290	175	0 ($p \geq 0.05$)	=
	30	0.007271	-2.6842	363	102	1 ($p < 0.05$)	+
	50	0.000616	-3.4246	399	66	1 ($p < 0.05$)	+
	100	0.000002	-4.7616	464	1	1 ($p < 0.05$)	+
F_{24}	10	0.000031	-4.1651	435	30	1 ($p < 0.05$)	+
	30	0.013194	-2.4785	353	112	1 ($p < 0.05$)	+
	50	0.002585	-3.0133	379	86	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_{25}	10	0.004390	-2.8487	371	94	1 ($p < 0.05$)	+
	30	0.000002	-4.7204	462	3	1 ($p < 0.05$)	+
	50	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_{26}	10	0.001197	-3.2395	390	75	1 ($p < 0.05$)	+
	30	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	50	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	100	0.000002	-4.7410	463	2	1 ($p < 0.05$)	+
F_{27}	10	0.002105	-3.0750	382	83	1 ($p < 0.05$)	+
	30	0.000136	-3.8154	418	47	1 ($p < 0.05$)	+
	50	0.000005	-4.5559	454	11	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_{28}	10	0.017518	-2.3756	348	117	1 ($p < 0.05$)	+
	30	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	50	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
	100	0.000002	-4.7821	465	0	1 ($p < 0.05$)	+
F_{29}	10	0.007271	-2.6842	363	102	1 ($p < 0.05$)	+
	30	0.013194	-2.4785	353	112	1 ($p < 0.05$)	+
	50	0.002765	-2.9927	378	87	1 ($p < 0.05$)	+
	100	0.000034	-4.1445	434	31	1 ($p < 0.05$)	+
F_{30}	10	0.020671	-2.3139	345	120	1 ($p < 0.05$)	+
	30	0.007271	-2.6842	363	102	1 ($p < 0.05$)	+
	50	0.001593	-3.1572	386	79	1 ($p < 0.05$)	+
	100	0.000174	-3.7537	415	50	1 ($p < 0.05$)	+

Diversity analysis and exploration-exploitation analysis of the GWOJOS algorithm for CEC2017 benchmarks are displayed in Figure 7. These analyses have shown how important the exploration-exploitation balance is. Additionally, the experimental results show that the effect of the JOS strategy on the exploration-exploitation balance. This is consistent with the purpose of the study.

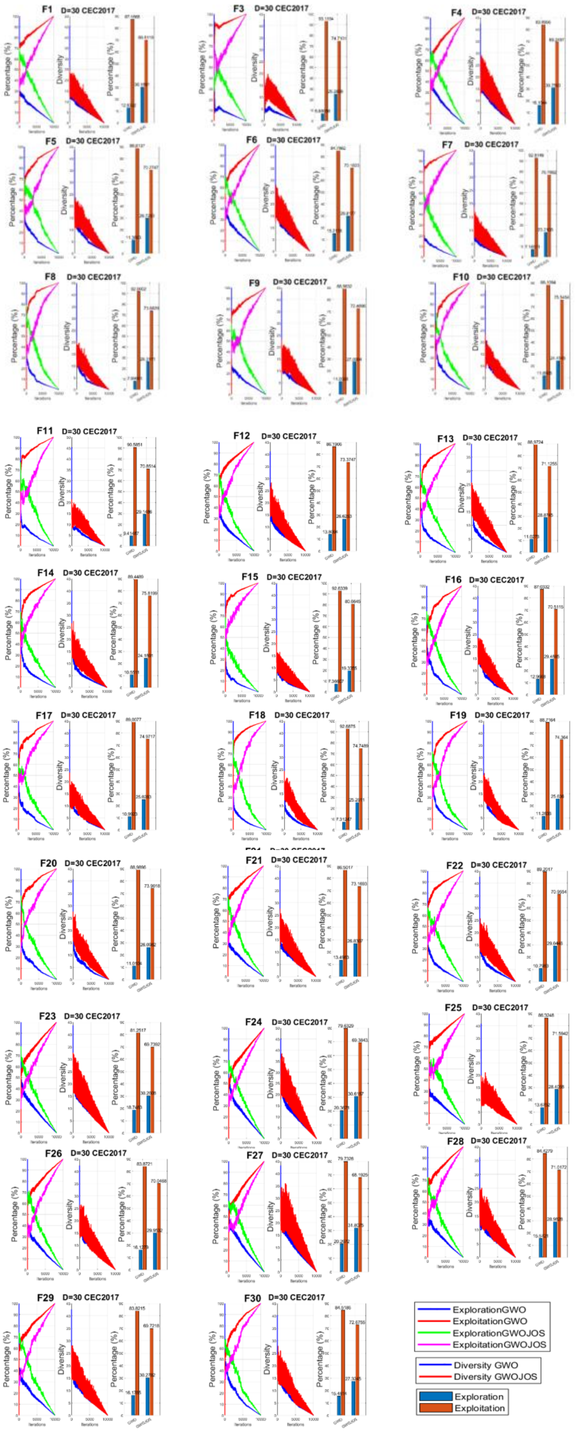

Figure 7: Exploration-Exploitation and diversity analysis of GWO and GWOJOS algorithms

Fitness values, fitness average value and box plot representations of GWOJOS and GWO algorithms for 30 runs are shown in Figure 8. From this analysis, the superiority of GWOJOS over GWO is seen.

