

Image Segmentation Method Based on Improved Grey Wolf Optimizer

Jiexuan Sha

*Portland Institute, Nanjing University of Posts and Telecommunications, Nanjing, China
2573345970@qq.com*

Abstract. Aiming at the drawbacks of slow convergence, proneness to local optima, and insufficient segmentation accuracy of traditional multi-threshold image segmentation algorithms in complex scenarios, an Improved Grey Wolf Optimizer (IGWO) integrated with opposition-based learning strategy and nonlinear dynamic convergence factor is proposed. Specifically, we use improved opposition-based learning for population initialization and iteration optimization. This step helps increase population diversity and speed up convergence. Instead of the traditional linear decreasing convergence factor, we adopt a nonlinear dynamic one. This change achieves an adaptive balance between global exploration and local exploitation of the algorithm. We take the maximum between-class variance (OTSU) as the fitness function and build a multi-threshold segmentation optimization model. We validate IGWO through 6 benchmark test functions and compare it with four advanced swarm intelligence algorithms. The results show that IGWO has obvious advantages in convergence speed, solution accuracy and stability. It also has a strong ability to avoid local optimal solutions. When applied to multi-threshold image segmentation, IGWO produces segmentation regions with clear boundaries and well-preserved details. This algorithm provides a new technical method for efficient and accurate segmentation of complex images. It can be used in fields such as computer vision and communication equipment fault detection.

Keywords: Grey Wolf Optimizer, Opposition-based Learning Strategy, Nonlinear Convergence Factor, Multi-threshold Segmentation

1. Introduction

Against the backdrop of the continuous advancement of 5G communications and intelligent communication network construction, image segmentation technology has been increasingly applied in key communication engineering scenarios, such as communication equipment fault detection, optical fiber end-face defect identification, and communication base station inspection image analysis. As a core preprocessing step, its segmentation accuracy directly affects the reliability of subsequent communication equipment fault location, operation status evaluation, and operation and maintenance decision-making analysis. However, images in communication scenarios are often affected by on-site electromagnetic interference at base stations, impurity reflection on optical fiber end-faces, and variable illumination in outdoor inspections, resulting in problems such as uneven

gray distribution, noise superposition, and blurred target edges. Traditional single-threshold segmentation methods struggle to meet the requirements for precise detection and condition monitoring of communication equipment.

Multi-threshold segmentation has become a mainstream technical direction for image processing in communication scenarios, owing to its ability to finely characterize the gray levels of images and accurately extract target regions such as communication equipment defects and optical fiber end-face impurities. The classic maximum between-class variance method (OTSU) is widely used in communication image segmentation due to its simple principle and strong stability. Nevertheless, as the number of thresholds increases, its computational complexity grows exponentially, and the traditional exhaustive search method is extremely inefficient, making it difficult to satisfy the engineering requirements for real-time inspection of communication equipment and rapid fault response. Therefore, it is urgent to introduce efficient optimization algorithms to improve the efficiency of threshold search.

Meta-heuristic algorithms are widely used in this field. Mirjalili proposed the Grey Wolf Optimizer (GWO) in 2014. This algorithm has few parameters and strong global search ability, so it is often used to find optimal thresholds for multi-threshold segmentation [1]. However, the GWO has obvious weaknesses. First, its linear convergence factor is hard to balance global exploration and local exploitation, which makes the algorithm easy to fall into local optimal solutions. Second, random population initialization leads to uneven distribution of individuals, which affects convergence speed and accuracy. Third, the algorithm only relies on three types of decision-making layers for guidance. Insufficient information exchange between individuals causes the loss of population diversity too early. These problems limit the application effect of the basic GWO.

In recent years, many scholars have studied how to improve GWO's defects. For example, some researchers combined GWO with the differential evolution algorithm. This combination enhances population diversity, but it brings high computational complexity and premature convergence [2]. Some people used Logistic mapping to optimize initialization. This method reduces the chance of falling into local optima, but it does not improve convergence speed much [3]. Others divided the population into multiple groups to enhance global optimization ability, but this increases the difficulty of parameter adjustment [4]. In terms of application, improved GWO has higher accuracy in scenarios like chip image segmentation. But under high threshold conditions, its convergence speed or algorithm stability is still not enough [5]. Although the opposition-based learning strategy has been proven to expand the search range, few studies have combined it with nonlinear convergence factors for multi-threshold segmentation [6].

