Performance Analysis of Nature-Inspired Algorithms for PID Control of Electric Wheelchairs

Article · December 2024

2 authors:

Ali Amrane
University of Skikda

Zennir Youcef
University of Skikda

International Journal of Automation and Safety (IJAS)
Vol 2, No 02, December, 2024, pp. 14-19
Journal homepage: https://www.asjp.cerist.dz/en/PresentationRevue/944

Performance Analysis of Nature-Inspired Algorithms for PID Control of Electric Wheelchairs

Ali Amrane*, Youcef Zennir[2]

[1] LRPCSI Laboratory Skikda, University 20 Août 1955 of Skikda, El Hadaik, Skida, 21000, Algeria
[2] Automatic Laboratory Skikda, University 20 Août 1955 of Skikda, El Hadaik, Skida, 21000, Algeria

Corresponding Author Email: a.amrane@univ-skikda.dz

Received: December 17th 2024
Accepted: December 19th 2024
Published: December 31st 2024

Keywords:
Electric wheelchair, PID control, metaheuristic optimization, simulation, trajectory tracking.

ABSTRACT

This study presents the development and optimization of an electric wheelchair designed to improve the mobility and independence of users with disabilities. The system integrates mechanical modeling using SOLIDWORKS, kinematic simulation in SIMULINK, and advanced control strategies employing nature-inspired metaheuristic algorithms. A model-free co-simulation approach between SOLIDWORKS and SIMULINK enables realistic and adaptable system simulations without relying on predefined mathematical models. Key algorithms, including Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), and Dragonfly Algorithm (DA), are utilized to tune PID controllers for optimal trajectory tracking and system responsiveness. Simulation tests in *Coppeliasim* demonstrate the wheelchair's capability for precise navigation, and safety enhancement. The Whale Optimization Algorithm (WOA) showcased superior performance in achieving smoother and more accurate trajectory tracking. This research highlights the potential of combining simulation tools and metaheuristic optimization techniques to enhance the usability and functionality of intelligent wheelchairs, offering a practical solution to improve the quality of life for individuals with limited mobility.

1. INTRODUCTION

Mobility limitations affect millions of people worldwide, significantly impacting their independence and quality of life. Traditional manual and electric wheelchairs provide basic mobility solutions but often fail to address the specific needs of users who require advanced assistance for navigating complex environments. To bridge this gap, intelligent electric wheelchairs (IEWs) have emerged, combining mobility with autonomy and safety through advanced sensing and control technologies. These systems integrate mobile robotics, artificial intelligence, and user-friendly interfaces, transforming traditional wheelchairs into intelligent mobility aids.

An IEW is a powered wheelchair equipped with sensors and a control system to perceive its environment, enabling semi-autonomous or fully autonomous navigation. These systems can perform tasks such as obstacle avoidance, path following, and docking, enhancing user safety and independence [1, 2]. The evolution of IEWs began in the 1980s with systems like the Autonomous Vehicle for the Physically Disabled (VAHM), which introduced wall-following and obstacle avoidance capabilities [3]. Subsequent projects, such as the NavChair and the SENARIO project, have improved navigation and user interface technologies, offering solutions for both indoor and outdoor mobility [4, 5]. Despite these advancements, challenges such as sensor reliability, computational complexity, and integration costs continue to limit the accessibility of intelligent wheelchairs in real-world applications [2, 6].

This study aims to contribute to the development of more effective and accessible IEWs by leveraging modern design and simulation tools, as well as advanced control algorithms. The system presented in this work integrates a 3D model designed in SOLIDWORKS with kinematic and dynamic simulations performed in SIMULINK. This co-simulation approach eliminates the need for a predefined mathematical model, allowing for greater adaptability to real-world conditions [1, 4]. To optimize the system's trajectory tracking and responsiveness, nature-inspired metaheuristic algorithms such as Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), and Dragonfly Algorithm (DA) are employed to tune PID controllers. These algorithms have been shown to outperform traditional optimization techniques in handling nonlinear and complex systems [3, 5].

Simulation results in *Coppeliasim* demonstrate the effectiveness of this approach, with significant improvements observed in navigation accuracy. Among the tested algorithms, WOA exhibited superior performance, enabling smoother and more precise trajectory tracking. These findings underline the potential of integrating co-simulation and metaheuristic optimization techniques in the design of intelligent mobility solutions [6].

