sensors

Article

A Novel Hybrid Harris Hawk-Arithmetic Optimization Algorithm for Industrial Wireless Mesh Networks

P. Arun Mozhi Devan [1], Rosdiazli Ibrahim [1], Madiah Omar [2], Kishore Bingi [1,*] and Hakim Abdulrab [1]

[1] Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; arun.selvam@utp.edu.my (P.A.M.D.); rosdiazli@utp.edu.my (R.I.); hakim_19001004@utp.edu.my (H.A.)

[2] Department of Chemical Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; madiah.omar@utp.edu.my

* Correspondence: bingi.kishore@utp.edu.my

Abstract: A novel hybrid Harris Hawk-Arithmetic Optimization Algorithm (HHAOA) for optimizing the Industrial Wireless Mesh Networks (WMNs) and real-time pressure process control was proposed in this research article. The proposed algorithm uses inspiration from Harris Hawk Optimization and the Arithmetic Optimization Algorithm to improve position relocation problems, premature convergence, and the poor accuracy the existing techniques face. The HHAOA algorithm was evaluated on various benchmark functions and compared with other optimization algorithms, namely Arithmetic Optimization Algorithm, Moth Flame Optimization, Sine Cosine Algorithm, Grey Wolf Optimization, and Harris Hawk Optimization. The proposed algorithm was also applied to a real-world industrial wireless mesh network simulation and experimentation on the real-time pressure process control system. All the results demonstrate that the HHAOA algorithm outperforms different algorithms regarding mean, standard deviation, convergence speed, accuracy, and robustness and improves client router connectivity and network congestion with a 31.7% reduction in Wireless Mesh Network routers. In the real-time pressure process, the HHAOA optimized Fractional-order Predictive PI (FOPPI) Controller produced a robust and smoother control signal leading to minimal peak overshoot and an average of a 53.244% faster settling. Based on the results, the algorithm enhanced the efficiency and reliability of industrial wireless networks and real-time pressure process control systems, which are critical for industrial automation and control applications.

Keywords: Arithmetic Optimization Algorithm; fractional-order Predictive PI; Harris Hawks Optimization; industrial wireless mesh networks; optimal node placement; pressure process; real-time control

check for updates

Citation: Devan, P.A.M.; Ibrahim, R.; Omar, M.; Bingi, K.; Abdulrab, H. A Novel Hybrid Harris Hawk-Arithmetic Optimization Algorithm for Industrial Wireless Mesh Networks. *Sensors* **2023**, *23*, 6224. https://doi.org/10.3390/s23136224

Academic Editors: Omprakash Kaiwartya, Sushil Kumar and Upasana Dohare

Received: 30 May 2023
Revised: 23 June 2023
Accepted: 23 June 2023
Published: 7 July 2023

1. Introduction

Network Control Systems (NCS) have become increasingly popular in mining, process, and manufacturing industries that produce various goods such as food and beverages, different metals, multiple chemicals, pulp and paper, automobiles, textiles, crude oil refineries, and power generation plants [1]. NCS monitors and controls field instruments effectively, ensuring efficient and accurate operations [2]. In recent times, WMN has gained widespread acceptance in the industry, particularly with the wide acceptance of different wireless networking protocols, namely WIA-PA, Zigbee, WirelessHART, Bluetooth, and ISA 100 wireless. One of the primary advantages of these wireless standards is eliminating clunky cabling—a costly and time-consuming process to repair and maintain [3,4]. Additionally, WMN enables the expansion of network capability to the regions where it is difficult or impossible to install the wired cables, thus making monitoring and controlling operations in remote locations possible. Moreover, using WMNs increases data reliability, reduces data loss or interference, and offers improved security measures, protecting sensitive data from unauthorized access or theft [5]. It is essential to have a thorough knowledge of several factors such as transmission power, network topology, and wireless node density. Ignoring these factors while deploying WMN routers can lead to inadequate client coverage and connectivity, substandard network range, control-loop failures, and transmission loss [6]. It is crucial to consider the technical constraints of the deployment location and the basic topology before deploying wireless mesh routers. These constraints include factors such as radio signal strength, frequency range, and interference from other wireless networks [7,8].

WMNs offer many benefits for all the process control and automation industries [9]. However, some issues must be addressed to enhance the network's performance, including network coverage, router connectivity, network traffic, compatibility, data security, etc. While comparing the outdoor regions, the indoor area is challenging while implementing the WMN for process monitoring and control because of the stochastic interferences [10]. In some outdoor areas, installations require fewer clients than in indoor regions. Furthermore, WMN can be implemented in multiple places with the availability of various control services in indoor and outdoor conditions. Determining the appropriate location for installing WMN routers for real-world applications is crucial. Therefore, optimization is essential for ensuring reliable and efficient WMN performance [11]. Optimization involves fine-tuning various parameters, such as router placement, transmission power, network topology, and routing protocols, to achieve the best possible network performance [12]. Optimizing these parameters allows the network to ensure that the WMN provides seamless coverage, reliable connectivity, and low latency, even in challenging industrial environments. Additionally, optimizing industrial WMNs can help improve network scalability and reduce maintenance costs [13].

At the same time, a feedback controller is essential in WMNs to ensure the system's stability. In addition, the feedback controllers will help the system to maintain stable and accurate control over the process variables [14]. However, to achieve optimal performance, the controller must be optimized based on the system characteristics, process model, and external disturbances of the system it controls. Optimization techniques such as trial and error, Ziegler–Nichols, and the Cohen–Coon methods are commonly used to optimize feedback controllers. These conventional techniques involve adjusting the controller's parameters to find the optimal values that result in suboptimal, unstable, and ineffective control. Thus, many researchers proposed using metaheuristic-based techniques to optimize the controller parameters for different applications [15–20].

The population-based evolutionary algorithms use mutations and crossovers, which produce new solutions using nature-inspired concepts from evolution. Swarm-based optimization techniques also employ these procedures [21]. However, they utilize two core processes for organizing initial swarms: scattering and intensifying them to specific high-potential regions [22]. Simultaneously, it should be noted that the No Free Lunch theorem affirms that no such existing metaheuristic optimization technique can be employed to solve every category of the existing issues in simulation or real-time conditions. In addition, this theorem highlights the need for a thorough understanding of the issue and the appropriate optimization technique selection to achieve optimal results [23]. Thus, a brief review of the Harris Hawk Optimization (HHO) and Arithmetic Optimization Algorithm (AOA) for different types of applications was carried out and is presented in the following sections for identifying the research gap for the proposed methodology.

1.1. Harris Hawk Optimization

Initial development of the Harris Hawk Optimization, a swarm-based optimization technique, was proposed by Heidari et al. [24]. This method is motivated by the cooperation and hunting strategies of Harris hawks in nature, which involve team action and reaction, as well as prey evasion. The main objective of HHO is to discover solutions to single-objective problems. The HHO method uses hawks' chasing actions as search agents and the prey as the best position. This approach has the potential to solve a variety of real-world optimization problems, such as engineering design, pattern and speech recognition, manufacturing optimization, bio-mechanical engineering, power production quality, feature selection, and medical image segmentation [25–29]. As with other metaheuristic techniques, HHO uses exploration and exploitation phases to identify and catch the prey. However, the HHO differs from others in the exploitation phase, where it catches the prey based on soft and hard besiege and rapid dives on soft and hard besiege movements. Detailed information on the HHO algorithm can be found in these articles [24,25,28]. As a result of the benefits provided by HHO, there has been a significant increase in HHO research, with the number of studies continuing to grow rapidly to numerous applications and creating new HHO variations.

The authors of [30] aimed to accelerate the convergence of the HHO by proposing an improved version of the algorithm that employs two strategies. The first strategy involves improving the exploration phase of HHO by incorporating opposition-based learning and logarithmic spiral techniques. The second strategy combines the Modify Rosenbrock method to enhance the HHO's local searching capability and convergence rate. The authors tested their algorithm by conducting experiments on 23 standard benchmark functions. They also compared their results with various state-of-the-art metaheuristic algorithms. In [31], two different variants of the HHO, namely binary HHO and quadratic binary HHO, were used to improve the feature selection problem at various data classification conditions. Binary HHO has a transfer function that can be either S-shaped or V-shaped, effectively converting a continuous variable into a binary one. The quadratic binary HHO is designed to improve the binary HHO by incorporating a binary quadratic model.

Dokeroglu et al. [32] developed a binary variant of the multi-objective HHO approach to address a classification task. Their novel discrete operators for exploitation (besieging) and exploration (perching) were introduced. Their study employed four machine learning methods applied to a COVID-19 dataset for binary classification problems to increase prediction accuracy. Reference [33] proposed a novel optimization algorithm, called the modified HHO, to reconfigure photovoltaic modules and disperse shadow regularly to increase the power generated. The algorithm was tested on different shade patterns with 9 × 9, 6 × 4, and 6 × 20 photovoltaic arrays based on the available simulation data. The outcomes were analyzed and compared with varying techniques of reconfiguration based on metrics such as mismatch power, fill factor, power loss, and power enhancement. Hussain et al. designed an improved variant of the HHO named long-term memory HHO that enhances the diversity of search agents by maintaining a broad search region for the exploration phase resulting in better convergence results. The proposed technique uses different engineering optimization test problems, including power flow optimization for the power generation systems [34].

The improved HHO algorithm, as introduced by Houssein et al. [35], addresses the challenge of optimizing large-scale wireless sensor networks. The approach involves utilizing Prim's shortest path algorithm to establish minimum transmission paths from the sink node to all other sensor nodes in order to reconstruct the network. The researchers have proposed energy-efficient wireless sensor networks using a hybrid Harris hawk-salp swarm optimization algorithm in [36]. The approach uses a hybrid optimization algorithm combining the HHO and salp swarm techniques to optimize the network's energy consumption. Additionally, a mobile sink strategy reduces energy consumption by allowing the destination node to move toward the sensor nodes instead of the other way around. For PID controller optimization, researchers used HHO to control the speed of the DC motor [37]. Using the HHO algorithm for tuning the controllers improves performance in terms of steady-state error, rise time, overshoot, and settling time, resulting in better efficiency, accuracy, and stability of the motor control system. An overall detailed review of the HHO literature is given in Table 1.

Table 1. Summary of different modifications in the HHO technique.

Ref.	Technique	Application	Improvements	Performance Metrics	Comparing Techniques	Results
[30]	RLHHO	Complex engineering problems	Enhancement of the exploration and exploitation	23 standard functions and 30 CEC 2014 test problems	HHO, DE, PSO, WOA, ABC, GOA, SSA, ALO, BLPSO, PPPSO, SADE, JDE, HHO-DE, DHHO/M	Good stability and prediction accuracy than other algorithms
[38]	HHO-JOS	Engineering design problems	Balancing of the exploration and exploitation phase	30 functions of CEC 2014 29 functions of CEC 2017	HHO, HHO-DO, HHO-SLO, HHO-SO	Good search space identification, better ability to escape from local optima
[31]	Binary HHO, Quadratic Binary BHO	Engineering problems	Binary variables with the feature selection improves phase transition	22 benchmark datasets	BDE, BFPA, BMVO, BSSA, GA	High classification accuracy with improved feature selection
[32]	multi-objective HHO	Real-world problems	New discrete operators for enhancing the hunting technique	13 benchmark datasets and COVID-19 patient datasets	GWO, ABC, GSA, TLBO, BOA	Higher prediction accuracy up to 11.5% with 83.8% better feature selection
[33]	MHHO	Real-world problems	Reconfigured switching techniques for each phases	9 × 9 photovoltaic array	TCT, GA, PSO, GOA	Excellent photovoltaic array reconfiguration with 12.5% on energy saving
[34]	Long-Term Memory HHO	Complex engineering problems	Enhances the diversity of search agents until search termination	10 benchmark functions and Optimal power flow problem	HHO, PSO, ABC, FA, EHO, TEO, BSDE, GOA,GWO	Up to 68% superior performance increase, minimized fuel cost, power loss and emission
[35]	HHO	Large-scale wireless sensor network	None	WSN nodes	EC, PSO, FPA, GWO, SCA, MVO, WOA	Improved network topology for for sink node placement
[36]	Hybrid HH-SS	Wireless sensor networks	A novel fitness function to control the balance between phases	WSN nodes	EAD, EASER, SEAR	Enhanced the packet delivery of 98% with 0.1 s delay
[37]	HHO	Controller tuning	None	DC motor control	HHO, GWO, SFS, IWO,	Improved steady-state error, rise time, overshoot, and settling time.
[39]	MOHHO, HMOHHO	Multi-objective Controller tuning	Hybrid strategies to promote global searching capability	7 test functions and hydraulic turbine governing system	NSGA-III, MOPSO, MOGWO	Better performance during varying load operating conditions

1.2. Arithmetic Optimization Algorithm

The AOA is a recently designed mathematics-inspired metaheuristic optimization technique that draws inspiration from the distribution behavioral nature of arithmetic functions from the mathematics proposed by Abualigah et al. [40]. It uses basic arithmetic operations, such as addition, subtraction, multiplication, and division, as the primary operator to explore the desired objective function and the optimal solution. AOA is also a population-based optimization technique, a set of solutions, called individuals, are generated randomly and evaluated based on their fitness concerning the considered problem. The best-fit individuals are chosen and subjected to various mathematical operations to generate new individuals based on the desired objective values in both phases. These new individuals are then evaluated for their fitness and the process is repeated until the desired optimization level is reached [41].