Existing research shows that optimizations such as population initialization and convergence factor adjustment can improve GWO's performance. However, it is still a research challenge to reasonably combine multiple strategies to improve convergence speed, accuracy and stability at the same time. To solve this problem, this paper proposes an Improved Grey Wolf Optimizer (IGWO). This algorithm integrates the opposition-based learning strategy and nonlinear convergence factor, and builds an efficient multi-threshold segmentation model. We use the OTSU between-class variance as the fitness function. This model can quickly find optimal thresholds, and experiments have verified the algorithm's effectiveness. The research results can provide optimal solutions for complex image segmentation and expand the engineering application scope of GWO.

2. Basic whale optimization algorithm

GWO works by simulating grey wolves' predation habits and group hierarchy to solve optimization problems. Its main working rules include dividing the group into different levels, surrounding prey,

and carrying out hunting actions. In a wolf pack, α , β and δ wolves form the decision-making group. They lead ω wolves to hunt. The optimization process of GWO has three steps: first, the wolves search for prey randomly; then, the decision-making group guides the pack to surround the prey; finally, the pack captures the prey with detailed local actions. A convergence factor controls when the algorithm switches between searching globally and focusing on local areas.

(1) Surrounding prey

When grey wolves hunt, they keep closing in on their prey. We can describe this behavior with math formulas like this:

$$F = |\mathcal{E} \cdot A_p(t) - A(t)| \quad (1)$$

$$A(t+1) = A_p(t) - H \cdot F \quad (2)$$

Here's what each symbol stands for: t is the number of iterations we've done so far; $A(t)$ is the current spot of a single grey wolf. $A_p(t)$ is the current spot of the prey. F is how far the grey wolf is from the prey. H is the convergence factor vector. \mathcal{E} is the swing factor vector, we use this to copy the random moving of grey wolves when they hunt.

(2) Convergence Factor and Swing Factor. We calculate the convergence factor H and swing factor \mathcal{E} with these formulas:

$$H = 2\eta(r_1 - 1) \quad (3)$$

$$\mathcal{E} = 2r_2 \quad (4)$$

Here's what each part means: r_1 and r_2 are random vectors. Their values range evenly between 0 and 1. η is a control parameter. Its value goes down straight from 2 to 0 as we do more iterations. The size of the convergence factor H decides how the algorithm searches: When $|H| > 1$, a single grey wolf moves away from the current prey. It looks for new possible best areas across the whole search space. When $|H| < 1$: The grey wolf moves closer to the current prey. It focuses on the local area to make the search for the best solution more precise. The swing factor \mathcal{E} makes the search more random. Its random value lets the grey wolf keep some ability to explore new areas while getting close to the prey. This stops the algorithm from settling on a local best solution too early.

(3) Hunting Behavior. ω wolves hunt when α , β and δ wolves lead them. They update their positions by working out how far they are from these three decision-making wolves. We can write this process with math like this:

$$\begin{cases} F_\alpha = |\mathcal{E}_1 \cdot A_\alpha(t) - A(t)| \\ F_\beta = |\mathcal{E}_2 \cdot A_\beta(t) - A(t)| \\ F_\delta = |\mathcal{E}_3 \cdot A_\delta(t) - A(t)| \end{cases} \quad (5)$$

The position update of ω wolves is expressed by Equation (6):

$$\begin{cases} A_1 = A_\alpha(t) - H_1 \cdot F_\alpha \\ A_2 = A_\beta(t) - H_2 \cdot F_\beta \\ A_3 = A_\delta(t) - H_3 \cdot F_\delta \end{cases} \quad (6)$$

To sum up, we use Equation (7) to show the new position of a ω wolf after it moves:

$$A(t+1) = (A_1 + A_2 + A_3)/3 \quad (7)$$

Here's what each symbol means: $A_\alpha(t)$, $A_\beta(t)$, and $A_\delta(t)$ are the positions of α , β and δ wolves in the t -th iteration; F_α , F_β , and F_δ are how far the current wolf is from each of these three decision-making wolves; H_1, H_2, H_3 and $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ are three separate sets of convergence factors and swing factors; $A(t+1)$ is the new position of the wolf after the update. When ω wolves take the positions of the three decision-making wolves, weight them, and calculate their average, they get closer to the best solution step by step. This process copies how grey wolves hunt in real life.

