Volitive Grey Wolf Optimizer

João Paulo F. G. da Silva
Department of Computer Engineering
University of Pernambuco
Recife, Brazil
jpfgs@ecomp.poli.br

Rodrigo Cesar Lira
Department of Computer Engineering
University of Pernambuco
Recife, Brazil
rcls@ecomp.poli.br

Mariana Macedo
Center for Collective Learning, ANITI
University of Toulouse
Occitania, France
mmacedo@biocomplexlab.org

Hugo Valadares Siqueira
Department of Electrical Engineering
Federal University of Technology - Parana
Ponta Grossa, Brazil
hugosiqueira@utfpr.edu.br

Carmelo Bastos-Filho
Department of Computer Engineering
University of Pernambuco
Recife, Brazil
carmelofilho@ieee.org

***Abstract*—Swarm-based metaheuristics have become the most prominent method for solving optimization problems. Several operators already proposed in the literature can also be reused to expand the current metaheuristics. We present in this paper the Volitive Grey Wolf Optimizer (VGWO), a Grey Wolf Optimizer variant created by the addition of the collective volitive movement proposed in Fish School Search. The Volitive operator allows a self-regulated balance between exploration and exploitation that generates diversity when necessary. We evaluate the performance of VGWO and five other metaheuristics by simulating them in ten different problems. VGWO has overcome in most cases compared to other well-known metaheuristics. Therefore, we found that by including a self-regulating operator as the volitive collective, we can improve the quality of results provided by GWO.**

***Index Terms*—Optimization, Metaheuristics, Swarm Intelligence, Search algorithms, Grey Wolf Optimization, Fish School Search**

I. Introduction

Swarm intelligence (SI) has become a relevant tool for solving optimization problems, especially in high-dimensional search spaces. SI algorithms are population-based approaches that use an iterative procedure to find a suitable solution from an initial state or set of possible solutions [1]. The simple reactive agents compose a population, and each agent represents a candidate solution within the search space [2], [3]. Simple reactive agents perceive, act, and exchange information in a small part of the environment in which they act using a communication protocol [4].In this way, no centralized control is necessary for its operation. The algorithm's behavior emerges as the synergistic result of the cooperation among the agents and their interactions with the environment [5].

After three decades since the proposal of the first swarm intelligence, hundreds of metaheuristics have been proposed in the literature [6]. Many bioinspired algorithms have shown to be suitable for solving different types of problems [7]. However, one of the main problems in this area is that many of these inspirations result in similar computational mechanisms, despite being inspired by swarms of different animals. Nevertheless, some canonical algorithms have proven to be very efficient in solving highly complex problems, such as Particle Swarm Optimization (PSO) [8], Artificial Bee Colony (ABC) [9], and Ant Colony Optimization (ACO) [10]. These examples are well-known algorithms inspired by flocking and foraging behavior.

Another interesting example that differs from the most well-known metaheuristics is the Fish School Search (FSS). The FSS was proposed in 2008 by Bastos-Filho and Lima-Neto [11] and was inspired by the behavior of fish schools. In this case, the swarm moves using three operators: (i) individual, (ii) collective instinctive, and (iii) collective volitive movements. Each agent has an attribute called 'weight' representing the success obtained during the search that is used to guide the balance between exploration and exploitation. The FSS is the first algorithm of the field with the ability to self-regulate the search granularity by an operator called Volitive movement.

On the other hand, the Grey Wolf Optimizer (GWO) [12] presents efficient exploitation features. The GWO is a swarm-based algorithm guided by the three fittest agents (alpha, beta, and delta). Because of them, GWO has presented a stable behavior in steep landscapes, and its variants have been applied to various problems [13]–[15]. However, many authors criticize GWO due to its limitations [16] and supposed lack of novelty [17]. Thus, like the PSO, the most significant limitation is the generation of diversity after exploitation in one of the regions in the search space.

This paper introduces adding the volitive collective movement proposed in Fish School Search to improve the Grey Wolf Optimizer capabilities. We aim to demonstrate that including a proper operator can improve the quality of the results provided by the GWO. We addressed GWO since it is simple to implement and presents promising results in many optimization problems. Nevertheless, it does not have a mechanism to avoid being trapped in local minima. However, any metaheuristic could be used following the same methodology.