A key advantage of AOA is that it can prevent getting stuck in local minima, a common issue many optimization algorithms face. AOA achieves this by employing a mechanism called elitism, which ensures that the best solutions in the population are always preserved and passed over to the next generation, allowing the algorithm to explore different areas of the solution and converge to the global minimum efficiently. Premkumar et al. [42] proposed a newly developed approach called Multi-Objective AOA (MOAOA) to solve real-world constrained multi-objective optimization problems. The latest version of the AOA algorithm has been improved by adding two new functions—elitist non-dominance sorting and a crowding distance-based mechanism. This new algorithm has been tested on 35 optimization problems with constraints and five test problems without constraints. Its effectiveness has been compared with four other advanced multi-objective optimization techniques. Additional performance metrics are also considered to evaluate the MOAOA for higher accuracy and effective convergence, in which MOAOA achieved superior performance.

In [43–45], the researchers developed an enhanced form of artificial neural network which uses a two-step process to identify, locate, and measure the extent of damage in plate structures made of functionally graded materials, petroleum products, and other electrical systems. In the first stage, a damage indicator based on the frequency response function is utilized to forecast which components of the material have been affected. In the second stage, the networks are employed to quantify the damage and it eliminates healthy elements from the numerical model and utilizes information from the defective components to estimate the degree of damage. Abualigah et al. [46] proposed a new method for various multilevel thresholding in data analysis and image segmentation using the AOA and differential evolution technique. The DAOA algorithm was assessed using traditional examination images from nature and CT COVID-19 images. The accuracy of the segmented images was measured using the peak signal-to-noise ratio and structural similarity index test.

Zheng et al. [47] developed a new hybrid algorithm called DESMAOA, combining two meta-heuristic algorithms—the Slime Mold Algorithm (SMA) and the AOA. The DESMAOA algorithm was designed to improve the optimization capability of the existing techniques. The SMAOA algorithm is first used to improve the SMA algorithm, and then two strategies from SMA and AOA are combined to create the DESMAOA. In [48], a new method is developed to control an automatic voltage regulator using a robust Model Predictive Controller (MPC). The suggested approach was crafted to address the challenge of uncertain automatic voltage regulation parameters. Frequency domain conditions are obtained through the Hermite–Biehler theorem to ensure stability in the face of perturbations. The tuning of MPC parameters is performed using the AOA technique while accommodating stability constraints. A time-domain objective is established to optimize the voltage regulator's performance by minimizing voltage peak overshoot and fastening the process settling.

In [49], the authors introduced a forced switching mechanism for AOA termed IAOA that helps search agents to switch between various local optima. The effectiveness of the IAOA was tested on multiple benchmark functions and real-world test problems and the results showed that it outperforms other optimization algorithms in most cases. Different types of metaheuristic optimization were compared for the various engineering problems in [50]. The researchers proposed an enhanced version of the AOA called nAOA. The nAOA technique uses logarithmic and exponential mathematical operators to improve algorithm performance. The overall summary of the AOA technique is given in Table 2.

Table 2. Summary of different modifications of the AOA technique.

Ref.	Technique	Application	Improvements	Performance Metrics	Comparing Techniques	Results
[42]	MOAOA	Complex engineering problems	Non-dominated sorting technique and maintains the diversity among the obtained best values	35 Real-World Multi-objective Problems and 5 un-constrained problems	NSGWO, MOMVO, MOALO, MOSMA	Best coverage, computational cost, with high efficiency in problem solving
[43]	IANN-AOA, IANN-BCMO	Engineering problems	None	FGM metal and ceramic materials	AOA, BCMO	Improved damage prediction with high accuracy and better damage quantification
[46]	DAOA	Real-world problems	Improved the convergence ability of AOA with better balancing between the phases	Nature and CT COVID-19 images	AO, WOA, SSA, AOA, PSO, MPA, DE	Improved results than comparing techniques, effective-segmented images, PSNR, and SSIM values
[47]	DESMAOA	Complex engineering problems	Random contraction, subtraction and addition strategies to expand the search regions to increase accuracy	23 benchmark functions and 3 engineering problems	SMA, AOA, GWO, WOA, SSA, MVO, PSO	Outperforms other optimization algorithms in terms of speed and accuracy with better local minima switching
[48]	AOA	MPC Controller tuning	Reconfigured switching techniques for each phases	Automatic voltage regulator	ABC, FSA, MOEO, NSGA-II	Good performance in minimizing voltage maximum overshoot and settling time
[49]	IAOA	Complex engineering problems	Forced switching mechanism to avoid local minima trapping	23 benchmark functions and 10 CEC2020 test functions	PSO, SCA, GWO, WOA, SSA, MVO	Better optimization performance in terms of speed and accuracy with better local minima switching
[50]	nAOA	Real-world engineering applications	Distributive mathematical operators to enhance the performance	30 benchmark functions and 3 engineering problems	SA, SCA, GWO, AOA, CPSOGSA, GSA	Better performance during different test problems by position relocation
[51]	AOA	Fuzzy-PID Controller tuning	None	Interconnected power system	TBLO, AOA	Desired performance in variable load, uncertain and offers better contingency
[52]	dAOA	Real-world engineering applications	Chaotic theory to improve the local minima relocation and convergence speed	Proton exchange membrane fuel cell	COA, ALO, WOA, AOA	Reliable and accurate with less training error for the developed machine learning model
[53]	AOAGA	Real-world engineering and complex datasets	Hybridizing GA to improve the searching abilities without compromising on algorithm speed	Several benchmark and two real-world problems	SMA, HHO, SA, MVO, SSA, MFO, GOA, PSO, GWO	Provided a good balance between the number of selected features and classification accuracy

Some of the main contributions of this research article are listed as follows:

1. The primary motivation of HHAOA is to solve the problem of placing the routers optimally to achieve adequate coverage and connectivity, along with finding the best parameter for the Fractional-Order Predictive PI (FOPPI) controller.
2. The proposed HHAOA technique makes use of all the arithmetic operator combinations, enabling smooth and efficient relocations between local minima without getting stuck. This approach effectively minimizes computational complexity, resulting in a more streamlined process outcome.

3. To achieve a better convergence rate and reach the desired optimal solution location, the proposed optimization was simulated and validated using 33 different benchmark functions.
4. The proposed technique was implemented to enhance nodes' placement and mitigate congestion in wireless mesh networks (WMNs). The results reveal that the HHAOA technique is highly effective in reducing the number of node placements, leading to significant cost savings and improved network congestion compared to alternative algorithms.
5. The HHAOA-optimized FOPPI controller was implemented on a real-time pressure process plant to validate our proposed optimization algorithm, and the results show a better performance than conventional controllers.

2. Proposed Hybrid Harris Hawks-Arithmetic Optimization Algorithm

The primary focus of this study was developing a hybrid HHAOA technique to enhance the coverage and connectivity of a WMN, minimize network congestion, and optimize parameters for the FOPPI controller [54]. The proposed technique hybridizes the HHO and AOA techniques. Despite the impressive capabilities of HHO and AOA, some drawbacks need to be addressed, including the risk of premature convergence, becoming trapped in multiple local optima, and its phase-switching mechanism. The proposed HHAOA approach improves convergence behavior, the position-switching mechanism, and solution quality. When implementing the hybrid method for searching, the process becomes notably more thorough and effective. This is due to the ability to navigate throughout the desired search area and avoid becoming trapped in local optima. As a result, a diverse range of potential solutions can be generated, increasing the likelihood of finding the optimal outcome. Furthermore, the HHAOA will be used to optimize the placement of WMN routers via simulation. Additionally, real-time experiment experimentation on the pressure process plant using the HHAOA-optimized FOPPI controller was compared with traditional techniques.

2.1. Proposed Hierarchical Structure

The proposed HHAOA hierarchical structure is depicted in Figure 1. The system comprises a top layer (primary layer) with M HHO search agents and a bottom layer (secondary layer) with groups containing an N AOA population. The AOA execution in the bottom layer initiates the process of updating the search agents' positions. In order to find the most optimal solution, it is crucial to ensure that the positions of all search agents in the upper layer are updated with the best solution discovered by the corresponding group in the lower layer. New equations can be formulated for both the exploitation and exploration phases, leading to an even more effective solution. This approach allows for a comprehensive problem analysis and significantly improves the overall outcome.

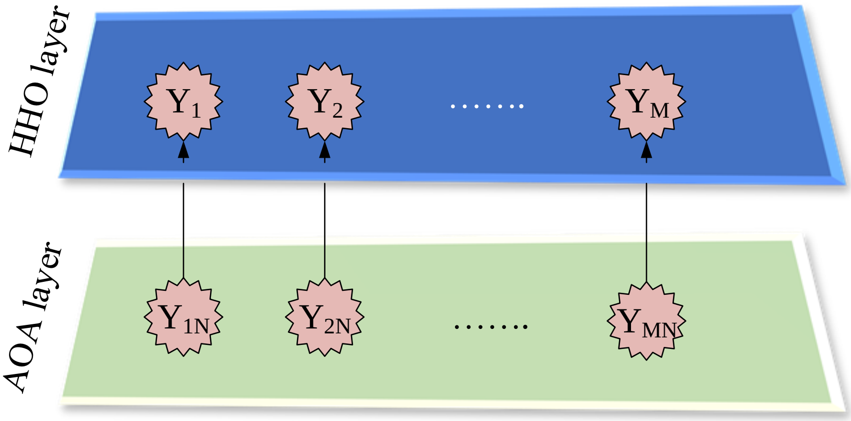


Figure 1. Hierarchy form of the proposed HHAOA optimization algorithm.

2.2. Exploration

This section proposes an exploration mechanism for hybridizing the HHO and AOA algorithms. The algorithm draws inspiration from the hunting behavior of Harris hawks, capitalizing on their exceptional eyesight to effectively track and recognize prey. However, the prey may not be immediately visible in some cases, prompting the hawks to patiently wait, observe, and monitor the objective location for several hours until detecting a potential target.