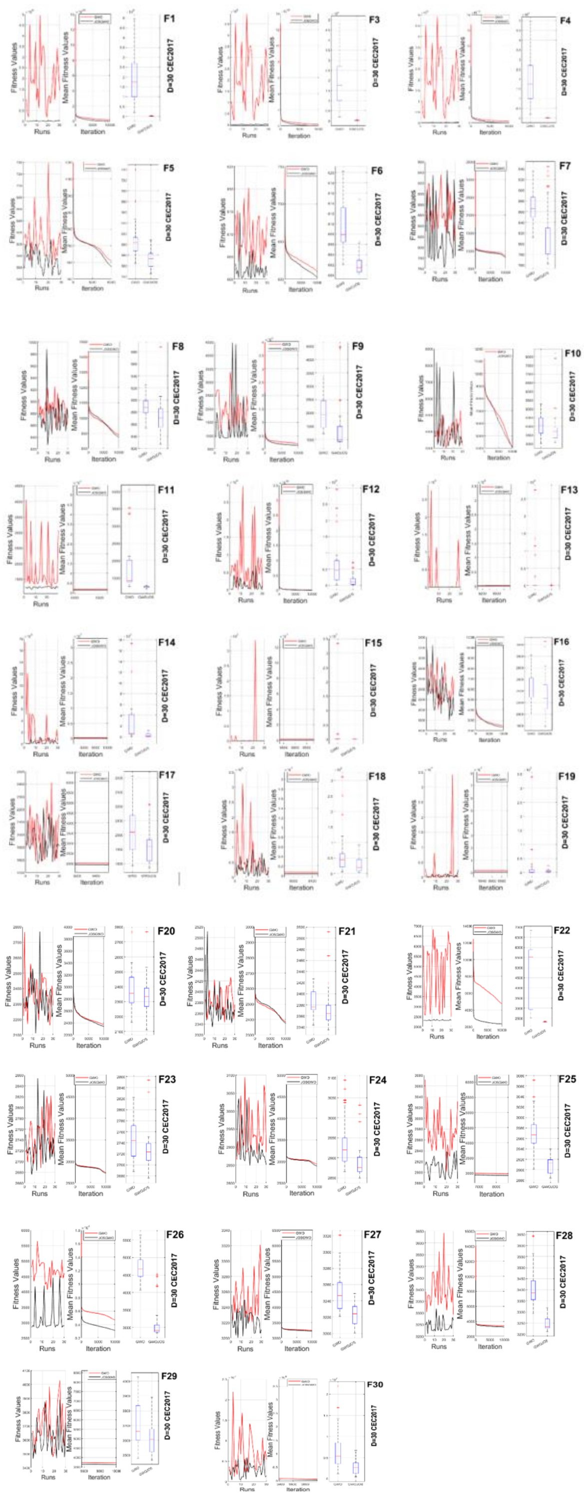

Figure 8: Convergence and stability analysis of GWO and GWOJOS.

4. Speed Control Development

4. 1. Modeling of DC Motor System

This section introduces a DC motor setup consisting of both a mechanical load and a DC motor. The main goal is to effectively regulate the motor's speed and torque through the implementation of a control system. The equivalent circuit for this specific type of DC motor is illustrated in Figure 9.

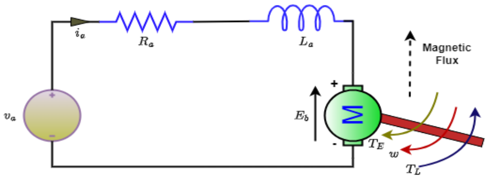

Figure 9. Equivalent circuit of DC motor.

This system is regarded as a linear system and the mechanical stress is represented as a constant torque (τ_L) to create a mathematical model. The speed of the DC motor is controlled by regulating the armature voltage $v_a(t)$. This produces an electromechanical force while armature current $i_a(t)$ adjusts proportionally to the rotational speed (İzci and Ekinci, 2023). To model the DC motor, the following differential expressions characterizing the motor's speed and torque dynamics are provided:

$$v_a(t) = i_a(t)R_a + L_a\frac{di(t)}{dt} + E_b \qquad (1)$$

while the flux remains constant, the induced voltage E_b in the motor is linearly proportional to angular velocity ω as follows

$$E_b = K_b\frac{d\theta(t)}{dt} = K_b\,\omega(t) \qquad (2)$$

A total torque consists of the impact of the inertia and fractional torques which is given by

$$T_E - T_L = J\frac{d\omega(t)}{dt} + B\,\omega(t) = K_m i_a(t) \quad (3)$$

where R_a and L_a are the resistance and inductance of the DC motor respectively. E_b is the back electromotive force, K_b is the constant, θ is the angular velocity, ω is the motor shaft velocity, T_E , T_L are the electric and load torques respectively, J indicates the motor's moment of inertia. B and K_m are frictional and torque constants respectively. Applying Laplace transform to equations (1-3) (with zero initial conditions) which leads to

$$\mathrm{v(s) = (Ls + R_a)i_a(s) + E_b(s)} \qquad (4)$$

$$\mathrm{E_b(s) = K_b\omega(s)} \qquad (5)$$

$$\mathrm{T_E(s) - T_L(s) = (Js + b)\omega(s) = K_m i_a(s)} \qquad (6)$$

Simplifying equations (4) and (6) results in

$$\mathrm{i(s) = \frac{v(s) - K_b\omega(s)}{L_a s + R_a}} \qquad (7)$$

$$\mathrm{\omega(s) = \frac{T_E(s) - T_L(s)}{Js + B} = \frac{K_m}{Js + B} i_a(s)} \qquad (8)$$

The DC motor’s transfer function can be expressed as follows:

$$\mathrm{G_p(s) = \frac{\omega(s)}{v(s)} = \frac{K_m}{(L_a s + R_a)(Js + B) + K_b K_m},\ T_L(s) = 0} \qquad (9)$$

4. 2. NIOPID Controller

Non-integer (fractional) calculus is a generalization of integration and differentiation to non-integer order fundamental operator ${}_a^tD^r$ where a and t are the limits and ($r \in R$) is the order of the operation. The two definitions used for the fractional differential ${}_a^tD^r$ are the Grünwald-Letnikov (GL) definition and the Riemann-Liouville (RL) definition (Xue et al., 2006). Further, it has been mentioned in the literature that for a wide class of functions, these two definitions are equivalent (Xue et al., 2006).