The structure of this paper is as follows: Section 2 outlines the materials and methods used, including the wheelchair design and control strategy development. Section 3 presents the results of simulation and optimization tests, followed by a discussion of the findings. Finally, Section 4 concludes the paper with recommendations for future research.

2. MATERIALS AND METHODS

Wheelchair Model Acquisition and Preparation

The electric wheelchair model used in this study was sourced from an open-source repository, eliminating the need for in-house mechanical design. The model was converted into a Unified Robot Description Format (*URDF*) file using tools available in *SOLIDWORKS*. The *URDF* conversion facilitated seamless integration into the *CoppeliaSim* simulation environment, enabling accurate representation of the wheelchair's mechanical structure and dynamics. After that the simulation of navigation was conducted in *CoppeliaSim*, a versatile platform known for its robust physics engine and support for dynamic robotic simulations. *MATLAB* was integrated with *CoppeliaSim* through the *RemoteAPI* interface, allowing real-time data exchange and co-simulation between control algorithms and the physical model.

Figure. 1 3D electric wheelchair model on CoppeliaSim.

Kinematic Modeling Of The EWH

The wheelchair's kinematic model was established based on the differential drive system, which uses two independently controlled wheels for movement and turning. The kinematic equations for linear and angular velocities were derived as follows:

Linear Velocity (v):

$$V = \frac{r}{2}(\omega_r + \omega_l) \quad (1)$$

Angular Velocity (ω):

$$\omega = \frac{r}{L}(\omega_r - \omega_l) \quad (2)$$

Where **r** is the wheel radius, **L** is the distance between the wheels, and **ω_r,ω_l** are the angular velocities of the right and left wheels, respectively.

Control System Design and parameters tuning

To achieve accurate trajectory tracking of a wheelchair, it is crucial to minimize two primary errors along the path: the distance error, **derr**, and the orientation error, **θerr** (Figure 2). The distance error represents the deviation between the wheelchair's current position and its target position, while the orientation error corresponds to the angular difference between the actual and desired orientations.

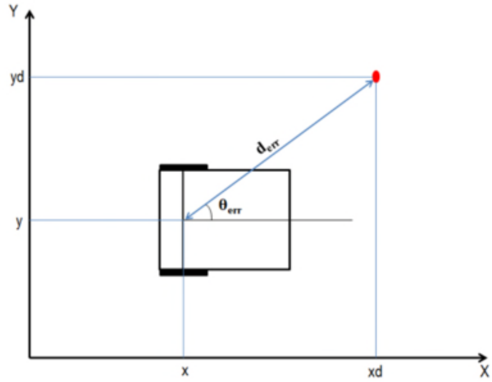

Figure. 2 Distance and orientation errors.

To address these issues, two PID controllers are employed. The first PID controller focuses on reducing the distance error by calculating the difference between the actual and desired distances and producing a control signal, **Vc**, to mitigate this error. The second PID controller takes the orientation error as input and generates a control signal, **ωc** , to correct the orientation discrepancy.

The control signals for linear velocity and angular velocity can be expressed as:

$$V_c = k_{p1} d_{err} + k_{d1}\frac{d(d_{err})}{dt} + k_{i1}\int d_{err}dt \quad (3)$$

$$\omega_c = k_{p2}\theta_{err} + k_{d2}\frac{d(\theta_{err})}{dt} + k_{i2}\int \theta_{err}dt \quad (4)$$

Here, k_{p1}, k_{i1}, and k_{d1} are the proportional, integral, and derivative gains for the distance controller, while k_{p2}, k_{i2}, and k_{d2} represent the respective gains for the orientation controller.

Metaheuristic Algorithms

A metaheuristic algorithm is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. These algorithms are designed to solve complex optimization problems by efficiently exploring the search space to find near-optimal solutions within a reasonable timeframe. Metaheuristics are particularly useful for addressing problems where traditional optimization methods are impractical due to the problem's complexity or the size of the search space.

Unlike exact optimization methods, metaheuristics do not guarantee the discovery of a globally optimal solution.

Instead, they aim to find a sufficiently good solution with less computational effort, making them suitable for large-scale and complex problems. Common examples of metaheuristic algorithms include Genetic Algorithms, particle Swarm Optimization, Ant Colony Optimization, Simulated Annealing, and Tabu Search [7].