The algorithm expertly utilizes Harris hawks to represent potential solutions, with the ultimate goal of comparing the best solution to the ideal outcome. The algorithm flawlessly imitates the hawks' actions by strategically placing them in different locations and using various approaches until the desired solution is found. Each strategy is meticulously employed in equal measure to ensure maximum efficiency. The hawks rest closer to the prey (rabbit) based on the positions of other family members during $q < 0.5$. When the condition is met with $q \geq 0.5$, the hawks will randomly perch on tall trees within the range of their family members. Instead of selecting the random tall trees, the best position nearest to the rabbit will be identified using the AOA algorithm, which will further enhance the chances of reaching and identifying the target much faster. Therefore, the exploration strategy of the proposed HHAOA is obtained using the equation given below.

$$Y_{j+1}^{i} = \begin{cases} y_{\text{rand}} - r_1 \mid y_{\text{rand}} - 2r_2(y_j \div (\text{MOP} + \epsilon) \times ((UB_j \\ -LB_j) \times \ \mu + LB_j) \mid, q \geq 0.5 \ \& \ c \geq 0.5 \\ y_{\text{rand}} - r_1 \mid y_{\text{rand}} - 2r_2(y_j \times \text{MOP} \times ((UB_j - LB_j) \\ \times \ \mu + LB_j) \mid, q \geq 0.5 \ \& \ c < 0.5 \\ y_{\text{B}} - y_m - r_3\left[LB_j + r_4\left(UB_j - LB_j\right)\right], q < 0.5, \end{cases} \tag{1}$$

where Y_i^{j+1} describes the location of t-th solution in the top layer (HHO) corresponding to the j-th search solution in the AOA layer. j represents the current iteration of the algorithm. r_1, r_2, r_3, r_4, c, and q are the random numbers that lies in the span of [0,1]. y_{rand}, y_B, y_m, UB, LB, MOP are the random hawk selected, the best location obtained so far in the present iteration, average mean of the hawk's position, upper and lower bound ranges of the variables, and the math-optimizer probability coefficient. The MOP and average mean of the hawk's location values are calculated based on the equation given below.

$$\text{MOP}(j) = 1 - \left(\frac{t}{T}\right)^{1/\alpha} \tag{2}$$

$$y_m(j) = \frac{1}{N}\sum_{j=1}^{N} y_i(j), \tag{3}$$

where α is a crucial parameter factor that determines the precision of the exploitation phase, t, and T represent the current and the highest possible number of iterations, respectively. The position of every hawk in iteration j is represented by $y_i(t)$, and the total number of hawks is denoted by N.

2.3. Transition Stage

The HHO algorithm can alternate between exploration and exploitation modes depending on the amount of energy the prey has left while attempting to escape. The prey's energy decreases notably during its escape. This energy is determined by using the equation below:

$$E = 2E_0\left(1 - \frac{t}{T}\right), \quad t = \{1, 2, 3, \ldots, T\}. \tag{4}$$

The escaping energy of the prey is represented as E and the maximum number of iterations is represented by T with the current iteration being represented as t. Additionally, E_0 indicates the initial energy state of the prey, which randomly changes within the interval of (-1,1) at each iteration in HHO. When E_0 decreases from 0 to -1, the prey is physically exhausted, while an increase in E_0 from 0 to 1 demonstrates that the prey is becoming more powerful. During the iterations, the escaping energy (E) gradually decreases. When the escaping energy is ∣E∣ $\geq$ 1, the hawks explore various locations to search for the prey, thereby performing the exploration phase. Conversely, when ∣E∣ < 1, the algorithm focuses on exploiting the neighbourhood of the solutions.

2.4. Exploitation

During the attacking phase, Harris hawks employ a surprise pounce maneuver to take down their intended prey, which is identified in the previous stage. However, the prey may attempt to evade danger, resulting in various chasing styles observed in actual real-life scenarios.

Based on the prey's fleeing behavior and the Harris hawks' pursuit strategies, the HHO proposes four potential approaches to simulate the attacking phase. As preys always try to escape threatening situations, they have a chance (r) of successfully escaping ($r < 0.5$) or not ($r \geq 0.5$) before the surprise pounce. Hawks employ a hunting technique known as hard or soft encirclement to capture their prey which involves surrounding the prey from various angles, considering its energy level. The hawks gradually move closer to their prey to increase their chances of a successful surprise attack in practical settings. As the fleeing prey loses energy over time, the hawks intensify the besiege process to capture the exhausted prey effortlessly. Parameter E models this strategy and enables the HHO to alternate between soft and hard besiege processes. Similar to the exploration, the AOA will reach the solution (prey) closer in this phase.

2.4.1. Soft Besiege

The rabbit retains sufficient energy to flee through random and deceptive jumps during $r \geq 0.5$ and ∣E∣ $\geq$ 0.5. However, its efforts prove to be unsuccessful, and the Harris hawks gradually encircle it, causing the rabbit to become increasingly tired before ultimately launching a surprise attack. The following Equation (5) represents this surprise attack of HHAOA:

$$Y^{i}_{j+1} = \begin{cases} y_{\mathrm{B}} - [F_A] - E \mid 2(1 - r_2)y_{\mathrm{B}} - [F_A] \mid, c < 0.5, \\ y_{\mathrm{B}} - [F_B] - E \mid 2(1 - r_2)y_{\mathrm{B}} - [F_B] \mid, c \geq 0.5, \end{cases} \tag{5}$$

where

$$F_A = y_j - \mathrm{MOP} \times \left((UB_j - \mathrm{LB}_j) \times \mu + LB_j\right)$$
$$F_B = y_j + \mathrm{MOP} \times \left((UB_j - \mathrm{LB}_j) \times \mu + LB_j\right).$$

2.4.2. Soft Besiege with Progressive Rapid Dives

When the rabbit's energy is sufficient (∣E∣ $\geq$ 0.5) for successful escape and the hawk's movement is rapid (r < 0.5), a gentle ambush is carried out before the surprise attack. This approach is more intelligent than the previous method. To mathematically model the prey's escape patterns and the predator's leapfrog movements, the HHAOA algorithm utilizes the idea of incorporating the Levy flight (LF) pattern. LF simulates the erratic zigzag movements of prey (especially rabbits) in the fleeing phase and hawk's irregular, sudden, and rapid dives as they approach the targeted prey. The LF mechanism used here is given below.

$$LF(x) = \frac{\mu \times \delta}{|\nu|^{\frac{1}{\zeta}}} \times 0.01, \delta = \left(\frac{\Gamma(1 + \zeta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\zeta}{2}\right) \times \zeta \times 2^{\left(\frac{\zeta-1}{2}\right)}} \right)^{\frac{1}{\zeta}}, \tag{6}$$

where μ and ν are random numbers that lie in the range of inside [0,1], ζ is a constant value that is set at 1.5.

During the soft besiege, hawks perform multiple rapid dives around the rabbit, constantly adjusting their position and direction to match the deceptive movements of the prey.

This mechanism is also observed in other competitive situations in nature. The movement of the hawks in this phase is obtained using the following position update rule given in Equation (7).

$$Y_{j+1}^{i} = \begin{cases} Z \text{ if } F(Z) < F\left(y_j\right) \& \\ y_j = \begin{cases} F_A, c < 0.5 \\ F_B, \ c \geq 0.5 \end{cases} \\ X\, if\ F(X) < F\left(y_j\right) \& \\ y_j = \begin{cases} F_A, c < 0.5 \\ F_B, \ c \geq 0.5, \end{cases} \end{cases} \tag{7}$$

where

$Z = X + S \times LF(D)$
$X = y_B - E \mid Jy_B - y_m \mid ; J = 2 - 2r_2$
S = Random vector in the dimension D $(1 \times D)$.

2.4.3. Hard Besiege

When the conditions meet $r \geq 0.5$ and $\mid$E$\mid$ < 0.5, the rabbit will become exhausted and have less energy to flee. As a result, the Harris hawks will attempt to capture the rabbit by surrounding it aggressively and launching surprise attacks. The following equation can describe this intense attack phase.

$$Y_{j+1}^{i} = \begin{cases} y_{\text{B}} - E \mid y_{\text{B}} - [y_j - \text{MOP} \times ((UB_j - \text{LB}_j) \times \\ \mu + LB_j)] \mid, \quad c < 0.5 \\ y_{\text{B}} - E \mid y_{\text{B}} - [y_j + \text{MOP} \times ((UB_j - \text{LB}_j) \times \\ \mu + LB_j)] \mid, \quad c \geq 0.5. \end{cases} \tag{8}$$

2.4.4. Hard Besiege with Progressive Rapid Dives

If $\mid$E$\mid$ < 0.5 and $r < 0.5$, the rabbit lacks the necessary energy to flee and must resort to a sudden, aggressive dive technique to catch its prey off guard. This maneuver places the prey in a position similar to a soft dive. Still, the hawks work to minimize the distance between themselves and the prey's average location while attempting to escape. The movement of this phase is obtained as follows:

$$Y_{j+1}^{i} = \begin{cases} Z \text{ if } F(Z) < F(y_j) \& \\ y_j = \begin{cases} F_A, c < 0.5 \\ F_B, \ c \geq 0.5 \end{cases} \\ X\, if\ F(X) < F(y_j) \& \\ y_j = \begin{cases} F_A, c < 0.5 \\ F_B, \ c \geq 0.5 \end{cases} \end{cases} \tag{9}$$

where,

$Z = X + S \times LF(D)$
$X = y_B - E \mid Jy_B - y_j \mid$.

The HHAOA proposal consists of two distinct phases: exploration and exploitation. The initial phase is responsible for finding a new position to enhance the optimization process. At the same time, the latter stage utilizes four different strategies to maximize the position update to converge effectively. Detailed information about when to switch between these two phases is presented in Table 3.

Table 3. Different HHAOA conditions and their respective cases in exploration and exploitation phases.

Phase	Technique	Primary Layer	Secondary Layer	Remark
Exploration		$\lvert E\rvert \geq 1$ & $q \geq 0.5$	$c \geq 0.5$	Condition 1 of Equation (1)
		$\lvert E\rvert \geq 1$ & $q \geq 0.5$	$c < 0.5$	Condition 2 of Equation (1)
		$\lvert E\rvert \geq 1$ & $q < 0.5$		Condition 3 of Equation (1)
Exploitation	Soft besiege	$\lvert E\rvert < 1$ & $\lvert E\rvert \geq 0.5$ & $r \geq 0.5$	$c < 0.5$	Condition 1 of Equation (5)
		$\lvert E\rvert < 1$ & $\lvert E\rvert \geq 0.5$ & $r \geq 0.5$	$c \geq 0.5$	Condition 2 of Equation (5)
	Soft besiege-rapid dives	$\lvert E\rvert < 1$ & $\lvert E\rvert \geq 0.5$ & $r < 0.5$	$c < 0.5$ or $c \geq 0.5$	Condition 1, case 1 or 2 of Equation (7)
		$\lvert E\rvert < 1$ & $\lvert E\rvert \geq 0.5$ & $r < 0.5$	$c < 0.5$ or $c \geq 0.5$	Condition 2, case 3 or 4 of Equation (7)
	Hard besiege	$\lvert E\rvert < 1$ & $\lvert E\rvert < 0.5$ & $r \geq 0.5$	$c < 0.5$	Condition 1 of Equation (8)
		$\lvert E\rvert < 1$ & $\lvert E\rvert < 0.5$ & $r \geq 0.5$	$c \geq 0.5$	Condition 2 of Equation (8)
	Hard besiege-rapid dives	$\lvert E\rvert < 1$ & $\lvert E\rvert < 0.5$ & $r < 0.5$	$c < 0.5$ or $c \geq 0.5$	Condition 1, case 1 or 2 of Equation (9)
		$\lvert E\rvert < 1$ & $\lvert E\rvert < 0.5$ & $r < 0.5$	$c < 0.5$ or $c \geq 0.5$	Condition 2, case 3 or 4 of Equation (9)

2.5. Pseudocode of Proposed Algorithm

The application of HHAOA for the industrial wireless mesh networks (WMN) and FOPPI controller parameter optimization is comprehensively explained in the pseudocode for the proposed technique presented in Algorithm 1. The respective implementation of the HHAOA using the flowchart is illustrated in Figure 2.