This paper is divided as follows: Section II briefly describes Grey Wolf Optimizer and Fish School Search. Our proposal, Volitive Grey Wolf Optimizer (VGWO), is shown in Section III. Section IV describes the methodology and parameterization for the experiments. Section V presents our findings and results, and we finish in Section VI with our conclusions.

II. Background

A. Grey Wolf Optimizer

Grey Wolf Optimizer (GWO) is a metaheuristic inspired by the hunting behavior of grey wolves. They have a hierarchy that divides the responsibilities among them. The most dominant individual (leader) is the alpha that leads the pack. Beta is the second layer. It reinforces the alpha's commands to the others and is an alpha advisor. Delta wolves are the scouts, sentinels, elders, hunters, and caretakers. Delta wolves submit to alphas and betas, but they dominate the lowest level in the pack, the omegas.

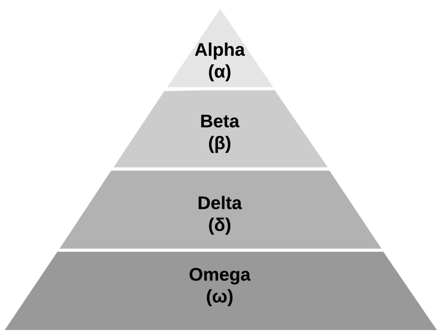

Fig. 1. The social hierarchy of the grey wolves.

The wolves hunt their prey in a group, following a set of specific steps: (i) tracking, chasing, and approaching the prey; (ii) pursuing, encircling, and harassing the prey until it stops moving; and (iii) attacking the prey.

These behaviors inspired the metaheuristic proposed by Mirjalili et al. [12]. Tracking is done from the reference of the three wolves at the top of the hierarchy (position indicated by $\mathbf{X}$). Their movement is based on a vector search, in which the next position of the wolf will be defined through vectors $\mathbf{A}$ and $\mathbf{C}$. All the other wolves follow their average vector position, as shown in Figure 2. One can observe that through vector $\mathbf{C}$, exploration (or pursuit) of the area is carried out, and with vector, direct hunting by prey (exploitation) is carried out.

The final position of the omega wolf would be in a random place inside a circular region that is defined by the positions of alpha, beta, and delta in the search space. In other words, alpha, beta, and delta indicate the position of the prey, and other wolves update their positions randomly around the prey. The vector has a decreasing behavior that imitates the wolf's behavior when approaching the prey to attack the prey. $\mathbf{C}$, on the other hand, refers to the obstacles between the wolf and the prey so that the wolf can explore the search space slowly. The GWO movement uses Equations 1 and 2.

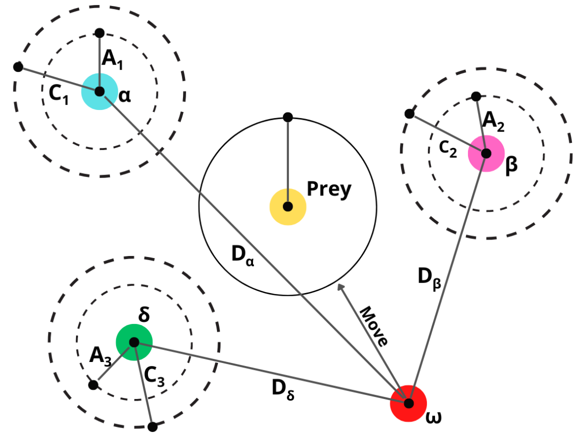

Fig. 2. Graphical demonstration of GWO movement, where ω is the agent that is being attracted to the region where α, β, and δ are located in the search space.

$$\mathbf{D} = \left|\mathbf{C} \cdot \mathbf{x}_p(t) - \mathbf{x}(t)\right|. \tag{1}$$

$$\mathbf{x}(t+1) = \mathbf{x}_p(t) - \mathbf{A} \cdot \mathbf{D}, \tag{2}$$

where $\mathbf{x}(t)$ and $\mathbf{x}_p(t)$ are the wolf position and the position of the prey in the iteration t, respectively. $\mathbf{A}$ and $\mathbf{C}$ are the coefficients calculated by:

$$\mathbf{C} = 2 \cdot \mathbf{r}_2, \tag{3}$$

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a}, \tag{4}$$

GWO pseudocode is described in Algorithm 1.