3. Improved Grey Wolf Optimizer

3.1. Introduction of improved opposition-based learning strategy

The main idea of the opposition-based learning strategy is this: For any workable solution, we calculate its opposite solution and check how good both solutions are. We keep the better one. This way, we make the search range wider and make the group of solutions more diverse. The formula to calculate the opposite solution in the traditional opposition-based learning strategy is:

$$A^* = (L_j + U_j) - A \quad (8)$$

Here's what each part means: A is the original position of a single solution; A^* is the traditional opposite solution; L_j and U_j are the lowest and highest limits of the search space. This method is simple to use. But the opposite solutions are spread in a fixed way. They can't cover all possible solutions well, so the method doesn't work as well as it could.

To make opposition-based learning better, this paper puts forward an improved version of the strategy. We add a random disturbance term to the traditional opposite solution. This makes the opposite solutions more diverse. The formula for this improved solution is:

$$A' = A^* + r_3 \cdot \left(\frac{L_j + U_j}{2} - A^* \right) \quad (9)$$

where A' is the improved opposite solution; r_3 is a random vector, its values are spread evenly between 0 and 1.

3.2. Design of nonlinear convergence factor

In the basic GWO, the control parameter η goes down in a straight line. This way can't meet the needs of solving complex optimization problems. At the start of iterations: η drops too fast. This limits how wide the algorithm can search, so it can't check all possible solutions fully. At the end of iterations: η becomes too small. This makes the local search steps too short, so the algorithm can't get close to the best solution accurately. To solve this problem, this paper creates a nonlinear dynamic convergence factor. We use the exponential function and iteration progress together to adjust η automatically. The formula for this factor is:

$$\eta(t) = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \cdot e^{-\varphi \cdot (t/T_{\max})^2} \quad (10)$$

Where $\eta_{\max} = 2$ and $\eta_{\min} = 0.1$ are the largest and smallest values of the convergence factor, $\varphi = 4$ is an adjustment parameter. It controls how fast the convergence factor drops. t is the number of iterations we've done so far, T_{\max} is the total number of iterations we plan to do.

3.3. Design of fitness function

When applying the improved GWO to image multi-threshold segmentation, the maximum between-class variance (OTSU) is used as the fitness function, and the goal is to find the optimal threshold combination to maximize the between-class variance. Assuming that the image gray level range is $[0255]$, m thresholds T_1, T_2, \dots, T_m ($0 < T_1 < T_2 < \dots < T_m < 255$) are selected to divide the image into $m + 1$ regions C_1, C_2, \dots, C_{m+1} , where the gray range corresponding to region C_i is $[T_{i-1} + 1, T_i]$ ($T_0 = -1, T_{m+1} = 255$).

The specific calculation steps of the fitness function are as follows:

1. Calculate the image gray histogram: Let $h(g)$ be the number of pixels corresponding to the gray value g , and the total number of image pixels $N = \sum_{g=0}^{255} h(g)$;

2. Calculate the pixel proportion of each region: $p_i = \sum_{g=T_{i-1}+1}^{T_i} h(g)/N$ ($i = 1, 2, \dots, m + 1$), satisfying $\sum_{i=1}^{m+1} p_i = 1$;

3. Calculate the average gray level of each region: $\mu_i = \sum_{g=T_{i-1}+1}^{T_i} g \cdot h(g) / \sum_{g=T_{i-1}+1}^{T_i} h(g)$;

4. Calculate the global average gray level: $\mu_{total} = \sum_{i=1}^{m+1} p_i \cdot \mu_i$;

5. Calculate the between-class variance: $\sigma^2 = \sum_{i=1}^{m+1} p_i \cdot (\mu_i - \mu_{total})^2$.

The larger the between-class variance σ^2 , the more significant the difference between the segmented regions, and the better the segmentation effect. Therefore, the optimization goal of the improved GWO is to maximize σ^2 , and the fitness function is defined as $f = \sigma^2$.