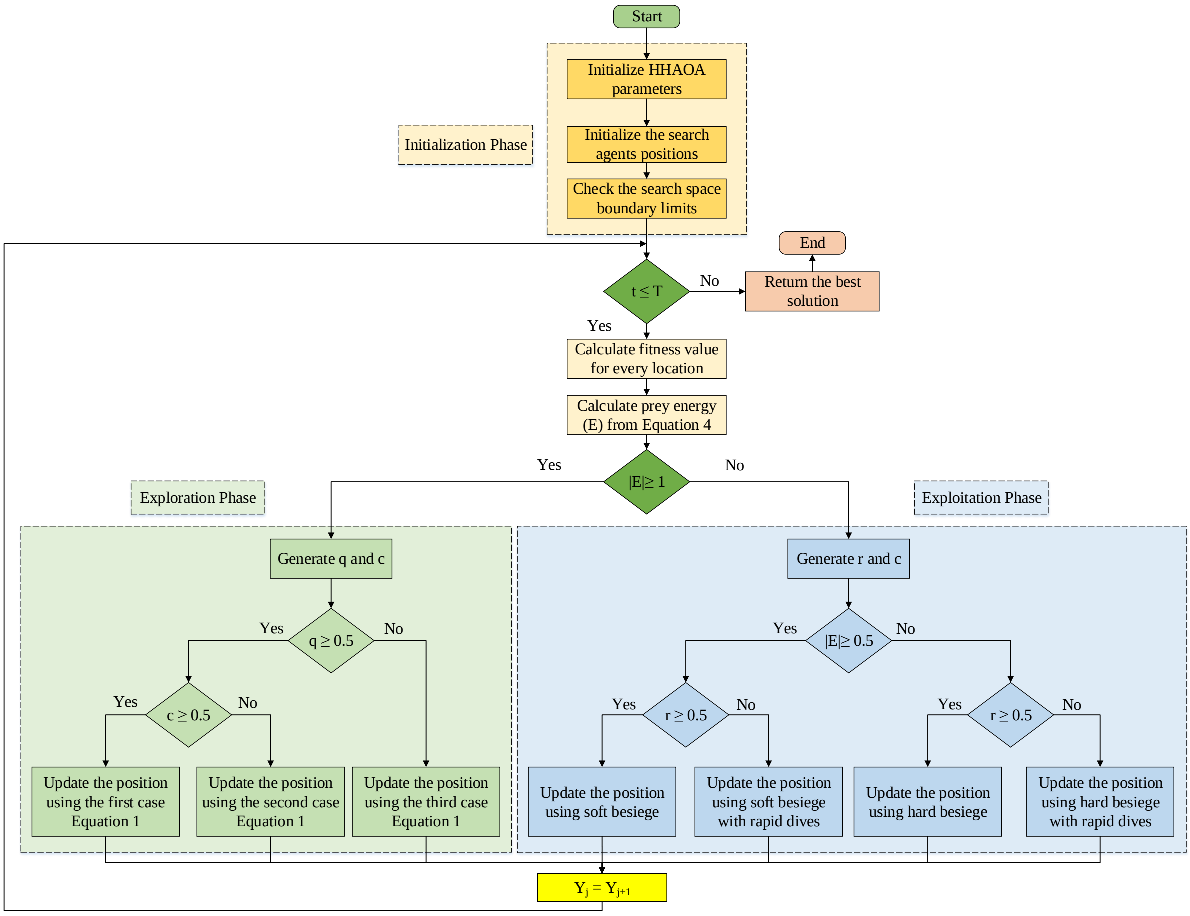

Start
Initialize HHAOA parameters
Initialization Phase
Initialize the search agents positions
Check the search space boundary limits
End
t ≤ T
No
Return the best solution
Yes
Calculate fitness value for every location
Calculate prey energy (E) from Equation 4
Yes
|E|≥ 1
No
Exploration Phase
Exploitation Phase
Generate q and c
Generate r and c
Yes
q ≥ 0.5
No
Yes
|E|≥ 0.5
No
Yes
c ≥ 0.5
No
Yes
r ≥ 0.5
No
Yes
r ≥ 0.5
No
Update the position using the first case Equation 1
Update the position using the second case Equation 1
Update the position using the third case Equation 1
Update the position using soft besiege
Update the position using soft besiege with rapid dives
Update the position using hard besiege
Update the position using hard besiege with rapid dives
$Y_j = Y_{j+1}$

Figure 2. Flowchart of the proposed HHAOA technique.

2.6. Algorithm Complexity

It is crucial to note that the complexity of the proposed HHAOA algorithm is heavily influenced by three critical processes: initialization of the transition stage, fitness function evaluation, and position updating of Hawks. The initialization process has a complexity of C(n) when there are n hawks. Updating the location vector of all hawks involves searching for the optimal location, which can be complex. This involves C(T × n) + C(T × n × D) operations, where T represents the maximum number of iterations and D is the dimension of the current particular problem in hand. Therefore, it is imperative to understand that the computational complexity of HHAOA can be expressed as C(n × (T + TD + 1)).

Algorithm 1 Pseudocode of HHAOA technique.

```
Input: Random WMN routers positions and manually calculated FOPPI controller parameters.
Output: Optimal WMN connection and FOPPI parameters
Initialize the search agents positions $y_j$ ($j$= 1, 2, 3, ..., N)
Check search space boundary and initiate the $t \leq T$ case
while ($t \leq$ T) do
    Attain the initial solution $y_B$
    for (All hawks ($y_j$)) do
        Amend the position update
        Use (4) to update the E
        Phase: Exploration
        if $|E| \geq 1$ & $q \geq 0.5$ then
            if $c \geq 0.5$ then
                Update the solution using Equation (1) condition 1
            else if $c < 0.5$ then
                Update the solution using Equation (1) condition 2
            end if
        else if $|E| \geq 1$ & $q < 0.5$ then
            Update the solution using Equation (1) condition 3
        end if
        Phase: Exploitation
        if $|E| < 1$ then
            if ($|E| \geq 0.5$ and r $\geq 0.5$) then
                if $c < 0.5$ then
                    Solution update using Equation (5) condition 1
                else if $c \geq 0.5$ then
                    Solution update using Equation (5) condition 2
                end if
            else if ($|E| \geq 0.5$ and r $< 0.5$) then
                if $c < 0.5$ or $c \geq 0.5$ then
                    Solution update using Equation (7) condition 1
                else if $c < 0.5$ or $c \geq 0.5$ then
                    Solution update using Equation (7) condition 2
                end if
            else if ($|E| < 0.5$ & r $\geq 0.5$) then
```

Algorithm 1 *Cont.*

```
                **if** $c < 0.5$ **then**
                    Solution update using Equation (8) condition 1
                **else if** $c \geq 0.5$ **then**
                    Solution update using Equation (8) condition 2
                **end if**
            **else if** $(|E| < 0.5 \;\&\; \mathrm{r} < 0.5)$ **then**
                **if** $c < 0.5$ or $c \geq 0.5$ **then**
                    Solution update using Equation (9) condition 1
                **else if** $c < 0.5$ or $c \geq 0.5$ **then**
                    Solution update using Equation (9) condition 2
                **end if**
            **end if**
        **end if**
    **end for**
    $t = t + 1$
**end while**
**Return** $Y_B$ for optimal WMN connectivity with all the routers and optimal FOPPI
controller parameters for real-time pressure process plant.
```

3. Problem Formulation

The main focus of this subsection is the implementation of the proposed HHAOA approach, which addresses the problem of optimal placement of WMN routers and finding the optimal FOPPI controller parameters. The effectiveness of the HHAOA method was compared against different metaheuristic optimization techniques, such as AOA, MFO, SCA, GWO, WOA, and HHO. The benchmark algorithms and the proposed method were evaluated using MATLAB software and the results were analyzed based on the original positions of the clients generated using the Atarraya simulator [25].

3.1. Industrial Wireless Mesh Networks

Proper planning is essential when implementing a wireless mesh network to ensure the optimal placement of mesh routers. Determining the ideal locations and the number of routers needed to achieve complete coverage and connectivity is crucial. Our approach assumes that mesh clients remain stationary and their locations are predetermined, as the placement of mesh routers in an industrial environment depends on the clients' locations. Despite this, finding the optimal placement of WMN routers in a timely and precise manner remains a computational challenge. To address this issue, we made certain assumptions about the placement of mesh routers in a wireless mesh network.

- The devices connected to the mesh network stay in one place within a desired 2D region;
- Each router in the network has the same range for transmitting signals (Identical transmission);
- The routers are connected based on their transmission range to ensure connectivity.

The location of mesh clients establishes the ideal positioning of mesh routers and it is represented by $M = \{L(p_1, q_1), L(p_2, q_2), L(p_3, q_3), \ldots, L(p_n, q_n)\}$. It is crucial to remember that a network $\mathbb{N}$ may not be entirely linked, meaning it could consist of multiple distinct subgraphs. In order to improve the connectivity of the WMN, it is important to enlarge the largest subgraph in the network until it reaches its maximum capacity. A network is considered fully linked when all mesh routers are interconnected. The network coverage for the clients can be obtained using the following equation.

$$\Omega(N) = \sum_{d=0}^{c} \Delta_d \tag{10}$$

where, $\Delta_d = \begin{cases} 1 \text{ if the router encloses the client d} \\ 0 \text{ Otherwise.} \end{cases}$

Similarly, the network connectivity between the WMN routers will be measured using the expression below.

$$\Phi(N) = \max_{d \in \{1,\ldots,h\}} |N_d|. \tag{11}$$

The primary goal of the HHAOA is to enhance the efficiency of the WMN, aiming to maximize the network range and connection while minimizing the usage of mesh routers to alleviate network traffic. Moreover, when establishing the fitness function, the parameters $\Phi\,(N)$, representing the level of connectivity among routers, and $\Omega\,(N)$, stating the network coverage, were considered. The weighted sum technique is a straightforward approach for streamlining a multi-objective problem. By expertly combining each objective and assigning a user-determined weight, a single-objective problem is formed.

$$F_j = \xi \cdot \left(1 - \frac{\Phi(N)}{n}\right) + (1 - \xi) \cdot \left(1 - \frac{\Omega(N)}{m}\right); \tag{12}$$

where, the m and n are the numbers of mesh network clients and routers, respectively. ξ is a weight-adjusting coefficient parameter that lies in the range of (0,1). Furthermore, the metrics for the statistical comparison will be carried out in terms of *mean*, *standard deviation* (std.), and *best* and *worst*.

3.2. Pressure Process Control

Figure 3 illustrates the schematics of the real-time pressure process plant, which operates in real-time. The primary buffer tank, VL 202, was designed to withstand up to 10 Bar of pressure from the centralized air compression system that supplies air to it. The pressure inside the tank can be controlled using the hand valve, HV 202, while the process control valve, PCV 202, ensures that the pressure inside the tank is maintained at the desired level. The pressure transmitter, PT 202, is utilized to measure the pressure, which is then converted to digital voltage signals ranging from 0 to 5 V. These signals are sent to the pressure indicating controller PIC 202, which transmits the control signal to the host PC via I/O interface boards.

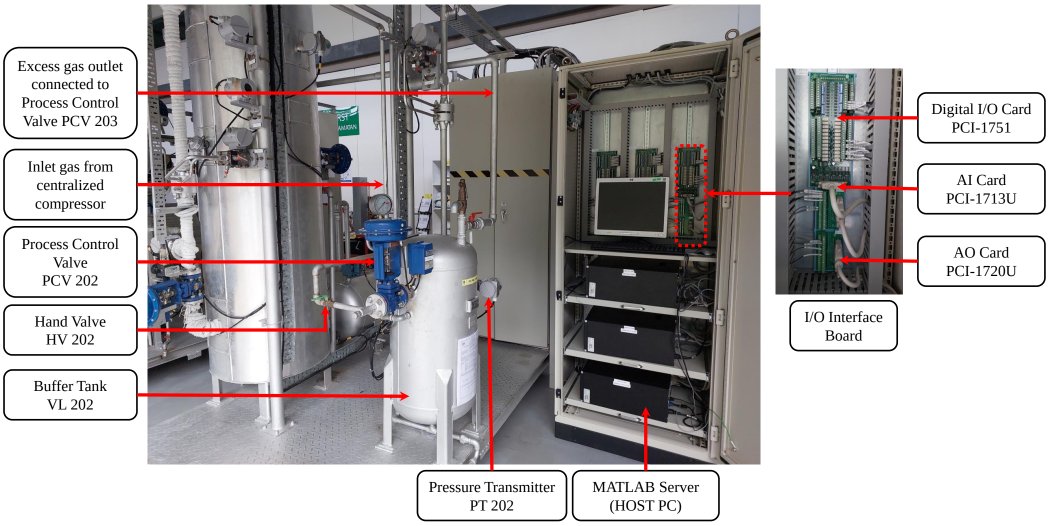

Excess gas outlet connected to Process Control Valve PCV 203
Inlet gas from centralized compressor
Process Control Valve PCV 202
Hand Valve HV 202
Buffer Tank VL 202
Pressure Transmitter PT 202
MATLAB Server (HOST PC)
Digital I/O Card PCI-1751
AI Card PCI-1713U
AO Card PCI-1720U
I/O Interface Board

Figure 3. Real-time schematic of the pressure process plant.