Algorithm 1: GWO Pseudocode

```
Initialize the a, A, and C
Initialize the N wolves randomly
Find the α, β, δ solutions based on fitness
while stop criterion is not reached do
    Update the wolves' position
    Update a, A and C
    Evaluate the current position of individual wolves
    Update α, β, δ
end while
Return the best solution
```

B. Fish School Search

Bastos-Filho et al. [11] proposed Fish School Search (FSS), inspired by the collective movement of a fish school to find food. Each agent (fish) i has two attributes: a position vector $\mathbf{x}_i$ and its weight, W_i. The current position $\mathbf{x}_i$ of a fish i is a D-dimensional vector within the search space, representing a candidate solution to an optimization problem. The school aims to reach the region in the search space with the highest food quality (optimal solution).

The original version of the FSS algorithm presented four operators, which can be grouped into feeding and swimming. The feeding process updates the fish weights, whereas the swimming operators drive the fish movements. FSS pseudocode is described in Algorithm 2.

Algorithm 2: FSS Pseudocode

```
Initialize the N fish randomly
while stop criterion is not reached do
    Apply the individual movement
    Apply the feeding operator
    Apply collective instinctive movement
    Calculate the fish school barycenter
    Apply collective volitive movement
    Update individual and volitive steps
end while
Return the best solution
```

1) Individual Movement: It is the first movement performed by FSS. Each fish finds a random new position in the search space in the individual movement using Equation 5.

$$\mathbf{n}_i(t+1) = \mathbf{x}_i(t) + \mathbf{s}(t+1), \tag{5}$$

where $\mathbf{s}(t+1)$ is a random vector calculated for the entire school at each iteration by $s_{\text{ind}}(t+1) \cdot \text{rand}[-1;1]$. The parameter $s_{\text{ind}}(t+1)$ is a hyperparameter called "individual step". $\text{rand}[-1;1]$ is a random value uniformly generated in $[-1;1]$. Each fish performs a greedy search and moves to $\mathbf{n}_i(t+1)$ if the neighbor's position (new position) is better than the current position.

2) Feeding operator: The feeding operator is exclusive for the fish that successfully moved in the individual movement. The neighbor position $\mathbf{n}_i(t+1)$ is evaluated in terms of the fitness function $f[\mathbf{n}_i(t+1)]$. For each fish, the difference between the fitness of the neighbor position $f[\mathbf{n}_i(t+1)]$ and the current position $f[\mathbf{x}_i(t)]$ is evaluated (shown in Equation 6). This difference $\Delta f_i(t+1)$ is used to calculate the weight $W_i(t+1)$ according to Equation 8.

$$\Delta f_i(t+1) = f[\mathbf{n}_i(t+1)] - f[\mathbf{x}_i(t)]. \tag{6}$$

$$G_i(t+1) = \begin{cases} \frac{\Delta f_i(t+1)}{\max[\Delta f(t+1)]}, & \max[\Delta f(t+1)] \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$W_i(t+1) = W_i(t) + G_i(t+1). \tag{8}$$

3) Collective Instinctive Movement: The collective instinctive attracts the school to the regions where the fish were moved successfully at iteration $t+1$ in the individual movement. If a fish i moved in the individual movement, it means that fish i now is in a better position than in the iteration t, so the fish i should attract the school to that region. Therefore, the instinctive movement is computed by the drift, $\mathbf{m}(t+1)$ in Equation 9, which considers all the successful fish displacements and fitness gain.

$$\mathbf{m}(t+1) = \frac{\sum_{i=1}^{N} \Delta\mathbf{x}_i(t+1) \cdot \Delta f_i(t+1)}{\sum_{i=1}^{N} \Delta f_i(t+1)}, \tag{9}$$

where $\Delta\mathbf{x}_i(t)$ is the displacement of the fish i generated by the individual movement and N is the number of fish in the school. This drift is then applied to update the positions of all fish according to Equation 10:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{m}(t+1). \tag{10}$$

4) Collective Volitive Movement: This movement is based on the overall success of the entire school, i.e., the total school weight, $W(t+1)$, which is the sum of weights of each fish at iteration $t+1$. If $W(t+1) > W(t)$, the school weight has increased, so the search was successful, and the radius of the school should contract to increase the exploitation ability of the school. On the other hand, if $W(t+1) \leq W(t)$, the school expands to allow more exploration of the search space. This movement is performed based on the swarm barycenter, calculated as Equation 11.