3.4. Complete flow of the improved algorithm

We add the elite opposition-based learning strategy and the nonlinear convergence factor to GWO. These changes stop GWO from getting stuck in local optimal solutions. This improved version of GWO is called IGWO. The full steps to use IGWO for multi-threshold image segmentation are as follows:

Step 1 Image preprocessing: Read the original image, convert it into a grayscale image, and calculate the grayscale histogram;

Step 2 Initialize parameters: Set the population size N , T_{\max} , number of thresholds m , etc.;

Step 3 Generate initial population: Randomly generate N m -dimensional individuals (each dimension corresponds to a threshold) according to Formula (9), generate opposite solutions through the improved opposition-based learning strategy, merge them, and select the optimal N individuals as the initial population;

Step 4 Calculate fitness values: Calculate the fitness value of each individual according to the OTSU between-class variance formula, sort them to determine the current $A_\alpha, A_\beta, A_\delta$;

Step 5 Update individual positions: Update the position of each individual according to Equations (1) to (7) and (9), and perform boundary processing;

Step 6 Opposition-based learning optimization: Generate improved opposite solutions for the updated individuals, calculate their fitness, and select the better ones to replace;

Step 7 Update optimal positions: Recalculate the fitness values of all individuals, and update A_α , A_β , A_δ ;

Step 8 Termination judgment: If the number of iterations reaches T_{max} , output the optimal threshold combination; otherwise, return to Step 5 to continue iteration;

Step 9 Image segmentation: Segment the grayscale image according to the optimal threshold combination and output the segmentation result.

4. Experimental results and analysis

To check how well the Improved Grey Wolf Optimizer (IGWO) works, we design two experiments:

1. A benchmark test function experiment. This experiment tests how accurately the algorithm can find optimal solutions and how fast it converges.

2. An image segmentation experiment. This experiment tests how well the algorithm works in real multi-threshold segmentation tasks.

4.1. Benchmark test function experiment

(1) Experimental Settings

We pick six widely used benchmark test functions. They include 4 unimodal functions (F1-F4) and 2 multimodal functions (F5-F6). These functions cover different types of optimization problems. We use them to test two abilities of the algorithm: finding optimal solutions globally and refining searches locally. Table 1 shows each function's name, search range, and best possible value.

Table 1. Information of benchmark test functions

Number	Function Name	Range	Optimal Value
F1	Schwefel's problem 2.21	[-100, 100]	0
F2	Quartic Function	[-1.28, 1.28]	0
F3	Schwefel's problem 1.2	[-100, 100]	0
F4	Sphere Function	[-100, 100]	0
F5	Generalized Penalized Function	[-50, 50]	0
F6	Generalized Penalized Function	[-50, 50]	0

In this experiment, we set the same parameters for all algorithms: population size $N = 30$, maximum iterations: $T_{max} = 500$, and problem dimension $d = 30$. We compare IGWO with four other algorithms: basic GWO, Slime Mould Algorithm (SMA) [7], Arithmetic Optimization Algorithm (AOA) [8], and Multi-Verse Optimizer (MVO) [9]. We use the parameter settings from each algorithm's original research paper. Each algorithm runs 30 times alone on every test function. We then calculate the average optimal value and standard deviation for each run. These two values are our main metrics to judge performance.

(2) Experimental Results and Analysis

Table 2 shows the average optimal values and standard deviations of the five algorithms on six benchmark test functions. As we can see from the table, for unimodal functions F1-F4, IGWO does better than the other four algorithms in both average optimal value and standard deviation. It also successfully reaches the theoretical optimal values of F3 and F4. This proves that IGWO has higher optimization accuracy when dealing with unimodal functions. For multimodal functions F5-F6,

IGWO also performs the best. This result shows that IGWO not only achieves high optimization accuracy but also keeps good stability in complex multimodal spaces.

Unimodal functions primarily test the local exploitation capability of algorithms. IGWO enhances its local refined search ability by reducing the search step size in the later stages of iteration through the nonlinear convergence factor. In contrast, multimodal functions contain multiple local optimal solutions. The improved opposition-based learning strategy of IGWO strengthens population diversity, effectively preventing the algorithm from falling into local optima. Consequently, IGWO demonstrates superior optimization accuracy in both types of functions.