Then, the fractional PID controlller is writen as:

$$\mathrm{G_{NIOPID}(s) = K_p + K_I s^{-\lambda} + K_D s^{\mu}, (\lambda, \mu > 0)} \quad . \qquad (10)$$

in which K_p, K_i and K_d are proportional, integral and derivative gains when λ and μ denote fractional integral and derivative orders respectively. Besides, a block diagram of a NIOPID-controlled DC motor system is displayed in Figure 10. The NIOPID controller can efficiently regulate the speed of DC motors as it has a more flexible control structure for the stabilization of dynamic systems. In addition to PID control, the NIOPID controller has fractional order terms (λ and μ) (İzci et al., 2023). Selection of λ, μ gives the classical controllers PID controller ($\lambda, \mu = 1$).

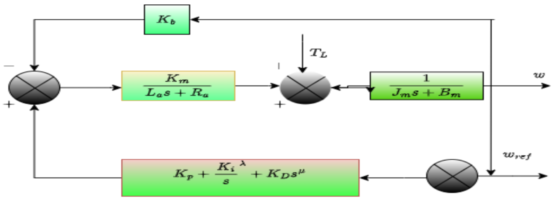

Figure 10. Diagram showcasing the NIOPID control implementation in the DC motor system

The NIOPID-controlled DC motor's closed-loop transfer function is provided as follows:

$$G_{cl}(s) = \frac{\omega(s)}{\omega_{ref}(s)} = \frac{G_{NIOPID}(s) \times G_P(s)}{1 + G_{NIOPID}(s) \times G_p(s)}, T_L = 0 \quad (11)$$

Substituting $G_P(s)$ and $G_{NIOPID}(s)$ into equation (11), one has

$$G_{cl} = \frac{K_m(K_p + K_i s^{-\lambda} + K_d s^{\mu})}{[\aleph_1 + \aleph_2]} \quad (12)$$

where $\aleph_1 = (Js + B)(L_a s + Ra) + K_b K_m$ and $\aleph_2 = K_m(K_p + K_i s^{-\lambda} + K_d s^{\mu})$

4. 3. Objective Function

The problem of DC motor speed regulation is considered a minimization problem treated by the GWOJOS. The following procedures define the related system as an optimization problem. Then, the NIOPID controller's settings will be ideal. In the first place, the problem's dimension is shown as $[x_1, .., x_5] = [K_p\ K_i\ K_d\ \ \lambda\ \ \mu]$ and the objective function, $F(\vec{K})$ (İzci and Ekinci, 2023) for the corresponding minimization problem is given as:

$$F(\vec{K}) = (1 - e^{-\sigma}) \times \left(E_{ss} + \frac{M_p}{100}\right) + e^{-\sigma} \times (t_{ST} - t_{RT}) \quad (13)$$

where σ is a balancing coefficient ($\sigma = 1$, in this paper), E_{ss} represents the steady-state error, M_p denotes the overshoot, t_{ST} signifies the settling period, and t_{RT} refers to the rise period. The limits of parameters are $0.001 \leq K_p,\ K_i\ K_d \leq 20$ and $0 \leq \lambda, \mu \leq 2$. These limits are identical to (Tepljakov and Tepljakov, 2017, Hekimoğlu, 2019, İzci et al., 2021, İzci and Ekinci, 2023, Ayinla et al., 2024, Sarma and Bardalai, 2024). Figure 11 shows a block schematic of the suggested approach to design the parameters of the NIOPID control scheme for the direct-current powered motor systems.

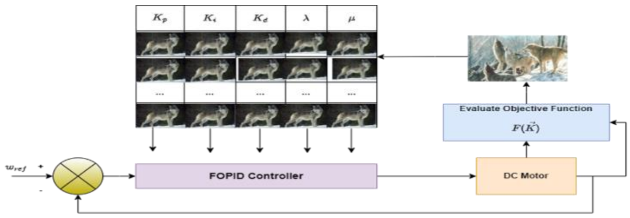

Figure 11. Schematic of NIOPID control tuning procedure for the DC motor system with the GWOJOS

4. 4. Statistical Analysis

This section evaluates the statistical performance of the GWOJOS. Figure 12 displays the curve of the objective function for the best run at each iteration and the best fitness values found for each run. Also, a boxplot illustrating the distribution of objective function values produced by algorithms is presented in Figure 12. The best run of the optimization process yields the following controller parameters: with GWOJOS, K_p =20, K_i =20, K_d =14.4397, $\gamma =$ 0.8181 and $\mu =$ 0.9988. Figure 13 shows the alteration of control parameters. This graphic aids in our comprehension of how the controller's settings vary throughout the optimization procedure. Figure 14 shows the best objective function values obtained over 30 runs. It is clear to see that the GWOJOS is significantly superior to other optimizers. The GWOJOS provides a fast convergence rate and the quality of the solution.