$$\mathbf{B}(t+1) = \frac{\sum_{i=1}^{N} \mathbf{x}_i(t+1) \cdot W_i(t+1)}{\sum_{i=1}^{N} W_i(t+1)}. \tag{11}$$

Based on the overall weights of the school at iteration $t+1$, the fish positions are updated as Equation 12.

$$\mathbf{x}_i(t+1) = \begin{cases} \mathbf{x}_i(t) - s_{\text{vol}}(t+1) \cdot \mathbf{m}_v, & \text{if } W(t+1) > W(t) \\ \mathbf{x}_i(t) + s_{\text{vol}}(t+1) \cdot \mathbf{m}_v, & \text{otherwise,} \end{cases} \tag{12}$$

where $s_{\text{vol}}(t+1)$ is the volitive step, $\text{rand}[0;1]$ is a vector of continuous random values from the interval of 0 and 1, and $\mathbf{m}_v$ is calculated as Equation 13.

$$\mathbf{m}_v = \text{rand}[0;1] \cdot \mathbf{x}_i(t) - \mathbf{B}(t+1) \tag{13}$$

where $s_{\text{vol}}(t+1)$ parameter controls the size of the displacement of each fish. The negative and positive signs in Equation 12 parameter attract or repel the agents toward the school's barycenter.

III. Volitive Grey Wolf Optimizer

This paper proposes a GWO variant called Volitive Grey Wolf Optimizer (VGWO) that incorporates the collective volitive movement from Fish School Search. Using this approach, we improve the GWO exploration ability, enabling a movement based on the dilation or contraction regarding the swarm's barycenter. Therefore, we improve the capability of VGWO to escape from local minima.

In our proposal, we changed the feeding operator to punish the agents who do not change their fitness over iterations. In this case, agents lose a small weight (ϵ) even when they do not move to another position. We changed the operator due to the GWO characteristics that lead the entire swarms to a region where it keeps exploiting the same position, as in Equation 14.

$$G_i(t+1) = \begin{cases} \frac{\Delta f_i(t+1)}{\max[\Delta f(t+1)]}, & \max[\Delta f(t+1)] \neq 0 \\ \epsilon, & \text{otherwise} \end{cases} \quad (14)$$

where ϵ is a small weight value that forces the swarm to explore different regions when the fitness remains the same.

We also incorporated the exponential function for the decay of each component of **a**, proposed by the modified GWO (mGWO) [18], as shown in Equation 15. According to the authors, the function improves the exploration capability.

$$\mathbf{a}(t) = 2 \cdot \Big(1 - \frac{t^2}{max\_iteration^2}\Big). \quad (15)$$

VGWO maintains the same structure as GWO (Pseudocode 3), adding the collective volitive movement after the GWO movement. The feeding operator and the step update are also required for the new movement to work properly.

Algorithm 3: VGWO Pseudocode

```
Initialize the a, A, and C
Initialize the N wolves randomly
Find the α, β, δ solutions based on fitness
while stop criterion is not reached do
    Update the wolves' position
    Apply the feeding operator
    Apply collective volitive movement
    Update a, A and C
    Update volitive steps
    Evaluate the current position of individual wolves
    Update α, β, δ
end while
Return the best solution
```

IV. Methodology

We compared our proposal VGWO with five metaheuristics from the literature. Beyond FSS and GWO, we included modified GWO (mGWO), the PSO with global topology (GPSO), and PSO with ring topology (LPSO) in our experiments.

We addressed ten benchmark functions with 30 dimensions, being Rotated Hyper Ellipsoid (RHE), Rosenbrock, Dixon-Price, Quartic-Noise, Schwefel, Ackley, Levy, Rastrigin, and Griewank, a set of unimodal (Table I) and multimodal functions (Table II). The five multimodal functions are differentiable. Schwefel and Rastrigin are separable functions, while Ackley, Levy, and Griewank are non-separable.