Table 2. Average optimal values and standard deviations of the four algorithms

Function	Indicator	IGWO	GWO	SMA	AOA	MVO
F1	Avg	1.584E-228	1.457E-21	3.316E-60	1.071E-46	8.979E-02
	SD	0.000E+00	7.834E-18	9.850E-03	4.368E-46	3.653E-02
F2	Avg	7.977E-05	6.502E-04	1.688E-02	8.930E-05	2.996E-04
	SD	7.246E-05	5.089E-04	1.656E-02	7.266E-05	1.568E-03
F3	Avg	0.000E+00	1.720E-25	1.364E+00	1.485E-107	9.793E-02
	SD	0.000E+00	5.721E-25	3.046E+00	6.087E-107	4.964E-02
F4	Avg	0.000E+00	1.205E-56	1.546E-03	2.603E-103	1.327E-02
	SD	0.000E+00	3.586E-56	5.805E-03	1.426E-102	6.532E-03
F5	Avg	3.501E-03	2.946E-02	7.349E-02	6.816E-01	5.766E-03
	SD	1.819E-03	6.380E-02	9.542E-02	1.145E-01	6.197E-03
F6	Avg	1.307E-03	4.937E-03	5.059E-02	7.890E-02	2.145E-02
	SD	4.844E-03	8.567E-03	9.911E-02	8.054E-02	7.917E-02

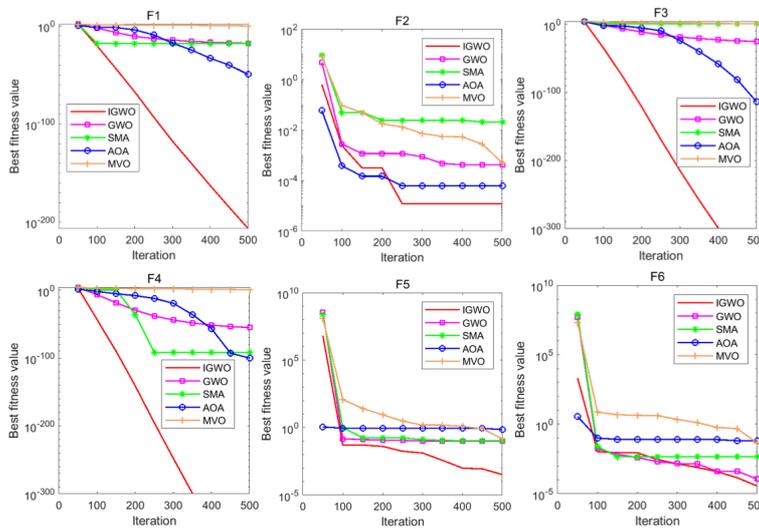


Figure 1. Convergence curves of partial comparison algorithms

Figure 1 displays the convergence curves of the five algorithms on the benchmark test functions. We can see from the curves of F1, F3 and F4 that IGWO converges faster in the early iteration stage. For F2, F5 and F6, other algorithms almost finish convergence after 100 iterations and then stay stable. But IGWO gradually gets rid of local optimal values in the later iteration stage. It finally achieves better results than the other algorithms. This result tells us two things. First, the improved opposition-based learning strategy raises the initial quality of the population. It also speeds up the

algorithm's convergence in the early stage. Second, the nonlinear convergence factor balances global exploration and local exploitation automatically. It stops the algorithm from stopping convergence in the later stage. In this way, the overall convergence efficiency of the algorithm is improved.

4.2. Application of image segmentation

To verify the actual segmentation performance of the IGWO algorithm, two representative images are selected for experiments: a stacked colored pepper image and a mandrill facial close-up image. Both images are 512×512 pixels in size. Both types of images contain multiple target regions with rich layers, which can effectively test the region division ability of the algorithm. The experiment adopts a multi-threshold segmentation strategy, and the IGWO is used to search for the optimal threshold combination to achieve accurate separation between the target and the background as well as between different targets.

The segmentation results are shown in Figure 2. The left side is the original image, and the right side is the segmentation result of IGWO. For the colored pepper image, the algorithm successfully distinguishes red bell peppers, green bell peppers, and slender green peppers in the stacked state. Each target region has a clear boundary without pixel mixing, and even the mutually occluded parts can be accurately segmented, completely retaining the contour shape of different colored peppers; for the mandrill facial image, IGWO accurately extracts key regions such as the eyes and the area around the nose, effectively divides different facial tissue parts through gray differences, retains complete detailed textures, and does not have regional adhesion or fracture problems.