Metaheuristics are often inspired by natural phenomena, such as biological evolution, animal behavior, or physical processes. For instance, Particle Swarm Optimization is inspired by the social behavior of birds flocking or fish schooling, while Ant Colony Optimization is based on the foraging behavior of ants. These nature-inspired algorithms have been successfully applied to a wide range of optimization problems in various fields, including engineering, economics, and logistics [8].

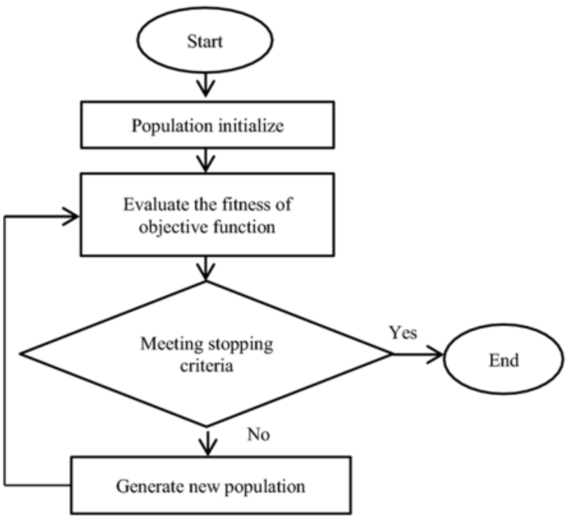

Figure. 3 General flowchart for metaheuristic algorithm [9].

To optimize the controller parameters in our simulation, many metaheuristic algorithms were employed, including:

Particle Swarm Optimization (PSO):

PSO is a computational method inspired by the social behavior of birds flocking or fish schooling. It optimizes a problem by iteratively improving candidate solutions based on their own experience and that of their neighbors[10].

Grey Wolf Optimizer (GWO):

GWO is a nature-inspired algorithm that mimics the leadership hierarchy and hunting mechanism of grey wolves in the wild. It is used for solving optimization problems by simulating the social hierarchy and hunting behavior of grey wolves [11].

Grasshopper Optimization Algorithm (GOA):

GOA is inspired by the swarming behavior of grasshoppers in nature. It is designed to solve optimization problems by modeling the grasshoppers' tendency to move towards each other while maintaining a comfortable distance, balancing exploration and exploitation in the search space [12].

Whale Optimization Algorithm (WOA):

WOA is a metaheuristic inspired by the bubble-net hunting strategy of humpback whales. It is employed to solve optimization problems by simulating the cooperative behavior of whales encircling prey and creating bubble nets to capture them [13].

Harris Hawks Optimization (HHO):

HHO is inspired by the cooperative hunting strategy of Harris' hawks. It models the surprise pounce technique used by these hawks to capture prey, providing a framework for solving complex optimization problems [14].

Dragonfly Algorithm (DA):

The Dragonfly Algorithm is a swarm intelligence-based optimization technique inspired by the static and dynamic swarming behaviors of dragonflies. It simulates the five primary principles of separation, alignment, cohesion, attraction to food sources, and distraction from enemies to explore and exploit the search space effectively [15].

3. RESULTS AND DISCUSSION

After running simulations to track the desired path using various metaheuristic algorithms, we obtained the following results :

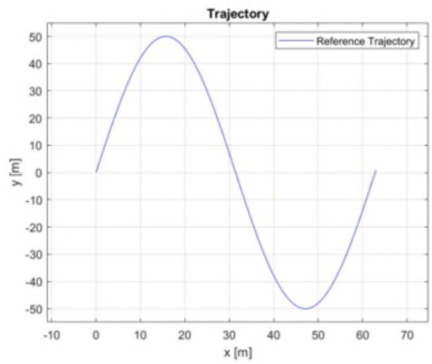

Figure. 4 The desired path.

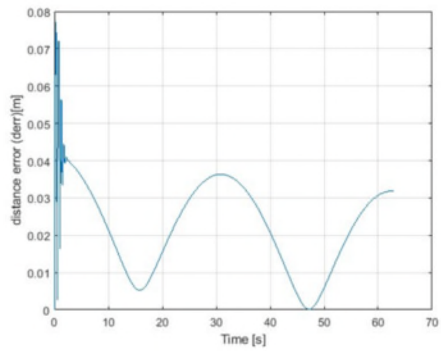

Figure. 5 Distance error of tracking with 'PSO'.