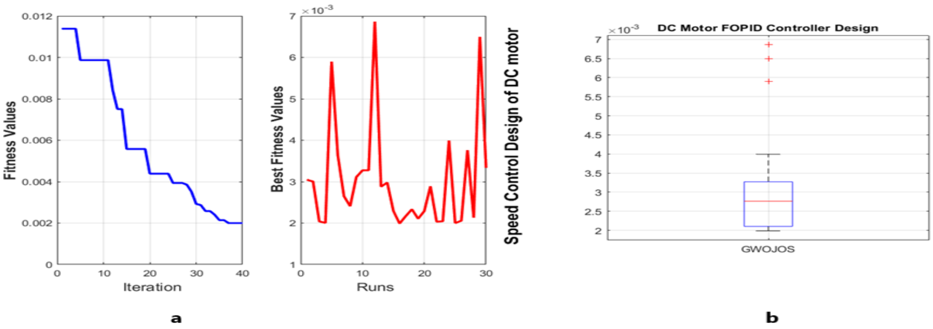

Figure 12. The convergence trends, best fitness value at each run (a), and boxplot (b) achieved by the GWOJOS algorithm.

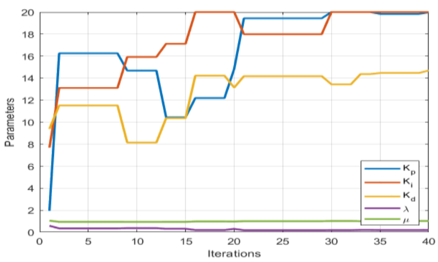

Figure 13. The varying of the NIOPID controller's parameters over the iterations with the GWOJOS algorithm

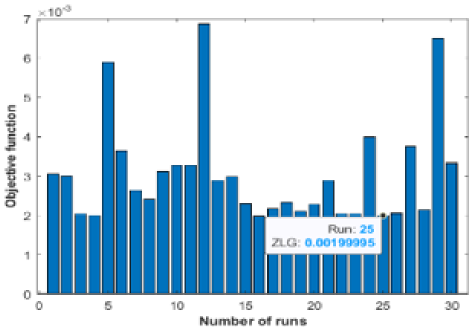

Figure 14. Best objective function values obtained from all independent runs of the GWOJOS algorithm

5. Simulation Results and Discussion

This section presents the simulation results of the developed controller. All simulations are conducted on MATLAB/Simulink software installed on a personal computer with an Intel ® core i5 processor at 2.4 GHz and 8 GB RAM. The FOMCON toolbox is employed to obtain a non-integer order PID controller. The closed-loop responses in terms of time and frequency domains are shown in Figures 15 and 16, respectively. The specifications of the closed-loop system in time and frequency domains are given in Table 4.

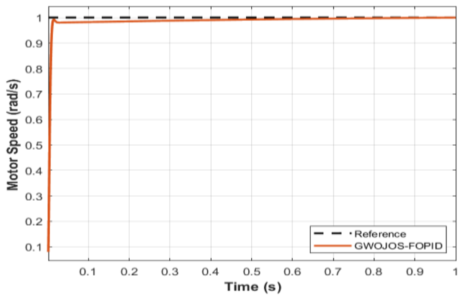

Figure15: Step response of the GWOJOS-NIOPID controlled system

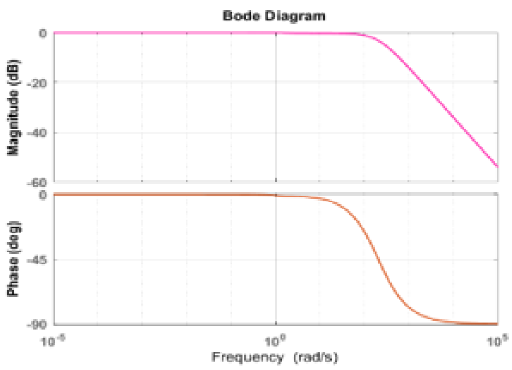

Figure 16: Comparison of Bode plot with NIOPID controllers

Table 4: Evaluation of performance regarding time and frequency response characteristics

Alg.	T_r	T_s	OS (%)	Ess(%)	Gain Margin (dB)	Phase Margin (deg)
GWOJOS (Proposed)	0.0071098	0.012091	0	0.026468	∞	178.9590

5. 1. Comparison with Recently Developed Methods

To verify the effectiveness of the proposed GWOJOS-NIOPID controller, this subsection performs comparisons using the recently developed methods such as SMA-NIOPID, OBLSMASA- NIOPID (Tepljakov and Tepljakov, 2017), ASO-NIOPID (Ayinla et al., 2024), GWO-NIOPID (Hekimoğlu, 2019). Figure 18 compares the closed-loop responses of DC motor with different controllers. To demonstrate the superiority of the GWOJOS-NIOPID controllers over other approaches documented in the literature, we present the results of a performance analysis focusing on time-domain features in Table 5.

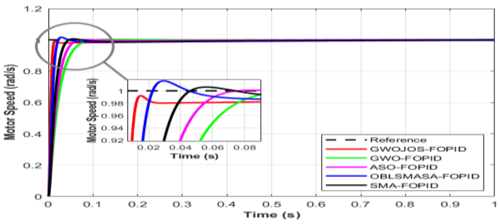

Figure 17. Comparison of step response dynamics between the proposed method and other methodologies.