For safety purposes, there is an analog pressure gauge that displays pressure changes inside the tank. In case of an emergency, there is a hand-operated valve at the bottom of the buffer tank that releases compressed air from the VL 202 if the valve PCV 202 fails. The hand valve can also be used as an external disturbance injection channel during experimentation. The pressure inside the buffer tank is regulated by releasing excess air through an outlet on the top of the process tank, which is connected to another process control valve (PCV 203). This valve is kept at a 50% opening during the experiment to prevent excessive pressure build-up inside the VL 202. The host PC sends signals to the control valve actuator PCV 202 based on the set-point value.

Figure 4 displays the piping and instrumentation diagram of the pressure process plant. The plant operates in "Remote Desktop Connection" mode for safety, controlling processes from the central control room. The communication between the mainframe PC and field devices, such as the control valve actuator, flow sensors, and pressure transmitter, is established with PCI cards. These cards provide isolation protection of 2500 V DC between the PCI bus outputs. The process plant's analog input is received by a 32-channel analog input card called PCI-1713U. It has a 12-bit resolution and a sampling rate of 100 k samples. The PCI-1720U module, a crucial component of the PCI card, employs a high-quality 12-bit, 4-channel analog output port to transmit precise control signals to the host PC. Furthermore, the PCI-1751 card allows for remote control of the pressure process plant by facilitating data transmission of digital signals from the PT 202 to the host PC and vice versa, with 48 bits of parallel digital input/output.

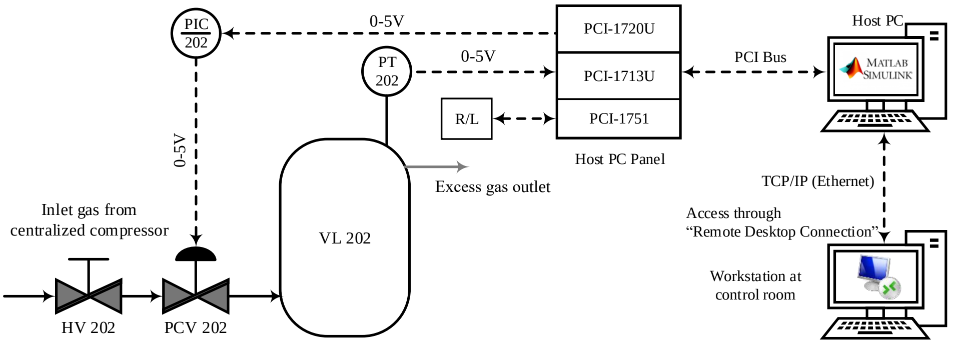


Figure 4. P&I diagram of the pressure process plant.

Obtaining the transfer function of the pressure process plant is achievable through mathematical modeling using the open-loop response for the step input signal. By applying the characteristic equation of the first-order plus dead-time system, the final transfer function of the plant is equated as given below.

$$G_p(s) = \frac{K}{1+Ts}e^{-sL_p} = \frac{0.866}{1+1.365s}e^{-s}; \tag{13}$$

where, K is the process gain, L_p represents system dead-time, and T is the system time constant.

Fractional-order predictive PI controller is a dead-time compensating control mechanism proposed by Arun et al. to overcome the limitations of conventional PI controllers [54]. The FOPPI controller's design is characterized by its simplicity and effectiveness, and it boasts the impressive capability to compensate for dead-time and is exceptionally adept at rejecting stochastic disturbances. The FOPPI controller integrates the dead-time compensating ability of the Smith predictor with the robust nature of fractional-order controllers. The FOPPI controller is particularly useful for non-linear, fast response, and sensitive applications such as pressure processes. The controller produces an adequate robust control signal unaffected by load changes or plant uncertainties. However, the controller parameters that are identified through analytical techniques may not be enough to create an effective control signal. This can cause problems with the performance of the plant. In order to overcome this issue, the proposed HHAOA and various optimization algorithms were used to obtain the optimal controller parameters. These parameters were then used to investigate the pressure process plant. The FOPPI control signal was generated using the equation provided below.

$$u(s) = K_p\left(1+\frac{1}{T_i s^{\lambda}}\right)e(s) - \frac{1}{T_i s^{\lambda}}(1-e^{-sL_p})u(s); \tag{14}$$

where, $K_p = \frac{1}{K}$. Let $u(s)$ and $e(s)$ represent the control and error signals, respectively. K_p represents the proportional gain, T_i denotes the integral time, λ represents the fractional-order integrator, and L_p represents the process dead-time. The proposed HHAOA technique was used for obtaining the FOPPI controller parameters with the goal of minimizing the integral time absolute error (ITAE) value. The ITAE value was used as the objective function for all optimization algorithms. Figure 5 shows how the FOPPI controller was tuned using the HHAOA method. The ITAE value can be calculated using the following formula:

$$\text{ITAE} = \int_0^\infty t|e(t)|dt. \tag{15}$$

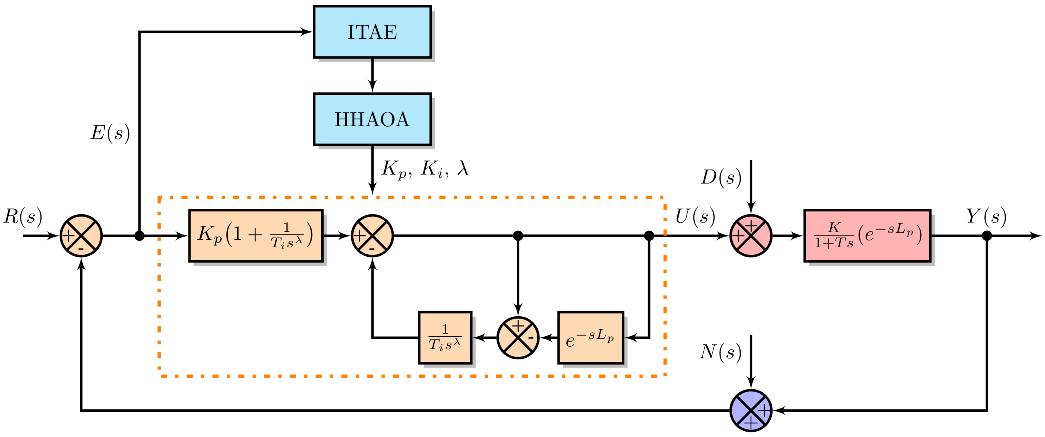

ITAE
HHAOA
$E(s)$
$K_p,\ K_i,\ \lambda$
$D(s)$
$R(s)$
$U(s)$
$Y(s)$
$K_p(1+\frac{1}{T_i s^\lambda})$
$\frac{1}{T_i s^\lambda}$
e^{-sL_p}
$\frac{K}{1+Ts}(e^{-sL_p})$
$N(s)$
+
-

Figure 5. FOPPI controller tuning using HHAOA.

4. Results and Discussion

This section presents the benchmark functions and wireless mesh network simulation analysis in the first and second subsections. Lastly, the HHAOA-optimized FOPPI controller was experimented on with the pressure process plant. During the simulation of the benchmark functions, the Hawks population was kept at 100, and ran for 300 iterations, while in the WMN the simulation ran for 500 iterations, keeping the Hawks population constant with dimensions of 1000 m by 1000 m. The experimentation and simulation were conducted in the Intel(R) Xeon PC 3.10 GHz and 16.00 GB RAM using the MATLAB/Simulink software (2021a). Additionally, extra parameters for the algorithms are listed as min = 0.2, max = 1.0, convergence constant $a = 2$, initial and escaping energies E_0 = [0 1] and E = [-1 1], α = 5, μ = 1.5, and $\epsilon = 2.2204 \times 10^{-16}$.

4.1. Performance Analysis on Benchmark Functions

A simulation analysis for 30 benchmark functions (refer to Tables 4 and 5) using AOA, MFO, SCA, GWO, WOA, HHO, and the proposed HHAOA was conducted. The selected benchmark functions were independent of each other in multiple dimensions and had various local minima, global minima (G_m), different ranges, and diverse boundary values to ensure a reliable comparative analysis. Furthermore, the algorithms were comprehensively tested using multimodal and hybrid composition functions in single and multiple dimensions.

Table 4. Optimization test functions—Unimodal and Multimodal benchmark functions.

Cat.	G_m	Func.	Description	Range
Unimodal	0	F1	$F(x) = \sum_{i=1}^{n} x_i^2$	[100,100]
	0	F2	$F(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[10,10]
	0	F3	$F(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2$	[100,100]
	0	F4	$F(x) = \max_i \lvert\ X_i\ \rvert, 1 \leq i \leq n$	[100,100]
	0	F5	$F(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1}^2 - 2x_{i+1}x_i^2 + x_i^4 \right) + \left(x_i^2 - 2x_i + 1 \right) \right]$	[30,30]
	0	F6	$F(x) = \sum_{i=1}^{n} \left(x_i^2 + 0.25 + x_i \right)$	[100,100]
	0	F7	$F(x) = \sum_{i=1}^{n} i x_i^4 + \text{random } (0,1)$	[128,128]
Multimodal	−148.9829 × n	F8	$F(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert} \right)$	[500,500]
	0	F9	$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	[5.12,5.12]
	0	F10	$F(x) = e + 20 \left(1 + \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} - \left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) \right) \right)$	[32,32]
	0	F11	$F(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	[600,600]
	0	F12	$F(x) = \frac{\pi}{n} \{ 10\sin(\pi y_1) \} + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1}) + \sum_{i=1}^{n} u(x_i, 10, 100, 4),$ where $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) \begin{cases} K(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \geq a \\ K(-x_i - a)^m & -a \leq x_i \end{cases}$	[50,50]
	0	F13	$F(x) = 0.1 \left[\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] + \left(x_n^2 - 2x_n + 1 \right) \right.$ $\left. \left[1 + \sin^2(2\pi x_n) \right] \right] + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[50,50]

Table 5. Optimization test functions—Fixed dimension multimodal benchmark functions.

Cat.	G_m	Func.	Description	Range
Fixed-dimension multimodal	1	F14	$F(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} \left(x_i - a_{ij} \right)^6} \right)^{-1}$	[−65,65]
	0.0003	F15	$F(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 \left(b_i^2 + b_i x_2 \right)}{b_i^2 + b_i x_3 + x_4} \right]^2$	[−5,5]
	−1.0316	F16	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[−5,5]
	0.398	F17	$F(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	[−5,5]
	3	F18	$F(x) = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2 \right) \right]$ $\times \left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2 \right) \right]$	[−2,2]
	−3.86	F19	$F(x) = \sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij} \left(x_j^2 - 2p_{ij} x_j + p_{ij}^2 \right) \right)$	[1,3]
	−3.32	F20	$F(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij} \left(x_j^2 - 2p_{ij} x_j + p_{ij}^2 \right)^2 \right)$	[−0,1]
	−1.01532	F21	$F(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0,10]