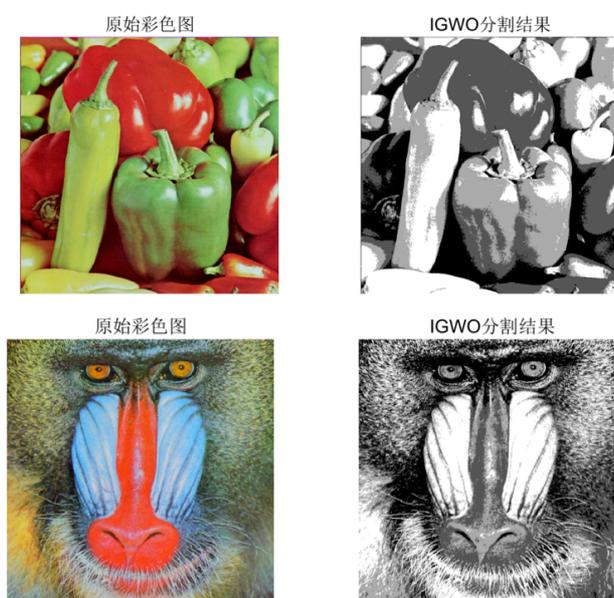


Figure 2. IGWO image segmentation results

From the visual effect, the segmentation results of IGWO achieve pixel-level accurate classification, which not only ensures the integrity of the target region but also highlights the boundary difference between regions, fully reflecting the advantage of the algorithm in complex grayscale image segmentation. This result indicates that IGWO can effectively mine image gray features, provide high-quality preprocessed data for subsequent tasks such as target recognition and feature extraction, and has strong engineering application value.

5. Conclusion

The basic GWO has problems in multi-threshold image segmentation: slow convergence, easy trapping in local optima, and poor population diversity. To solve these issues, we propose an Improved Grey Wolf Optimizer (IGWO). It combines the opposition-based learning strategy and a nonlinear convergence factor. We also build a multi-threshold segmentation model based on IGWO. The main research findings are as follows: First, we add an improved opposition-based learning strategy. It adds a random disturbance term to the traditional opposite solution, which increases population diversity. Second, we design a nonlinear convergence factor. It uses an exponential decay method to balance global exploration and local exploitation automatically. Benchmark test function experiments show that IGWO performs better. It has higher optimization accuracy and faster convergence speed in both unimodal and multimodal function optimization. Image segmentation experiments show good results too. Images segmented by IGWO have clear boundaries and well-preserved details. The algorithm can effectively handle multi-threshold segmentation tasks for real-scene images.

References

- [1] Mirjalili, S., Mirjalili, S. M., & A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- [2] Ou, Y., Qin, F., Zhou, K. Q., Yin, P. F., Mo, L. P., & Mohd Zain, A. (2024). An improved grey wolf optimizer with multi-strategies coverage in wireless sensor networks. *Symmetry*, 16(3), 286.
- [3] Zhu, Z., Sun, Z., Xie, X., & Sun, Z. (2024). Improved grey wolf optimizer based on neighborhood trust model for parameter identification of PEMFC. *International Journal of Hydrogen Energy*, 60, 769-779.
- [4] Zhu, A., Zhao, Q., Yang, T., Zhou, L., & Zeng, B. (2024). Wind speed prediction and reconstruction based on improved grey wolf optimization algorithm and deep learning networks. *Computers and Electrical Engineering*, 114, 109074.
- [5] Elsedimy, E. I., Elhadidy, H., & Abohashish, S. M. (2024). A novel intrusion detection system based on a hybrid quantum support vector machine and improved Grey Wolf optimizer. *Cluster Computing*, 27(7), 9917-9935.
- [6] Li, J., Luo, W., Bai, M., & Song, M. (2024). Fault diagnosis of high-speed rolling bearing in the whole life cycle based on improved grey wolf optimizer-least squares support vector machines. *Digital Signal Processing*, 145, 104345.
- [7] Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111, 300-323.
- [8] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376, 113609.
- [9] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2), 495-513.