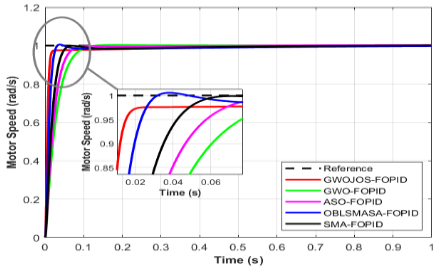

Figure 18. Comparison of closed-loop responses in the DC motor for Case I.

Robustness Analysis

The robustness analysis was performed by varying the electrical resistance (R_a) of the DC motor with ±25% and torque constant (K_m) with ±20% separately. This leads to four different testing cases. The closed-loop step responses for all cases are shown in Figures 18, 19, 20 and 21. Despite varying parameters in the DC motor system, the proposed GWOJOS-NIOPID controller provides a satisfying performance over the other controllers. Table 6 compares results achieved by PID and GWOJOS-NIOPID controllers for the time-domain performance assessment.

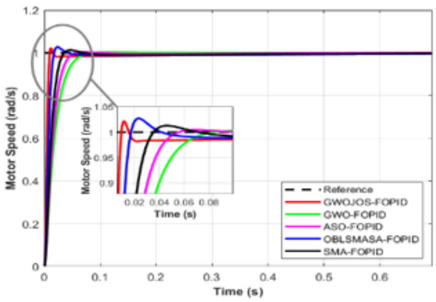

Figure 19. Comparison of closed-loop responses in the DC motor for Case II.

Table 5. Performance analysis in case of different scenarios

CASES	Metrics	GWOJOS	GWO	ASO	OBLSMASA	SMA
Nominal Case	T_r	0.0071098	0.043995	0.033151	0.012688	0.024175
	T_s	0.012091	0.075312	0.055502	0.019787	0.038442
	OS (%)	0	0.30006	0	1.8925	0.60724
	Ess(%)	0.026468	0.0091974	0.12963	0.27943	0.0031573
Case I: R_a=0.30 and K_m=0.012	T_r	0.009831	0.055929	0.042453	0.016461	0.03101
	T_s	0.16124	0.095608	0.072934	0.026277	0.05086
	OS (%)	0	0.24162	0	0.81883	0
	Ess(%)	0.045856	0.12579	0.27829	0.2535	0.11581
Case II: R_a=0.30 and K_m=0.018	T_r	0.0057112	0.036032	0.027055	0.010264	0.019705
	T_s	0.012445	0.061703	0.044593	0.029981	0.030878
	OS (%)	2.1888	0.37418	0.33004	2.9356	1.2651
	Ess(%)	0.045856	0.12579	0.12192	0.21698	0.019382
Case III: R_a=0.50 and K_m=0.012	T_r	0.0098104	0.055918	0.042431	0.016429	0.030965
	T_s	0.15189	0.095845	0.073016	0.026189	0.050757
	OS (%)	0	0.20236	0	0.86782	0
	Ess(%)	0.056653	0.034993	0.14006	0.37494	0.023642
Case IV: R_a=50 and K_m=0.018	T_r	0.0057059	0.03603	0.027047	0.010254	0.01969
	T_s	0.012534	0.061808	0.044617	0.030134	0.030851
	OS (%)	2.2274	0.34511	0.31307	2.9676	1.273
	Ess(%)	0.076308	0.09864	0.030561	0.2981	0.07315

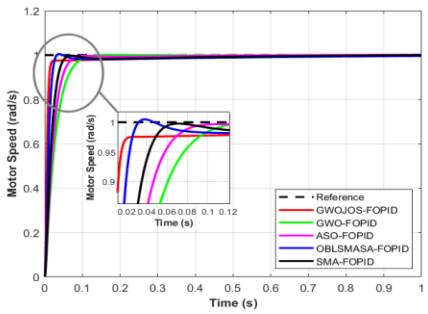

Figure 20. Comparison of closed-loop responses in the DC motor for Case III.

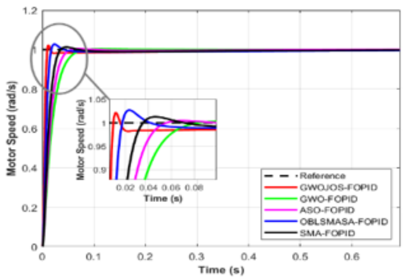

Figure 21. Comparison of closed-loop responses in the DC motor for Case IV.

Table 6. Time domain performance analysis of direct current motor system employing various PID controllers.

Adjustment Method	**Transient response and quality indicators**			
	Maximum Overshoot (%)	**Rise time (Sec) (0:1 0:9)**	**Settling Time (sec) (62%)**	**ZLG**
GWOJOS-NIOPID	0	**0.0071098**	**0.012091**	**0.0020**
Levy flight distribution with Nelder–Mead algorithm baed Proportional integral-derivative (PID) (Izci, 2021)	0	0.0462	0.0813	0.0129
Harris-hawks optimization based PID (Ekinci et al., 2020)	0	0.0568	0.1003	0.0160
Henry gas solubility optimization based PID (Ekinci et al., 2021)	0	0.0684	0.1186	0.0185
Slime mould algorithm based PID (Izci and Ekinci, 2021)	0	0.0491	0.0857	0.0135
Atom search optimization based PID (Hekimoğlu, 2019)	0	0.0692	0.1535	0.0310
Grey wolf optimization based PID (Agarwal et al., 2018)	1.5062	0.1388	0.2052	0.0340
Stochastic fractal search algorithm based PID (Bhatt et al., 2019)	0	0.5436	1.4475	0.3325
Kidney-inspired algorithm-based PID (Hekimoğlu, 2019)	0	0.0445	0.0922	0.0176
Invasive weed optimization algorithm based PID (Khalilpour et al., 2011)	6.9759	0.4189	1.2533	0.3511
Particle swarm optimization based PID (Khalilpour et al., 2011)	24.2406	0.3560	1.8028	0.6855