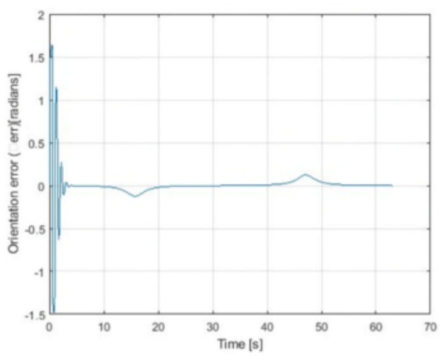

Figure. 6 Orientation error of tracking with 'PSO'.

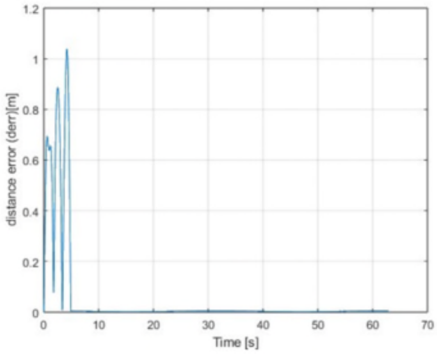

Figure. 7 Distance error of tracking with 'GJO'.

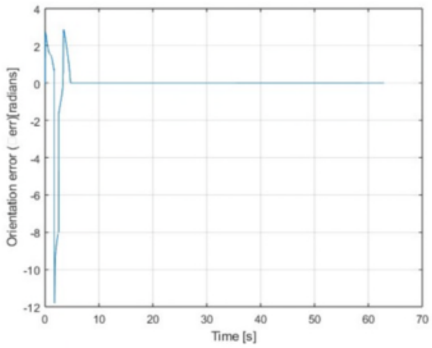

Figure. 8 Orientation error of tracking with 'GJO'.

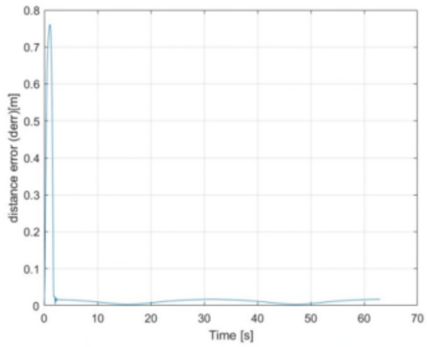

Figure. 9 Distance error of tracking with 'WOA'.

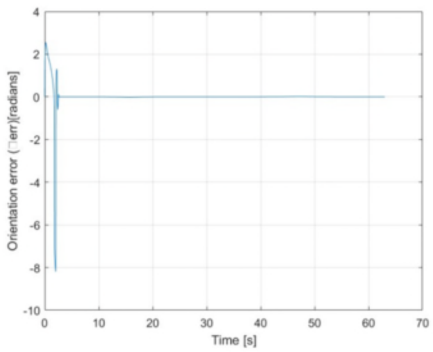

Figure. 10 Orientation error of tracking with 'WOA'.

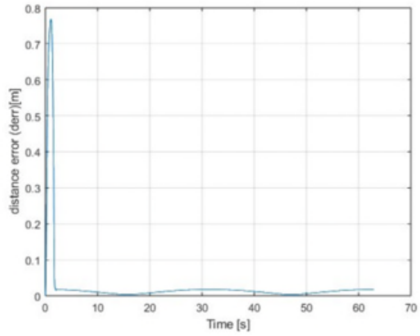

Figure. 11 Distance error of tracking with 'HHO'.

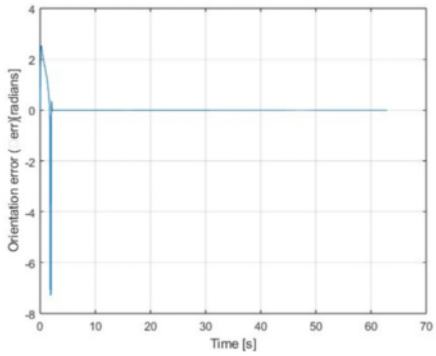

Figure. 12 Orientation error of tracking with 'HHO'.