Table 5. *Cont.*

Cat.	G_m	Func.	Description	Range
Hybrid	−19.2085	F22	$F(x)=\left\lvert\sin(x_1)\cos(x_2)\exp\left(\left\lvert 1-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert\right)\right\rvert$	[−10,10]
	−117.49797	F23	$F(x)=\frac{1}{2}\sum_{i=1}^{d}(x_i^4-16x_i^2+5x_i)$	[−5,5]
	0	F24	$F(x)=\sin^2(\pi w_1)+\sum_{i=1}^{d-1}(w_i-1)^2\Big[1+10\sin^2(\pi w_i+1)\Big]+(w_d-1)^2\Big[1+\sin^2(2\pi w_d)\Big],$ where $, w_i=1+\frac{x_i-1}{4},$ for all $i=1,\ldots,d$	[−10,10]
	0	F25	$F(x)=100\Big(x_1^2-x_2\Big)^2+(x_1-1)^2+(x_3-1)^2+90\Big(x_3^2-x_4\Big)^2$ $+10.1\Big((x_2-1)^2+(x_4-1)^2\Big)+19.8(x_2-1)(x_4-1)$	[−10,10]
	−3.86278	F26	$F(x)=-\sum_{i=1}^{4}\alpha_i\exp\left(-\sum_{j=1}^{3}A_{ij}\Big(x_j-P_{ij}\Big)^2\right)$, where, $\alpha=(1.0,1.2,3.0,3.2)^T$ $\mathbf{A}=\begin{pmatrix}3.0 & 10 & 30\\ 0.1 & 10 & 35\\ 3.0 & 10 & 30\\ 0.1 & 10 & 35\end{pmatrix}\ \mathbf{P}=10^{-4}\begin{pmatrix}3689 & 1170 & 2673\\ 4699 & 4387 & 7470\\ 1091 & 8732 & 5547\\ 381 & 5743 & 8828\end{pmatrix}$	[0,1]
	−10.5364	F27	$F(x)=-\sum_{i=1}^{m}\left(\sum_{j=1}^{4}\Big(x_j-C_{ji}\Big)^2+\beta_i\right)^{-1},$ where, $m=10;\beta=\frac{1}{10}(1,2,2,4,4,6,3,7,5,5)^T$ $\mathbf{C}=\begin{pmatrix}4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0\\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6\\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0\\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6\end{pmatrix}$	[0,10]
	−3.32237	F28	$F(x)=-\sum_{i=1}^{4}\alpha_i\exp\left(-\sum_{j=1}^{6}A_{ij}\Big(x_j-P_{ij}\Big)^2\right),$ where, $\alpha=(1.0,1.2,3.0,3.2)^T$ $\mathbf{A}=\begin{pmatrix}10 & 3 & 17 & 3.50 & 1.7 & 8\\ 0.05 & 10 & 17 & 0.1 & 8 & 14\\ 3 & 3.5 & 1.7 & 10 & 17 & 8\\ 17 & 8 & 0.05 & 10 & 0.1 & 14\end{pmatrix}$ $\mathbf{P}=10^{-4}\begin{pmatrix}1312 & 1696 & 5569 & 124 & 8283 & 5886\\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991\\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650\\ 4047 & 8828 & 8732 & 5743 & 1091 & 381\end{pmatrix}$	[0,10]
	−9.66015	F29	$F(x)=-\sum_{i=1}^{d}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$	[0,π]
	0	F30	$F(x)=(x_1-1)^2+\sum_{i=2}^{d}(2x_i^2-x_{i-1})^2$	[−10,10]

4.1.1. Benchmark Functions

The algorithm's accuracy was tested using unimodal functions (F1 to F7). As for multimodal functions, they contain multiple local minimum points, which serve as a measure of the algorithm's exploration capacity. This capacity pertains to its ability to move seamlessly from local to global minima without getting stuck in a single position. Table 4 shows the F8 to F13 multimodal benchmark functions that have many local minima. The F14 to F21 functions also have multiple local minima with fixed dimensions. These functions help to evaluate the stability of optimization algorithms. It is necessary to mention that the first 13 benchmark functions have higher dimension values of 30, 100, 500, and 1000 to evaluate the proposed HHAOA.

The remaining functions, from F22 to F30, are hybrid functions as detailed in Table 5. While working with unimodal functions, finding the optimal solutions was comparatively effortless as they were conveniently located and easily reachable from the current location. The likelihood of being trapped in a particular search area is minimal. Finding the optimal position and movement between hybrid functions F19 and F33 is very difficult because there are many local minimum values to consider in the search space. Figure 6 displays the surface plots for all functions, providing a clear visualization of the search space needed to identify the global minimum for both single and multi-minima functions. Functions can be classified into different categories based on their surface area and shape. These categories include bowl-shaped, plate-shaped, and valley-shaped functions. There are also functions with single or multiple local minima. The latter type of function has a broad search space, wide range, numerous layers, and multi-dimensional features due to its multiple local minima values.

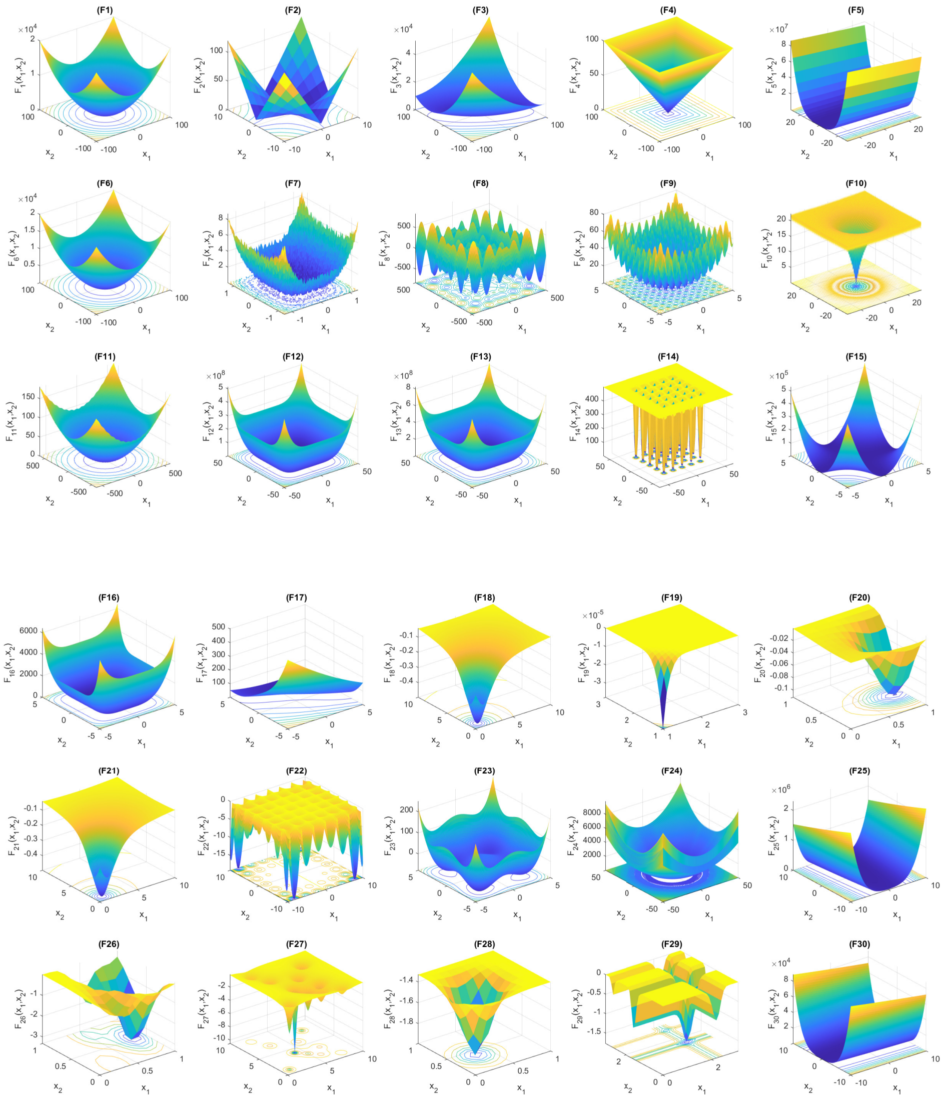


Figure 6. Benchmark functions search space plots.

4.1.2. Convergence Analysis

The benchmark procedures were evaluated through 300 iterations, with 100 for the loop condition, to determine their convergence properties. Convergence, in this context, indicates the point at which an algorithm locates the smallest possible fitness value within the specified number of maximum iterations. Figure 7 illustrates the convergence analysis plot for all the benchmark functions in the presence of all the optimization algorithms.

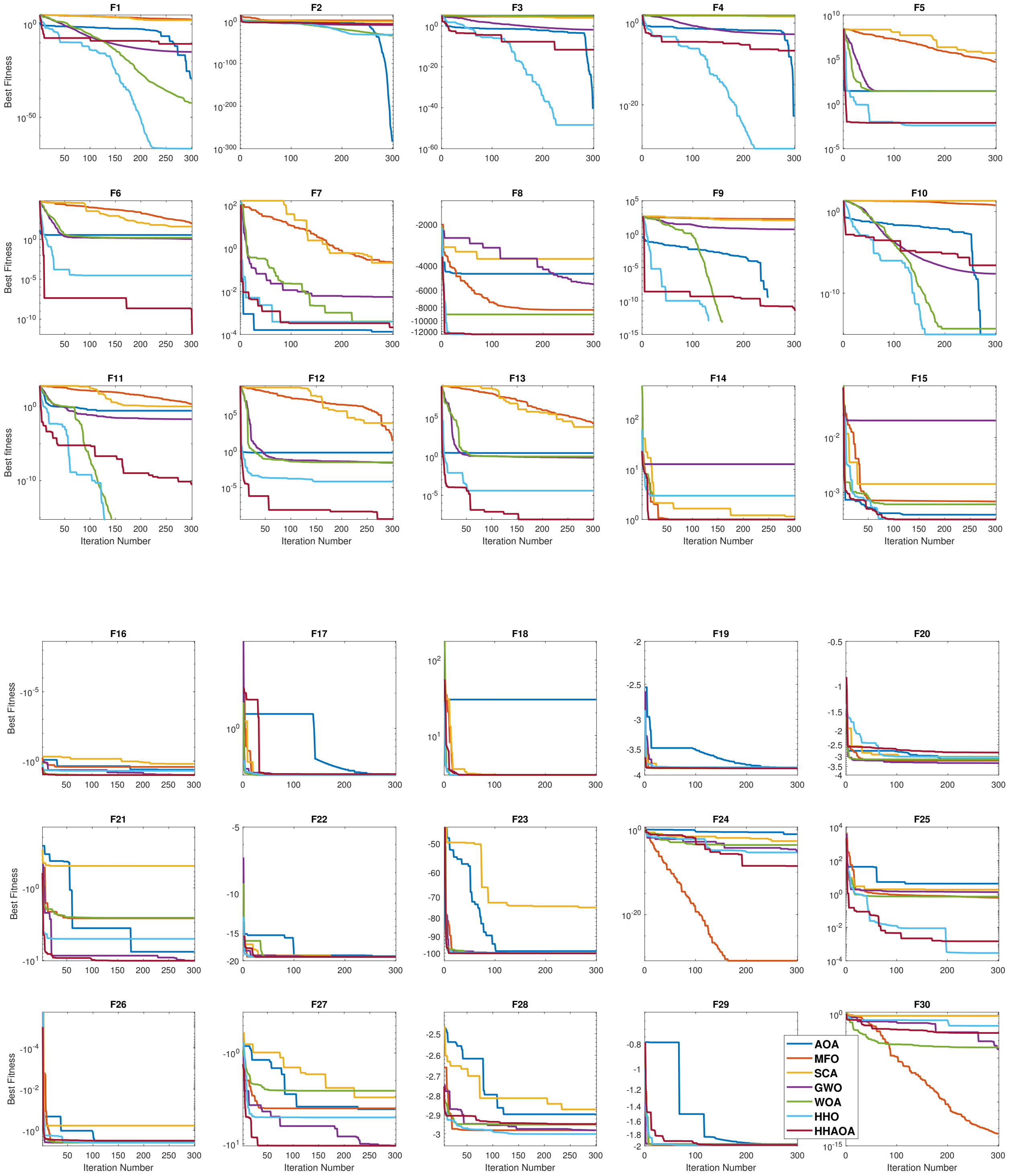


Figure 7. Convergence performance of all the benchmark functions.

In the unimodal functions (F1–F4), the conventional HHO had the fastest convergence in reaching the global minima. However, in the remaining functions, the proposed HHAOA relocated the positions faster, reaching the objective function in the first place. In this performance, the MFO had the major setback of exploring desired search regions, leading to inadequate performance and securing the last position. In the multimodal functions (F8–F13), using the hawk's population effectively, the proposed HHAOA had a better transition phase from exploration to exploitation while noticing the convergence speed, which has a more significant speed of reaching the global minima with fewer iterations. In addition, the amplitude reduction rate is directly proportional to its convergence speed in the HHAOA performance in these functions. These results, in turn, illustrate the ability of HHAOA to focus more on the desired search locations during the iterations.

It is important to note that, in F9, many algorithms were unable to reach the desired value because of the existence of multiple minimum values. (see Figure 6). Still, the HHAOA explicitly performed well and had the best and faster convergence near the global minima. SCA and MFO had the least convergence speed in these functions, followed by GWO, WOA, and AOA, respectively. Notably, the conventional HHO had performance comparable to that of the proposed HHAOA but could not outperform it because of the premature convergence towards reaching the objective function. In the fixed-dimension multimodal functions (F14–F21), the proposed HHAOA continued the first position by effectively finding the best minimum value. In these functions, AOA has the slowest convergence rate. Surprisingly, the HHO had the least performance and lost the ability to relocate the position to find the desired search regions within the desired number of iterations.