We developed the code using the Python programming language, and the experiments were conducted on a PC with an i7-10510U 1.80GHz, 16 GB memory RAM, and Ubuntu 22.04 (LTS) 64-bit. We used 30 simple reactive agents and the maximum function evaluation as the stop criterion (*MFE*=3000). In GWO, we linearly decreased the components of **a** from 2 to 0 as described in [12]. In mGWO and VGWO, we used

TABLE I
Unimodal functions

Functions	Equation
Sphere	$\sum_{i=1}^{D} x_i^2$
RHE	$\sum_{i=1}^{D} \sum_{j=1}^{i} x_j^2$
Rosenbrock	$\sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Dixon-Price	$(x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_{i-1})^2$
Quartic-Noise	$\sum_{i=1}^{d} i \cdot x_i^4 + rand[0;1]$

TABLE II
Multimodal functions

Functions	Equation
Schwefel	$418.9829 \cdot D - \sum_{i=1}^{D} x_i sin(\sqrt{\lvert x_i \rvert})$
Ackley	$-20exp\Big(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\Big) - exp\Big(\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)\Big) + 20 + exp(1)$
Levy	$sin^2(\pi\omega_1) + \sum_{i=1}^{D-1}(\omega_i - 1)^2 \cdot [1 + 10sin^2(\pi w_i + 1)] + (\omega_d - 1)^2[1 + sin^2(2\pi\omega_d)]$, where $\omega_i = 1 + \frac{x_i - 1}{4}, \forall i = 1, ..., D$
Rastrigin	$\sum_{i=1}^{D} (x_i{}^2 - 10cos(2\pi x_i) + 10 \cdot D$
Griewank	$\frac{1}{4000}\sum_{i=1}^{D} x^2 - \prod_{i=0}^{D} cos(\frac{x_i}{\sqrt{i}}) + 1$

the same limits but with an exponential decay (Equation 15) to calculate it over iteration. In FSS and VGWO, the volitive step linearly decreased from 0.1 to 0.01 over iterations, and we set 1 as the minimum weight. In FSS, the individual step was decreased from 0.1 to 0.0001 over iteration. As we have not tuned the metaheuristics' hyperparameters, we used the same values as described in FSS and GWO proposal.

For the PSO implementation, the adaptive w, c_1 and c_2 hyperparameters [19] are calculated through Equation 16, 17 and 18.

$$w(t) = \frac{0.4(t - max\_iteration)^2}{max\_iteration^2} + 0.4 \quad (16)$$

$$c_1(t) = \frac{-2 \cdot t}{max\_iteration} + 2.5 \quad (17)$$

$$c_2(t) = \frac{2 \cdot t}{max\_iteration} + 0.5 \quad (18)$$

where $max\_iteration$ in PSO is calculated as 1000 ($\frac{max\_function\_evaluation}{num\_agents}$).

V. Results and Discussion

This section shows the results of 30 simulations for each algorithm and function. We start by plotting the boxplot of the fitness values. Subsequently, we show the fitness convergence for each benchmark function. Finally, we evaluated the results using a statistical test.

Figure 3 shows the results found for each function. The boxplots indicate that our proposal found competitive results, overcoming the other metaheuristics in different benchmark functions. Compared with GWO and mGWO, VGWO found better results in Rastrigin, Schwefel, Sphere, RHE, and