6. Conclusion

The Joint Opposite Selection operator consists of Selective Leading Opposition and Dynamic Opposite methods. In a particular search space, the DO and SLO enhance the balance of exploration to exploitation, respectively. SLO calculates the search agents' close distance dimension using a threshold value that decreases linearly. DO gives search agents opportunities to develop their capabilities in the search domain. The JOS operator is combined with the GWO algorithm, which is widely used in the literature. Although GWO is widely used, it has disadvantages in finding the best results. This combination has improved the performance of GWO. The effectiveness of the improved GWO algorithm has been tested using CEC2017 benchmarks. The results show that the improved GWO gives better results than the original GWO and its superiority has been proven with the Wilcoxon sign test. These results are analyzed and visually displayed. In addition, the NIOPID speed control design problem for a DC motor has been solved. Extensive simulation results have shown that the improved GWO-based NIOPID controller outperforms existing methods when comparing PID and NIOPID controllers designed using various optimization techniques in the literatüre.

Author contribution

K. DOĞAN: Research, Data collection, Data processing, Conceptualization, Literature review, Visualization, Editing; writing

H. BAŞAK: Formal analysis, Review and Editing

Funding / Acknowledgements

The authors would like to thank Artvin Çoruh University Scientific Research Projects

Coordination Office for supporting this study (BAP2024.F15.02.01).

Conflict of Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical standards

No Ethics Committee Approval is required for this study.

References

Agarwal, J., Parmar, G., Gupta, R., Sikander, A. (2018). Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor. *Microsystem Technologies*, 24, 4997–5006. https://doi.org/10.1007/s00542-018-3920-4

Alavi, M., and Henderson, J.C. (1981). An evolutionary strategy for implementing a decision support system. *Management science*, 27(11), 1309-1323. https://doi.org/10.1287/mnsc.27.11.1309

Ali, E.S. (2015). Speed control of DC series motor supplied by photovoltaic system via firefly algorithm. *Neural Computing and Applications*, 26, 1321–1332 https://doi.org/10.1007/s00521-014 1796-5

Al-Qunaieer, F.S., Tizhoosh, H.R., and Rahnamayan, S. (2010). `Opposition based computing_A survey,' *The 2010 International Joint Conference on Neural Networks (IJCNN)*, Barcelona, Spain, 2010, 1-7. https://doi.org/10.1109/IJCNN.2010.5596906

Arini, F.Y., Chiewchanwattana, S., Soomlek, C., Sunat, K. (2022). Joint Opposite Selection (JOS): A premiere joint of selective leading opposition and dynamic opposite enhanced Harris' hawks optimization for solving single-objective problems. *Expert Systems with Applications*, 188, 116001. https://doi.org/10.1016/j.eswa.2021.116001

Arini, F.Y., Sunat, K. and Soomlek, C. (2022). Golden jackal optimization with joint opposite selection: An enhanced nature-inspired optimization algorithm for solving optimization problems. *IEEE Access*, 10, 128800–128823. https://doi.org/10.1109/ACCESS.2022.3227510

Awad, N.H., Ali, M.Z., Suganthan, P. N., Liang, J.J. and Qu, B.Y. (2016). Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real parameter numerical optimization, Nanyang Technological University., Singapore, Tech. Rep., 2016. [Online]. Available: https://www3.ntu.edu.sg/home/epnsugan/index_les/CEC2017/CEC2017.htm

Ayinla, S.L., Amosa, T.I., Ibrahim, O., Rahman, M.S., Bahashwan, A.A., Mostafa, M.G., Yusuf, A.O. (2024). Optimal control of DC motor using leader-based Harris Hawks optimization algorithm. *Franklin open*, *6*, 100058. https://doi.org/10.1016/j.fraope.2023.100058

Bhatt, R., Parmar, G., Gupta, R., Sikander, A. (2019). Application of stochastic fractal search in approximation and control of LTI systems. *Microsystem Technologies*, 25, 105–114. https://doi.org/10.1007/s00542-018-3939-6

Cao, B., Zhao, J., Liu, X., Arabas, J., Tanveer, M., Singh, A. K., Lv, Z. (2022). Multiobjective evolution of the explainable fuzzy rough neural network with gene expression programming. *IEEE Transactions on Fuzzy Systems*, *30*(10), 4190-4200. https://doi.org/10,1109/TFUZZ.2022.3141761

Chen, H., Qiao, H., Xu, L., Feng, Q., Cai, K. (2019). A fuzzy optimization strategy for the implementation of RBF LSSVR model in vis–NIR analysis of pomelo maturity. *IEEE Transactions on Industrial Informatics*, 15(11), 5971-5979. https://doi.org/10.1109/TII.2019.2933582

Črepinšek, M., Liu, S.H., Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys*, *45*(3), 1-33. https://doi.org/10.1145/2480741.24807

Çelik, E., Öztürk, N. (2018). First application of symbiotic organisms search algorithm to off-line optimization of PI parameters for DSP based DC motor drives. *Neural Computing and Applications*. 30, 1689–1699. https://doi.org/10.1007/s00521-017-3256-5

Dhargupta, S., Ghosh, M., Mirjalili, S., Sarkar, R. (2020). Selective opposition based grey wolf optimization. *Expert Systems with Applications*, *151*, 113389. https://doi.org/10.1016/j.eswa.2020.113389

Dorigo, M., Stützle, T. (2003). The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. **In**: *Handbook of Metaheuristics*, Glover, F., Kochenberger, G.A. (eds). International Series in Operations Research and Management Science, Springer, Boston, MA. https://doi.org/10.1007/0-306-

48056-5_9

Ekinci, S., D. Izci and B. Hekimoğlu, "PID Speed Control of DC Motor Using Harris Hawks Optimization Algorithm," *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ICECCE49384.2020.9179308.