As the number of dimensions increases, the quality of outcomes and the efficiency of alternative approaches deteriorate noticeably. This indicates that HHO can effectively uphold a favorable equilibrium between exploration and exploitation behaviors in scenarios involving multiple variables. The performances in the hybrid functions (F22–F30) have a similar results trend. Notably, in F24, MFO has the best convergence performance, followed by the HHAOA. The proposed HHAOA performs best in non-zero global minima functions by converging nearer to the global minima with fewer iterations, even at less than 50 iterations in most of the functions. Lastly, the proposed HHAOA can deliver exceptional outcomes across all dimensions and consistently outperforms other methods when dealing with problems involving many variables.

4.1.3. Quantitative Analysis

In this section, the efficiency of each algorithm was evaluated by measuring how closely the statistical data match the global minimum of the benchmark function being tested. HHAOA performs better than other algorithms, as it has significant performance outcomes. However, this also means that HHAOA can identify a better optimal solution with fewer iterations and is closer to the objective function, as evidenced by its lower mean value. In contrast, a lower standard deviation (Stdv) signifies more excellent convergence stability, effectiveness, and reliability. Consequently, HHAOA is capable of avoiding local optima with great success. Table 6 show the numerical comparison for all the optimization techniques. Here, the abbreviated terms are $Func.$—Benchmark Functions, G_m—Global minima, and $Std.Dev$—Standard deviation.

Table 6. Quantitative analysis of the different benchmark functions for all the optimization algorithms.

Func.	G_m	Statistics	AOA	MFO	SCA	GWO	WOA	HHO	HHAOA
F1	1	Mean	1.20×10^{-14}	2.00×10^{3}	2.4136	2.05×10^{-33}	5.28×10^{-85}	9.02×10^{-96}	3.16×10^{-6}
		Best	1.40×10^{-270}	0.2900	0.0036	6.88×10^{-35}	1.47×10^{-96}	1.75×10^{-121}	6.84×10^{-17}
		Worst	6.00×10^{-13}	2.00×10^{4}	27.5418	1.10×10^{-32}	1.62×10^{-83}	4.51×10^{-94}	1.99×10^{-12}
		Std. Dev	8.49×10^{-14}	4.95×10^{3}	4.3477	2.80×10^{-33}	2.34×10^{-84}	2.34×10^{-84}	5.19×10^{-13}
F2	0	Mean	0.0	31.5216	0.0100	6.27×10^{-20}	4.56×10^{-53}	1.89×10^{-50}	1.68×10^{-20}
		Best	0.0	0.1102	9.06×10^{-5}	1.06×10^{-20}	1.07×10^{-61}	4.45×10^{-65}	2.13×10^{-9}
		Worst	0.0	80.0085	0.0740	2.05×10^{-19}	1.32×10^{-51}	9.29×10^{-49}	7.75×10^{-7}
		Std. Dev	0.0	21.6905	0.0140	4.19×10^{-20}	2.20×10^{-52}	1.31×10^{-49}	1.71×10^{-7}

Table 6. *Cont.*

Func.	G_m	Statistics	AOA	MFO	SCA	GWO	WOA	HHO	HHAOA
F3	0	Mean	0.0055	1.99×10^{4}	6.44×10^{3}	3.42×10^{-8}	2.75×10^{4}	5.34×10^{-81}	1.09×10^{-7}
		Best	3.63×10^{-224}	1.73×10^{3}	381.3511	3.48×10^{-12}	3.72×10^{3}	3.05×10^{-103}	4.51×10^{-13}
		Worst	0.0711	4.71×10^{4}	2.33×10^{4}	4.48×10^{-7}	4.97×10^{4}	1.85×10^{-79}	2.06×10^{-6}
		Std. Dev	0.0138	1.27×10^{4}	5.13×10^{3}	8.42×10^{-8}	1.11×10^{4}	2.75×10^{-80}	3.61×10^{-7}
F4	0	Mean	0.0221	56.8052	25.9687	2.23×10^{-8}	40.4751	3.71×10^{-51}	7.87×10^{-8}
		Best	7.89×10^{-95}	31.0704	4.4921	2.17×10^{-9}	9.20×10^{-5}	1.68×10^{-59}	0.0
		Worst	0.0514	77.9037	58.7352	1.37×10^{-7}	8.80×10^{1}	6.46×10^{-50}	3.62×10^{-7}
		Std. Dev	0.0201	10.3480	1.23×10^{1}	2.37×10^{-8}	28.5894	1.25×10^{-50}	8.83×10^{-8}
F5	0	Mean	28.2947	1.36×10^{4}	3.17×10^{4}	26.985	27.3725	0.0067	5.22×10^{-4}
		Best	26.9909	133.2200	38.2476	25.2216	26.6853	1.47×10^{-5}	3.08×10^{-12}
		Worst	28.9331	9.04×10^{4}	6.42×10^{5}	28.5425	28.7346	0.0465	0.0088
		Std. Dev	0.4033	3.12×10^{4}	1.12×10^{5}	0.7227	0.4473	0.0086	0.0018
F6	0	Mean	2.8830	797.8558	9.8129	0.4537	0.0811	4.67×10^{-5}	1.78×10^{-6}
		Best	2.2110	0.2982	4.0532	3.75×10^{-5}	0.0124	8.73×10^{-8}	1.65×10^{-10}
		Worst	3.5479	1.01×10^{4}	61.0512	1.2572	0.3644	3.50×10^{-4}	3.41×10^{-5}
		Std. Dev	0.2956	2.73×10^{3}	10.1503	0.3654	0.0844	7.22E-05	5.43×10^{-6}
F7	0	Mean	3.29×10^{-5}	3.3920	0.0682	0.0012	0.0023	1.03×10^{-4}	5.78×10^{-5}
		Best	2.34×10^{-7}	0.0592	0.0023	2.11×10^{-4}	8.20×10^{-5}	1.16×10^{-6}	1.34×10^{-7}
		Worst	1.41×10^{-4}	32.3393	0.4079	0.0031	0.0092	8.79×10^{-4}	3.78×10^{-4}
		Std. Dev	3.40×10^{-5}	7.2993	0.0741	6.13×10^{-4}	0.0021	1.49×10^{-4}	6.68×10^{-5}
F8	−418.982 × n	Mean	-5.59×10^{3}	-8.80×10^{3}	-3.84×10^{3}	-6.27×10^{3}	-1.09×10^{4}	-1.26×10^{4}	-1.25×10^{4}
		Best	-6.81×10^{3}	-1.06×10^{4}	-4.43×10^{3}	-7.77×10^{3}	-1.26×10^{4}	-1.26×10^{4}	-1.26×10^{4}
		Worst	-4.43×10^{3}	-7.08×10^{3}	-3.15×10^{3}	-4.66×10^{3}	-7.61×10^{3}	-1.26×10^{4}	-1.26×10^{4}
		Std. Dev	445.5937	836.3389	236.3415	600.7231	1.70×10^{3}	0.2895	0.1582
F9	0	Mean	0.0075	150.3165	34.7350	1.5057	0.0	0.0	0.0
		Best	0.0	50.8095	0.0348	0.0	0.0	0.0	0.0
		Worst	0.0	244.1201	124.7699	16.4049	0.0	0.0	0.0
		Std. Dev	0.0	43.4749	29.2383	3.6694	0.0	0.0	0.0
F10	0	Mean	8.88×10^{-16}	13.2023	12.8245	4.27×10^{-14}	4.30×10^{-15}	8.88×10^{-16}	7.25×10^{-8}
		Best	8.88×10^{-16}	0.4113	0.0237	3.64×10^{-14}	8.88×10^{-16}	8.88×10^{-16}	1.42×10^{-10}
		Worst	8.88×10^{-16}	19.9599	20.3176	5.42×10^{-14}	7.99×10^{-15}	8.88×10^{-16}	2.91×10^{-7}
		Std. Dev	0.0	8.1829	9.1160	3.84×10^{-15}	2.27×10^{-15}	0.0	7.46×10^{-8}
F11	0	Mean	0.1010	24.1859	0.8524	0.0028	0.0122	1.96×10^{-11}	0.0
		Best	0.0027	0.3535	0.0323	0.0	0.0	4.44×10^{-14}	0.0
		Worst	0.2423	91.0085	1.5062	0.0274	0.2028	4.39×10^{-10}	0.0
		Std. Dev	0.0722	39.8384	0.2655	0.0062	0.0427	7.02×10^{-11}	0.0
F12	0	Mean	0.4309	4.6051	2.77×10^{3}	0.0298	0.0168	2.52×10^{-6}	2.24×10^{-8}
		Best	0.3170	0.3583	0.559	0.0033	0.0012	5.46×10^{-9}	2.08×10^{-13}
		Worst	0.5359	17.6617	1.02×10^{5}	0.0731	0.1415	1.55×10^{-5}	7.07×10^{-7}
		Std. Dev	0.0482	3.5177	1.49×10^{4}	0.0128	0.0274	3.41×10^{-6}	1.01×10^{-7}
F13	0	Mean	2.8075	9.5337	1.09×10^{4}	0.3805	0.2106	2.72×10^{-5}	1.85×10^{-6}
		Best	2.4792	0.6747	2.3552	9.01×10^{-5}	0.0188	7.41×10^{-7}	4.10×10^{-13}
		Worst	2.9930	28.3836	1.48×10^{5}	0.7516	0.8096	1.64×10^{-4}	3.97×10^{-5}
		Std. Dev	0.1253	6.0884	3.28×10^{4}	0.1971	0.1581	3.85×10^{-5}	6.47×10^{-6}
F14	1	Mean	8.9976	1.5926	1.514	3.5875	1.7886	1.1578	0.998
		Best	0.9980	0.9980	0.998	0.998	0.998	0.998	0.998
		Worst	12.6705	4.9500	2.9821	12.6705	10.7632	1.992	0.998
		Std. Dev	4.2660	1.0769	0.8791	3.6212	1.5727	0.3678	0.0
F15	0.0003	Mean	0.0115	9.08×10^{-4}	9.91×10^{-4}	0.0035	5.47×10^{-4}	3.29×10^{-4}	3.45×10^{-4}
		Best	3.59×10^{-4}	3.15×10^{-4}	3.33×10^{-4}	3.07×10^{-4}	3.08×10^{-4}	3.08×10^{-4}	3.08×10^{-4}
		Worst	0.1171	0.0015	0.0016	0.0204	0.0014	4.01×10^{-4}	4.35×10^{-4}
		Std. Dev	0.0244	2.89×10^{-4}	3.69×10^{-4}	0.0074	2.55×10^{-4}	2.24×10^{-5}	3.17×10^{-5}
F16	−1.0316	Mean	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
		Best	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
		Worst	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
		Std. Dev	8.21×10^{-8}	0.0	0.0	0.0	0.0	0.0	0.0