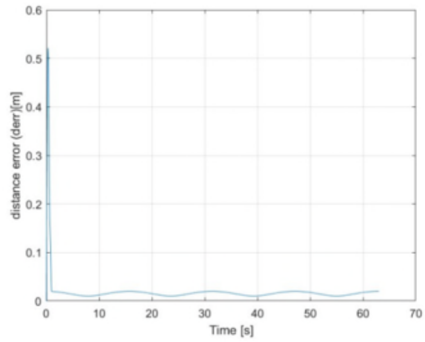

Figure. 13 Distance error of tracking with 'DA'.

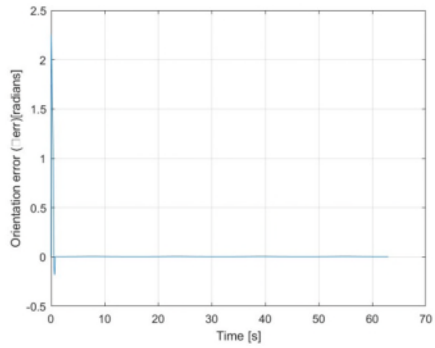

Figure. 14 Orientation error of tracking with 'DA'.

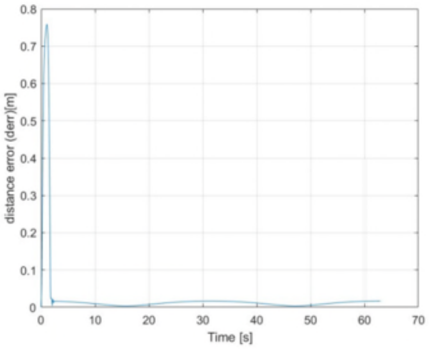

Figure. 15 Distance error of tracking with 'GWO'.

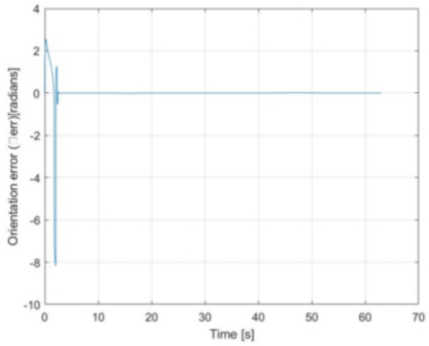

Figure. 16 Orientation error of tracking with 'GWO'.

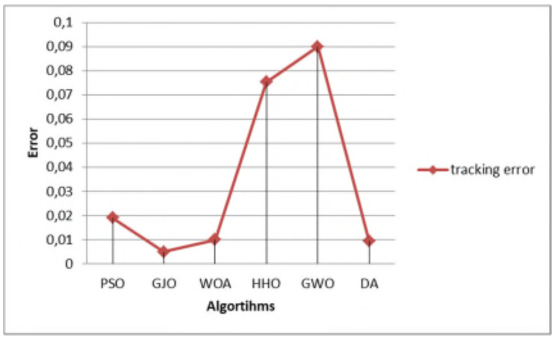

Figure. 17 Estimated error for each used algorithm.

The performance of the proposed control strategies for the intelligent electric wheelchair was evaluated using different nature-inspired metaheuristic algorithms, including PSO, GJO, WOA, HHO, GWO, and DA. The primary metric for comparison was the tracking error, which indicates the accuracy of the wheelchair in following the desired trajectory.

From the results, it is evident that the GJO (Grey Wolf Optimizer) achieved the lowest tracking error of 0.005, demonstrating its superior performance in optimizing the control system. This can be attributed to GJO's effective exploration and exploitation capabilities, which allowed it to fine-tune the PID controller parameters efficiently.

The DA (Dragonfly Algorithm) followed closely with a tracking error of 0.0095, indicating its ability to strike a balance between local and global search. Similarly, the WOA (Whale Optimization Algorithm) achieved a low error of 0.01, further showcasing its effectiveness in reducing the system's deviation from the desired path.

On the other hand, PSO (Particle Swarm Optimization) exhibited a slightly higher error of 0.019, but it still performed well compared to the other algorithms. This result highlights PSO's robustness and fast convergence properties, although it may sometimes get trapped in local optima.