Ekinci, S., Izci, D., Hekimoğlu, B. (2021). Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm. *Arabian Journal for Science and Engineering*, *46*(2), 1395-1409. https://doi.org/10.1007/s13369-020-05050-z

Ergezer, M., Simon, D., Du, D. (2009, October). Oppositional biogeography-based optimization. *2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, 2009, pp. 1009-1014. https://doi.org/10.1109/ICSMC.2009.5346043

Gonzalez, T.F. (2007). Handbook of Approximation Algorithms and Metaheuristics, CRC Press, Boca Raton, FL, USA. https://doi.org/10.1201/9781420010749

Griffin, I. (2003). On-line PID Controller Tuning using Genetic Algorithms, Master's Thesis, Dublin City University, August 22, 2003.

Gursoy, F., Gunnec, D. (2018) Influence maximization in social networks under Deterministic Linear Threshold Model. *Knowledge-Based Systems*, 161(2018), 111–123. https://doi.org/10.1016/j.knosys.2018.07.040.

Hekimoğlu, B. (2019). Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. *IEEE Access*, 7, 38100-38114. https://doi.org/10.1109/ACCESS.2019.2905961

Holland, J.H. (1992) Genetic Algorithms. *Scientific American*, 267, 66-72. http://dx.doi.org/10.1038/scientificamerican0792-66

Izci, D., Ekinci, S., Zeynelgil, H. L., Hedley, J. (2021). Fractional order PID design based on novel improved slime mould algorithm. *Electric Power Components and Systems*, 49(9-10), 901-918. https://doi.org/10.1080/15325008.2022.2049650

Izci, D., Ekinci, S. (2023). Fractional order controller design via gazelle optimizer for efficient speed regulation of micromotors. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, *6*, 100295. https://doi.org/10.1016/j.prime.2023.100295

J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95-*International Conference on Neural Networks*, Perth, WA, Australia, 4, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Lampinen, J., Storn, R. (2004). Differential Evolution. **In:** *New Optimization Techniques in Engineering. Studies in Fuzziness and Soft Computing*, Springer, Berlin, Heidelberg. 141, 123-166. https://doi.org/10.1007/978-3-540-39930-8_6

Li, N., Wang, L., Jiang, Q., Li, X., Wang, B., He, W. (2021). An improved genetic transmission and dynamic-opposite learning strategy for multitasking optimization. *IEEE Access*, 9, 131789-131805. https://doi.org/10.1109/ACCESS.2021.3114435

Lin, Y., Song, H., Ke, F., Yan, W., Liu, Z., Cai, F. (2022). Optimal caching scheme in D2D networks with multiple robot helpers. *Computer Communications*, 181, 132-142. https://doi.org/10.1016/j.comcom.2021.09.027

Liu, H., Wu, Z., Li, H., Wang, H., Rahnamayan, S., Deng, C. (2014). Rotation-Based Learning: A Novel Extension of Opposition-Based Learning. **In:** Pham, D.N., Park, S.B. (eds) *Trends in Artificial Intelligence. PRICAI 2014*. Lecture Notes in Computer Science (), 8862. Springer, Cham. https://doi.org/10.1007/978-3-319-13560-1_41

Liu, C., Wu, D., Li, Y., Du, Y. (2021). Large-scale pavement roughness measurements with vehicle crowdsourced data using semi-supervised learning. *Transportation Research Part C: Emerging Technologies*, 125, 103048. https://doi.org/10.1016/j.comcom.2021.09.027

Mahdavi, S., Rahnamayan, S., Deb, K. (2018). Opposition based learning: A literature review. *Swarm and evolutionary computation*, 39, 1-23. https://doi.org/10.1016/j.swevo.2017.09.010

Meng, F., Pang, A., Dong, X., Han, C. Sha, X. (2018). H∞ optimal performance design of an unstable plant under bode integral constraint. *Complexity*, 2018 (2018) 1–10. https://doi.org/10.1155/2018/4942906

Meng, F., Wang, D., Yang, P., Xie, G. (2019). Application of sum of squares method in nonlinear H∞ control for satellite attitude maneuvers. *Complexity*, 2019(1), 5124108. https://doi.org/10.1155/2019/5124108

Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F. and Rodríguez, A. (2020). A better balance in metaheuristic algorithms: Does it exist? *Swarm and Evolutionary Computation*, 54, 100671. https://doi.org/10.1016/j.swevo.2020.100671

Nasrabadi, M. S., Sharafi, Y., Tayari, M. (2016). A parallel grey wolf optimizer combined with opposition-based learning. *2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, Bam, Iran,18-23p. https://doi.org/10.1109/CSIEC.2016.7482116