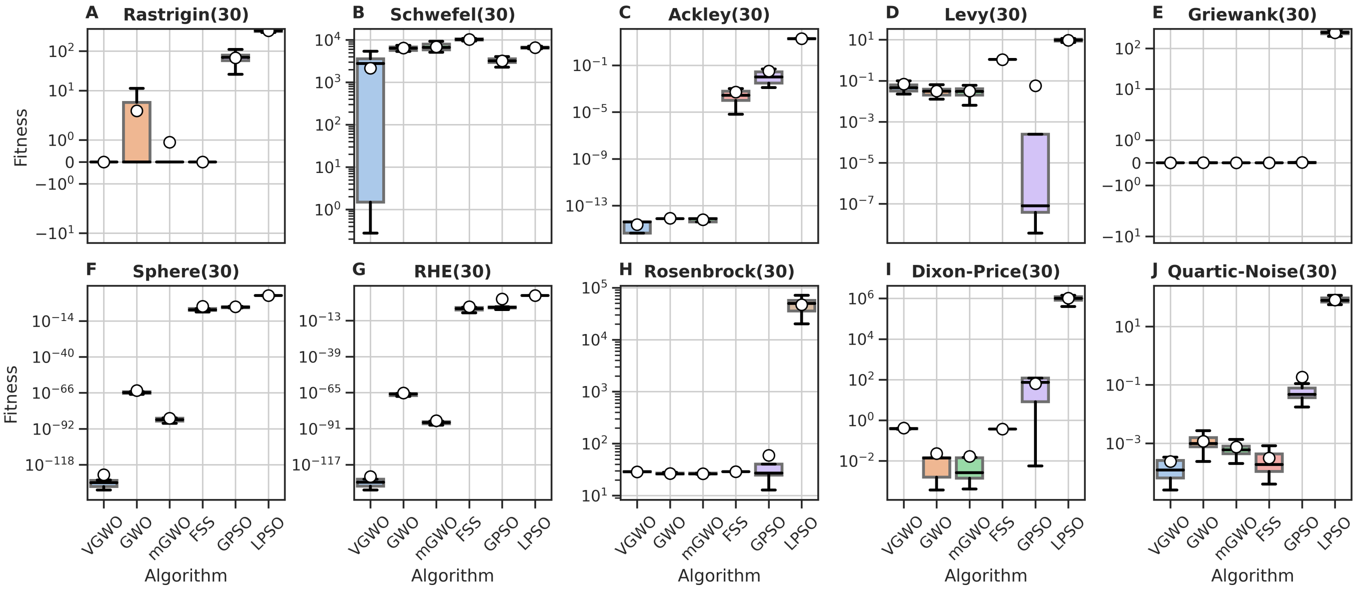

Fig. 3. Boxplot of the best fitness found on 30 simulations of each algorithm on all benchmark functions.

Quartic-Noise. GPSO was better in Levy, while GWO and mGWO were better in Dixon-Price. LPSO found the worst results in the majority of scenarios. We argue that LPSO would need more iterations to converge.

Figure 4 shows the fitness improvement as a function of the number of fitness evaluations. VGWO (blue dots) reaches lower fitness in different functions. However, in the Dixon-Price function, a valley-shaped function, we observe that VGWO presented the worst convergence.

We applied the Wilcoxon test to compare the efficiency across metaheuristics using a confidence level of 99%. The results are shown in Table III, in which '–' indicates no statistical difference between the solutions, '▲' indicates the VGWO achieved better fitness results than the other metaheuristic compared, and '▽' represents that our proposal reached worse results than the algorithm compared. The statistical tests show that adding the volitive movement was relevant for improving the GWO capabilities.

VI. Conclusions

One of the FSS contributions was the proposal of an operator called Collective Volitive movement. Based on the overall success of the entire swarm, the movement contracts or expands the agents towards its barycenter, creating a self-regulated approach to guide the balance between exploration and exploitation.

Our paper improved Grey Wolf Optimizer exploration capability by adding the Volitive movement. Therefore, we propose the Volitive Grey Wolf Optimizer (VGWO). In addition, we also used an exponential decay function to increase the exploration over exploitation.

We compared the proposal with five other metaheuristics: GWO, mGWO, FSS, GPSO and LPSO, using ten different benchmark functions. Based on a statistical test, we can state

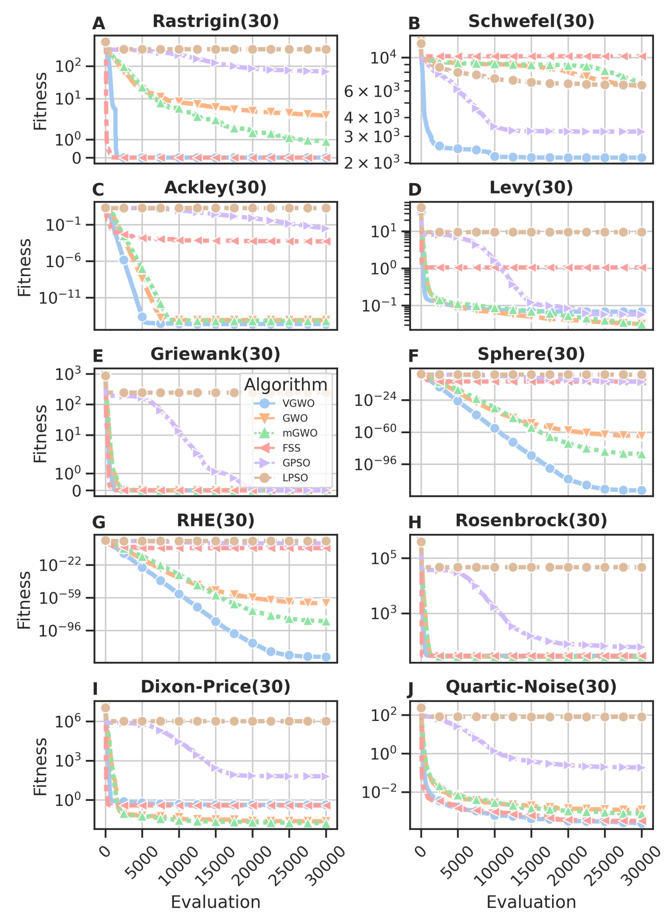

Fig. 4. Best fitness found per function evaluation on all benchmark functions.