In contrast, the HHO (Harris Hawks Optimization) and GWO (Grey Wolf Optimizer) algorithms demonstrated relatively higher tracking errors of 0.0753 and 0.09, respectively. The higher errors may suggest limitations in their exploration capabilities or potential convergence issues for this specific application.

4. CONCLUSIONS

This study focused on optimizing trajectory tracking for an electric wheelchair using various metaheuristic algorithms, including PSO, GJO, WOA, HHO, GWO, and DA. The simulations, conducted in a MATLAB-CoppeliaSim integrated environment, aimed to evaluate the effectiveness of these algorithms in minimizing tracking error. The results highlighted that metaheuristic algorithms, particularly GJO, DA, and WOA, significantly improved the wheelchair's trajectory tracking performance. These findings emphasize the potential of nature-inspired optimization techniques to enhance the control systems of autonomous assistive devices, contributing to their efficiency and reliability.

For future work, several directions can be explored to further enhance the system. One possible approach is to investigate hybrid metaheuristic algorithms, combining the strengths of different techniques to achieve even better performance. Additionally, incorporating adaptive control methods and reinforcement learning can enable the system to better handle dynamic environments and improve its robustness over time.

REFERENCES

[1] Erturk, E., Kim, S., & Lee, D. (2024). Driving Assistance System with Obstacle Avoidance for Electric Wheelchairs. Sensors (Basel, Switzerland), 24(14), 4644.

[2] Simpson, R. C., Levine, S. P., Bell, D. A., Jaros, L. A., Koren, Y., & Borenstein, J. (n.d.). NavChair: An assistive wheelchair navigation system with automatic adaptation. In Lecture Notes in Computer Science (pp. 235–255). Springer-Verlag.

[3] Sahoo, S. K., & Choudhury, B. B. (2024). Autonomous navigation and obstacle avoidance in smart robotic wheelchairs. In Journal of Decision Analytics and Intelligent Computing (Vol. 4, Issue 1, pp. 47–66). Scientific Oasis.

[4] P. Beattie, "SENARIO: SENsor Aided intelligent wheelchaiR navigatIOn," IEE Colloquium on New Developments in Electric Vehicles for Disabled Persons, London, UK, 1995, pp. 2/1-2/4,

[5] Bourhis, G., Horn, O., Habert, O., & Pruski, A. (2001). An autonomous vehicle for people with motor disabilities. In IEEE Robotics & Automation Magazine (Vol. 8, Issue 1, pp. 20–28). Institute of Electrical and Electronics Engineers (IEEE).

[6] Simpson, R. C. (2005). Smart wheelchairs: A literature review. In The Journal of Rehabilitation Research and Development (Vol. 42, Issue 4, p. 423). Journal of Rehabilitation Research & Development.
[7] M. Almufti, S., Ahmad Shaban, A., Arif Ali, Z., Ismael Ali, R., & A. Dela Fuente, J. (2023). Overview of Metaheuristic Algorithms. In Polaris Global Journal of Scholarly Research and Trends (Vol. 2, Issue 2, pp. 10–32). Polaris Global Research Organization Inc.
[8] Trojovský, P., & Dehghani, M. (2023). A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. In Scientific Reports (Vol. 13, Issue 1). Springer Science and Business Media LLC.
[9] Chong, H. Y., Yap, H. J., Tan, S. C., Yap, K. S., & Wong, S. Y. (2021). Advances of metaheuristic algorithms in training neural networks for industrial applications. In Soft Computing (Vol. 25, Issue 16, pp. 11209–11233).
[10] Kennedy, J., & Eberhart, R. (n.d.). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks (Vol. 4, pp. 1942–1948). ICNN'95 - International Conference on Neural Networks. IEEE.
[11] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. In Advances in Engineering Software (Vol. 69, pp. 46–61). Elsevier BV.
[12] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper Optimisation Algorithm: Theory and application. In Advances in Engineering Software (Vol. 105, pp. 30–47). Elsevier BV.
[13] Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. In Advances in Engineering Software (Vol. 95, pp. 51–67). Elsevier BV.
[14] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. In Future Generation Computer Systems (Vol. 97, pp. 849–872). Elsevier BV.
[15] Mirjalili, S. (2015). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. In Neural Computing and Applications (Vol. 27, Issue 4, pp. 1053–1073).