Petras, I. (1999). The fractional order controllers: Methods for their synthesis and application, *Journal of Electrical Engineering*, 50, (9-10), 284-288. https://arxiv.org/pdf/math.OC/0004064

Podlubny, I. (1999). Fractional-order systems and PIλDδ controllers. *IEEE Transaction on Automatic Control*, 44(1) 208-213. https://doi.org/10.1109/9.739144

Potnuru, D., Mary, K.A., Babu, C.S. (2019). Experimental implementation of Flower Pollination Algorithm for speed controller of a BLDC motor. *Ain Shams Engineering Journal*, *10*(2), 287-295. https://doi.org/10.1016/j.asej.2018.07.005

Pradhan, M., Roy, P.K., Pal, T. (2018). Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system. *Ain Shams Engineering Journal*, 9(4), 2015-2025. https://doi.org/10.1016/j.asej.2016.08.023

Qu, S., Han, Y., Wu, Z., Raza, H. (2021). Consensus modeling with asymmetric cost based on data-driven robust optimization. *Group Decision and Negotiation*, 30(6), 1395-1432. https://doi.org/10.1007/s10726-020-09707-w

Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. (2008). Opposition versus randomness in soft computing techniques. *Applied Soft Computing*, 8(2), 906-918. https://doi.org/10.1016/j.asoc.2007.07.010

Rahnamayan, S., Jesuthasan, J., Bourennani, F., Salehinejad, H. Naterer, G.F. (2014). Computing opposition by involving entire population. *IEEE Congress on Evolutionary Computation (CEC)*, 22 September 2014, Beijing, China, 1800–1807. https://doi.org/10.1109/CEC.2014.6900329

Rodríguez-Molina, A., Villarreal-Cervantes, M.G., Aldape-Pérez, M. (2017). An adaptive control study for a DC motor using meta-heuristic algorithms. *IFAC-PapersOnLine*, 50 (1), 13114–13120. https://doi.org/10.1016/j.ifacol.2017.08.2164

Sarma, H., Bardalai, A. (2024). Improvisation of artificial hummingbird algorithm through incorporation of chaos theory in intelligent optimization of fractional order PID controller tuning. *International Journal of Information Technology*, 2024. https://doi.org/10.1007/s41870-024-01791-4

Sebald, A.V. and Fogel, L. J. (1994). Evolutionary Programming. *Proceedings of the Third Annual Conference*. 24-26 February 1994, San Diego, CA, USA. 384p. https://doi.org/10.1142/2401

Tepljakov, A. (2017). FOMCON: fractional-order modeling and control toolbox. **In:** *Fractional-order Modeling and Control of Dynamic Systems*. Springer Theses, Springer International Publishing AG, Cham, ISBN: 978-3-319-52949-3. https://doi.org/10.1007/978-3-319-52950-9_6

Tizhoosh, H.R. (2005, November). Opposition-based learning: A new scheme for machine intelligence. *In International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, 28-30 November 2005, Vienna, Austria. https://doi.org/10.1109/CIMCA.2005.1631345

GitHub, (2024) https://github.com/FlorentinaJOS/JOS_in_Optimization/releases/tag/v1.1.1, Accession date: 5 September 2024.

Wang, N., Xu, Q., Fei, R., Wang, L., Shi, C. (2019). Are two opposite points better than one. *IEEE Access*, 7, 146108-146122. https://doi.org/10.1109/ACCESS.2019.2946089

Wang, Y., Wang, H., Zhou, B., Fu, H. (2021). Multi-dimensional prediction method based on Bi-LSTMC for ship roll. *Ocean Engineering*, 242, 110106. https://doi.org/10.1016/j.oceaneng.2021.110106

Wolpert, D.H., Macready, W.G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82. https://doi.org/10.1109/4235.585893

Xu, X., Wang, C., Zhou, P. (2021). GVRP considered oil-gas recovery in refined oil distribution: from an environmental perspective. *International Journal of Production Economics*, 235, 108078. https://doi.org/10.1016/j.ijpe.2021.108078

Xu, Q., Wang, L., Wang, N., Hei, X., Zhao, L. (2014). A review of opposition-based learning from 2005 to 2012. *Engineering Applications of Artificial Intelligence*, 29, 1-12. https://doi.org/10.1016/j.engappai.2013.12.004

Xu, Y., Yang, Z., Li, X., Kang, H., Yang, X. (2020). Dynamic opposite learning enhanced teaching_learning-based optimization. *Knowl.-Based Systems.*, 188, 104966. https://doi.org/10.1016/j.knosys.2019.104966.

Xue, D., Zhao, C., Chen, Y.Q. (2006). Fractional Order PID Control of a DCMotor with Elastic Shaft: A Case Study, *2006 American Control Conference,* 14-16 June 200, Minneapolis, Minnesota, USA, 3182–3187p. https://doi.org/10.1109/ACC.2006.1657207

Yang, X.-S., Deb, S., Fong, S. (2014). Metaheuristic algorithms: Optimal balance of intensi_cation and diversi_cation. *Applied Mathematics & Information Sciences*. 8(3), 977. https://doi.org/10.12785/amis/080306.

Zhang, S., Luo, Q., Zhou, Y. (2017). Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method. *International Journal of Computational Intelligence and Applications*, 16(02), 1750012. https://doi.org/10.1142/S1469026817500122

Zhang, Z., Wang, L., Zheng, W., Yin, L., Hu, R., Yang, B. (2022). Endoscope image mosaic based on pyramid ORB. *Biomedical Signal Processing and Control,* 71, 103261. https://doi.org/10.1016/j.bspc.2021.103261.