TABLE III
RESULTS OF FITNESS VALUES AND WILCOXON TEST WITH A CONFIDENCE LEVEL OF 99% COMPARING THE VGWO WITH THE OTHER METAHEURISTICS WITH 30 DIMENSIONS.

Function		**VGWO**	**GWO**	**mGWO**	**FSS**	**GPSO**	**LPSO**
Rastrigin	Fitness	0.00E+00	3.07E+00	8.94E-01	1.67E-05	6.79E+01	3.28E+02
	STD	0.00E+00	4.10E+00	2.39E+00	4.32E-05	2.11E+01	1.55E+01
	Wilcoxon		▲	–	▲	▲	▲
Schwefel	Fitness	2.16E+03	6.47E+03	6.92E+03	1.02E+04	3.22E+03	6.56E+03
	STD	1.79E+03	1.02E+03	1.35E+03	4.55E+02	5.89E+02	2.52E+02
	Wilcoxon		▲	▲	▲	–	▲
Ackley	Fitness	2.40E-15	8.26E-15	6.31E-15	5.03E-04	3.18E-02	1.87E+01
	STD	1.77E-15	2.13E-15	1.69E-15	5.69E-04	6.33E-02	5.18E-01
	Wilcoxon		▲	▲	▲	▲	▲
Levy	Fitness	6.82E-02	3.19E-02	3.18E-02	1.06E+00	5.77E-02	9.48E+00
	STD	6.49E-02	1.40E-02	1.40E-02	1.43E-01	1.35E-01	1.24E+00
	Wilcoxon		–	▽	▲	–	▲
Griewank	Fitness	0.00E+00	3.50E-03	0.00E+00	3.17E-04	1.28E-02	2.44E+02
	STD	0.00E+00	6.18E-03	0.00E+00	6.37E-04	1.52E-02	2.06E+01
	Wilcoxon		–	–	▲	▲	▲
Sphere	Fitness	5.39E-126	5.51E-65	3.93E-85	4.40E-04	1.97E-04	2.60E+04
	STD	2.19E-125	2.24E-64	7.50E-85	1.11E-03	2.58E-04	2.29E+03
	Wilcoxon		▲	▲	▲	▲	▲
RHE	Fitness	2.25E-126	6.70E-66	5.12E-86	1.32E-03	4.29E+02	1.47E+05
	STD	6.19E-126	1.44E-65	1.49E-85	5.07E-03	1.29E+03	1.84E+04
	Wilcoxon		▲	▲	▲	▲	▲
Rosenbrock	Fitness	2.87E+01	2.66E+01	2.64E+01	2.88E+01	5.96E+01	4.69E+04
	STD	1.01E-01	5.66E-01	5.01E-01	2.29E-07	1.06E+02	1.45E+04
	Wilcoxon		▽	▽	▲	–	▲
Dixon-Price	Fitness	4.14E-01	2.33E-02	1.67E-02	3.75E-01	6.36E+01	1.04E+06
	STD	1.25E-01	3.68E-02	3.16E-02	1.39E-06	5.02E+01	3.65E+05
	Wilcoxon		▽	▽	–	▲	▲
Quartic-Noise	Fitness	2.39E-04	1.17E-03	7.44E-04	3.13E-04	1.87E-01	8.15E+01
	STD	2.67E-04	6.48E-04	4.89E-04	2.86E-04	5.80E-01	1.72E+01
	Wilcoxon		▲	▲	–	▲	▲

that adding Volitive movement improved the metaheuristic. VGWO outperforms the other metaheuristics when compared using the majority of the functions.

Despite our limited simulation scenario, we demonstrate that it is possible to be more efficient in solving a set of problems by only using the proper operators. Creating different metaheuristics based on similar mechanisms makes choosing the right metaheuristics for a problem more difficult. In the future, we aim to demonstrate the addition of Collective Volitive in different metaheuristics, and we also plan to investigate in-depth the impact of hyperparameters in different scenarios.

ACKNOWLEDGMENT

The authors thank the Foundation for the Support of Science and Technology of the State of Pernambuco (FACEPE), University of Pernambuco (UPE), Federal Institute of Pernambuco (IFPE), Brazilian National Council for Scientific and Technological Development (CNPq), processes number 40558/2018-5, 315298/2020-0, and Araucaria Foundation, process number 51497, for their financial support. Mariana Macedo was supported by the Artificial and Natural Intelligence Toulouse Institute (ANITI) - Institut 3iA: ANR-19-PI3A-0004.

REFERENCES

[1] X.-S. Yang, "Swarm intelligence based algorithms: a critical analysis," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 17–28, Apr. 2014. [Online]. Available: https://doi.org/10.1007/s12065-013-0102-2

[2] W. Song, W. Ma, and Y. Qiao, "Particle swarm optimization algorithm with environmental factors for clustering analysis," *Soft Computing*, vol. 21, no. 2, pp. 283–293, Jan. 2017. [Online]. Available: http://link.springer.com/10.1007/s00500-014-1458-7

[3] A. A. A. Esmin, R. A. Coelho, and S. Matwin, "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data," *Artificial Intelligence Review*, vol. 44, no. 1, pp. 23–45, Jun. 2015. [Online]. Available: http://link.springer.com/10.1007/s10462-013-9400-4

[4] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, Jun. 2014. [Online]. Available: https://doi.org/10.1007/s10462-012-9328-0

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial isystems*. New York: Oxford University Press, 1999.

[6] F. Campelo and C. de Castro Aranha, "Sharks, zombies and volleyball: Lessons from the evolutionary computation bestiary," in *Lifelike Computing Systems Workshop*, 2021, pp. 1–6.

[7] H. Siqueira, C. Santana, M. Macedo, E. Figueiredo, A. Gokhale, and C. Bastos-Filho, "Simplified binary cat swarm optimization," *Integrated Computer-Aided Engineering*, vol. 28, no. 1, pp. 35–50, 2021.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[9] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (abc) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.

[10] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[11] C. J. A. Bastos Filho, F. B. de Lima Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima, "A novel search algorithm based on fish school behavior," in *2008 IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 2646–2651.

[12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965997813001853

[13] N. M. Hatta, A. M. Zain, R. Sallehuddin, Z. Shayfull, and Y. Yusoff, "Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014–2017)," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2651–2683, Dec. 2019. [Online]. Available: https://doi.org/10.1007/s10462-018-9634-2

[14] V. Yadav, G. Parmar, and R. Bhatt, "Application of gwo with different performance indices for bh system," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 2019, pp. 742–745.

[15] K. Parkh, G. Paliwal, and M. Singh, "Automatic generation control of multi-area multi-source realistic system using gwo algorithm," in *2019 3rd International Conference on Recent Developments in Control, Automation Power Engineering (RDCAPE)*, 2019, pp. 425–430.

[16] P. Niu, S. Niu, N. liu, and L. Chang, "The defect of the grey wolf optimization algorithm and its verification method," *Knowledge-Based Systems*, vol. 171, pp. 37–43, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705119300188

[17] C. L. Camacho Villalón, T. Stützle, and M. Dorigo, "Grey wolf, firefly and bat algorithms: Three widespread algorithms that do not contain any novelty," in *Swarm Intelligence*, M. Dorigo, T. Stützle, M. J. Blesa, C. Blum, H. Hamann, M. K. Heinrich, and V. Strobel, Eds. Cham: Springer International Publishing, 2020, pp. 121–133.

[18] N. Mittal, U. Singh, and B. S. Sohi, "Modified Grey Wolf Optimizer for Global Engineering Optimization," *Applied Computational Intelligence and Soft Computing*, vol. 2016, p. e7950348, May 2016. [Online]. Available: https://www.hindawi.com/journals/acisc/2016/7950348/

[19] G. Sermpinis, K. Theofilatos, A. Karathanasopoulos, E. F. Georgopoulos, and C. Dunis, "Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization," *European Journal of Operational Research*, vol. 225, no. 3, pp. 528–540, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221712007667