Table 6. *Cont.*

Func.	G_m	Statistics	AOA	MFO	SCA	GWO	WOA	HHO	HHAOA
F17	0.398	Mean	0.4080	0.3979	0.3993	0.3979	0.3979	0.3979	0.398
		Best	0.3983	0.3979	0.3979	0.3979	0.3979	0.3979	0.3980
		Worst	0.4434	0.3979	0.4062	0.3981	0.3979	0.3979	0.398
		Std. Dev	0.0089	0.0	0.0016	2.86×10^{-5}	0.0	0.0	0.0
F18	3	Mean	8.4006	3.0	3.0	3.0	3.0	3.0	3.0
		Best	3.0	3.0	3.0	3.0	3.0	3.0	3.0
		Worst	30.0	3.0	3.0	3.0001	3.0001	3.0	3.0
		Std. Dev	10.9094	0.0	0.0	1.43×10^{-5}	2.03×10^{-5}	0.0	0.0
F19	−3.86	Mean	−3.8536	−3.8628	−3.8548	−3.8615	−3.8587	−3.862	−3.86
		Best	−3.8626	−3.8628	−3.8613	−3.8628	−3.8628	−3.8628	−3.86
		Worst	−3.8405	−3.8628	−3.8512	−3.8549	−3.8277	−3.8552	−3.86
		Std. Dev	0.0037	0.0	0.0024	0.0026	0.0065	0.0014	0.0
F20	−3.32	Mean	−3.0815	−3.2168	−3.006	−3.2532	−3.2412	−3.1334	−3.32
		Best	−3.1835	−3.3220	−3.1714	−3.322	−3.3219	−3.311	−3.32
		Worst	−2.8472	−3.1376	−2.5881	−3.0604	−3.0807	−2.8095	−3.32
		Std. Dev	0.0675	0.0448	0.1305	0.077	0.0915	0.0976	0.0
F21	−10.153	Mean	1.20×10^{-14}	2.00×10^{3}	2.4136	2.05×10^{-33}	5.28×10^{-85}	9.02×10^{-96}	3.16×10^{-16}
		Best	1.40×10^{-270}	0.2900	0.0036	6.88×10^{-35}	1.47×10^{-96}	1.75×10^{-121}	6.84×10^{-17}
		Worst	6.00×10^{-13}	2.00×10^{4}	27.5418	1.10×10^{-32}	1.62×10^{-83}	4.51×10^{-94}	1.99×10^{-12}
		Std. Dev	8.49×10^{-14}	4.95×10^{3}	4.3477	2.80×10^{-33}	2.34×10^{-84}	2.34×10^{-84}	5.19×10^{-13}
F22	−19.2085	Mean	0.0	31.5216	0.0100	6.27×10^{-20}	4.56×10^{-53}	1.89×10^{-50}	1.68×10^{-20}
		Best	0.0	0.1102	9.06×10^{-5}	1.06×10^{-20}	1.07×10^{-61}	4.45×10^{-65}	2.13×10^{-9}
		Worst	0.0	80.0085	0.0740	2.05×10^{-19}	1.32×10^{-51}	9.29×10^{-49}	7.75×10^{-7}
		Std. Dev	0.0	21.6905	0.0140	4.19×10^{-20}	2.20×10^{-52}	1.31×10^{-49}	1.71×10^{-7}
F23	−117.497	Mean	0.0055	1.99×10^{4}	6.44×10^{3}	3.42×10^{-8}	2.75×10^{4}	5.34×10^{-81}	1.09×10^{-7}
		Best	3.63×10^{-224}	1.73×10^{3}	381.3511	3.48×10^{-12}	3.72×10^{3}	3.05×10^{-103}	4.51×10^{-13}
		Worst	0.0711	4.71×10^{4}	2.33×10^{4}	4.48×10^{-7}	4.97×10^{4}	1.85×10^{-79}	2.06×10^{-6}
		Std. Dev	0.0138	1.27×10^{4}	5.13×10^{3}	8.42×10^{-8}	1.11×10^{4}	2.75×10^{-80}	3.61×10^{-7}
F24	0	Mean	0.0221	56.8052	25.9687	2.23×10^{-8}	40.4751	3.71×10^{-51}	7.87×10^{-8}
		Best	7.89×10^{-95}	31.0704	4.4921	2.17×10^{-9}	9.20×10^{-5}	1.68×10^{-59}	0.0
		Worst	0.0514	77.9037	58.7352	1.37×10^{-7}	8.80×10^{1}	6.46×10^{-50}	3.62×10^{-7}
		Std. Dev	0.0201	10.3480	1.23×10^{1}	2.37×10^{-8}	28.5894	1.25×10^{-50}	8.83×10^{-8}
F25	0	Mean	28.2947	1.36×10^{4}	3.17×10^{4}	26.985	27.3725	0.0067	5.22×10^{-4}
		Best	26.9909	133.2200	38.2476	25.2216	26.6853	1.47×10^{-5}	3.08×10^{-12}
		Worst	28.9331	9.04×10^{4}	6.42×10^{5}	28.5425	28.7346	0.0465	0.0088
		Std. Dev	0.4033	3.12×10^{4}	1.12×10^{5}	0.7227	0.4473	0.0086	0.0018
F26	−3.8628	Mean	2.8830	797.8558	9.8129	0.4537	0.0811	4.67×10^{-5}	1.78×10^{-6}
		Best	2.2110	0.2982	4.0532	3.75×10^{-5}	0.0124	8.73×10^{-8}	1.65×10^{-10}
		Worst	3.5479	1.01×10^{4}	61.0512	1.2572	0.3644	3.50×10^{-4}	3.41×10^{-5}
		Std. Dev	0.2956	2.73×10^{3}	10.1503	0.3654	0.0844	7.22×10^{-5}	5.43×10^{-6}
F27	−10.5364	Mean	3.29×10^{-5}	3.3920	0.0682	0.0012	0.0023	1.03×10^{-4}	5.78×10^{-5}
		Best	2.34×10^{-7}	0.0592	0.0023	2.11×10^{-4}	8.20×10^{-5}	1.16×10^{-6}	1.34×10^{-7}
		Worst	1.41×10^{-4}	32.3393	0.4079	0.0031	0.0092	8.79×10^{-4}	3.78×10^{-4}
		Std. Dev	3.40×10^{-5}	7.2993	0.0741	6.13×10^{-4}	0.0021	1.49×10^{-4}	6.68×10^{-5}
F28	−3.3224	Mean	-5.59×10^{3}	-8.80×10^{3}	-3.84×10^{3}	-6.27×10^{3}	-1.09×10^{4}	-1.26×10^{4}	-1.25×10^{2}
		Best	-6.81×10^{3}	-1.06×10^{4}	-4.43×10^{3}	-7.77×10^{3}	-1.26×10^{4}	-1.26×10^{4}	-1.26×10^{4}
		Worst	-4.43×10^{3}	-7.08×10^{3}	-3.15×10^{3}	-4.66×10^{3}	-7.61×10^{3}	-1.26×10^{4}	-1.26×10^{4}
		Std. Dev	445.5937	836.3389	236.3415	600.7231	1.70×10^{3}	0.2895	0.1582
F29	−9.6602	Mean	0.0075	150.3165	34.7350	1.5057	0.0	0.0	0.0
		Best	0.0	50.8095	0.0348	0.0	0.0	0.0	0.0
		Worst	0.0	244.1201	124.7699	16.4049	0.0	0.0	0.0
		Std. Dev	0.0	43.4749	29.2383	3.6694	0.0	0.0	0.0
F30	0	Mean	8.88×10^{-16}	13.2023	12.8245	4.27×10^{-14}	4.30×10^{-15}	8.88×10^{-16}	7.25×10^{-8}
		Best	8.88×10^{-16}	0.4113	0.0237	3.64×10^{-14}	8.88×10^{-16}	8.88×10^{-16}	1.42×10^{-10}
		Worst	8.88×10^{-16}	19.9599	20.3176	5.42×10^{-14}	7.99×10^{-15}	8.88×10^{-16}	2.91×10^{-7}
		Std. Dev	0.0	8.1829	9.1160	3.84×10^{-15}	2.27×10^{-15}	0.0	7.46×10^{-8}

The unimodal function results show that the conventional HHO had a lead in reaching the global minima value more than the other algorithms. Surprisingly, GWO and WOA have an average performance compared to the newly developed AOA and SCA algorithms. The proposed HHAOA has notable performances in F4–F7 due to the arithmetic operators' and hawks' ability to narrow down the desired global minima and avoid further searching for the exploration locations. Meanwhile, the MFO performs poorly in these functions, which shows its inability to reach even for the singular objective functions.

The HHAOA method is able to reach the global minimum in both multimodal and fixed-dimension functions better than other techniques. HHO reached the desired value in most functions, but HHAOA converged at the exact minimum values in many cases. It's important to note that almost all algorithms converged at the desired global minimum values in F9 and F16. The MFO competed with the proposed HHAOA in mean and best value in the fixed dimension functions. At the same time, the well-performed GWO and WOA in the unimodal function have a significant setback in this multi-objective function. They struggled to reach the best value in most of the and had an almost massive difference in the final convergence values than the desired global minima.

Lastly, the same results' trend has been repeated in the hybrid functions. Notably, the HHO and SCA had poor performance and these algorithms quickly progressed in the first few iterations, but then the rate of improvement slowed down. The GWO and MFO had a closer convergence value in most functions, showing the same approach to finding the best optimum values (see Figure 7), whereas the proposed HHAOA had a stable and smooth value closer to the global minima, it suggests that the algorithm converges stably. However, in F23, the HHAOA has a lower standard deviation, indicating that the algorithm consistently finds a better solution for a stable convergence. Overall, these statistical measurements prove that the convergence behavior of the HHAOA helps to identify effective search locations for improving the exploration and exploitation phases of the algorithm.

The Friedman ranking test, which compares algorithms based on their best mean values in numerical analysis, is displayed in Table 7. In this comparative analysis, the Friedman ranking test is used to evaluate the performance of various metaheuristic approaches. The algorithms were ranked according to their mean value for the corresponding functions and the resulting list clearly compares each approach's effectiveness. The top-performing algorithm is awarded a rank of 1 as it is the closest to the global minimum. On the other hand, the algorithm with the lowest mean value is assigned a rank of 5, indicating its significant deviation from the desired global minimum value. In this analysis, the proposed HHAOA has the smallest average value of 1.967, securing the first rank, which signifies the technique's effectiveness in identifying the best optimum value with fewer iterations in most benchmark functions. In addition, the HHAOA has a massive 181.342% faster and best mean value compared with the SCA, which has a final average value of 5.534 with the last rank. The HHAOA has 35.587%, 64.412%, and 79.664% of increased performance than the HHO, GWO, and respectively. Lastly, the MFO has an average of 4.2 and secures fifth rank, followed by AOA with the value of 5.067, securing the second-to-last rank.

Table 7. Different benchmark functions of Friedman ranking test for the comparing algorithms.

Function	AOA	MFO	SCA	GWO	WOA	HHO	HHAOA
1	5	7	6	3	2	1	4
2	1	7	6	5	2	3	4
3	4	6	5	2	7	1	3
4	4	7	5	2	6	1	3
5	5	6	7	3	4	2	1
6	5	7	6	4	3	2	1
7	1	7	6	4	5	3	2
8	3	5	2	4	6	7	1
9	4	7	6	5	3	2	1
10	1	7	6	4	3	1	5
11	5	7	6	3	4	2	1
12	5	6	7	4	3	2	1
13	5	6	7	4	3	2	1
14	7	4	3	6	5	2	1
15	7	4	5	6	3	1	2
16	1	1	1	1	1	1	1
17	7	1	6	1	1	1	5
18	7	1	1	1	1	1	1
19	7	1	6	3	5	2	4
20	6	4	7	2	3	5	1
21	6	4	7	2	3	5	1
22	7	1	6	1	1	1	1
23	6	1	5	7	1	3	4
24	7	1	6	3	5	4	2
25	3	4	6	5	7	2	1
26	7	2	6	4	5	3	1
27	7	4	6	1	3	5	2
28	6	4	7	2	3	5	1
29	7	3	6	2	4	5	1
30	6	1	7	3	4	5	2
Average	**5.067**	**4.2**	**5.534**	**3.234**	**3.534**	**2.667**	**1.967**
Final Rank	**6**	**5**	**7**	**3**	**4**	**2**	**1**

4.2. Simulation Analysis of Industrial WMNs

Understanding the convergence behavior of optimization algorithms is crucial for obtaining optimal solutions. In the performance evaluation of the proposed HHAOA algorithm against other algorithms, the convergence behavior was analyzed using 500 iterations with 100 runs for 100 search agent sets, shown in Figure 8. Table 8 shows the statistical analysis of the convergence. By observing the convergence ability of these algorithms, their performance can be determined clearly. The objective function was utilized to find the minimum fitness value, which demonstrates improved connectivity and coverage.

Table 8. Convergence analysis for the WMN for all the algorithms.

Optimization	Mean	Best	Worst	Std. Dev
AOA	0.531	0.485	0.673	0.0858
MFO	0.488	0.475	0.735	0.0841
SCA	0.472	0.46	0.524	0.0675
GWO	0.463	0.450	0.720	0.0587
WOA	0.456	0.435	0.634	0.0046
HHO	0.469	0.455	0.755	0.0674
HHAOA	0.449	0.435	0.613	0.0025

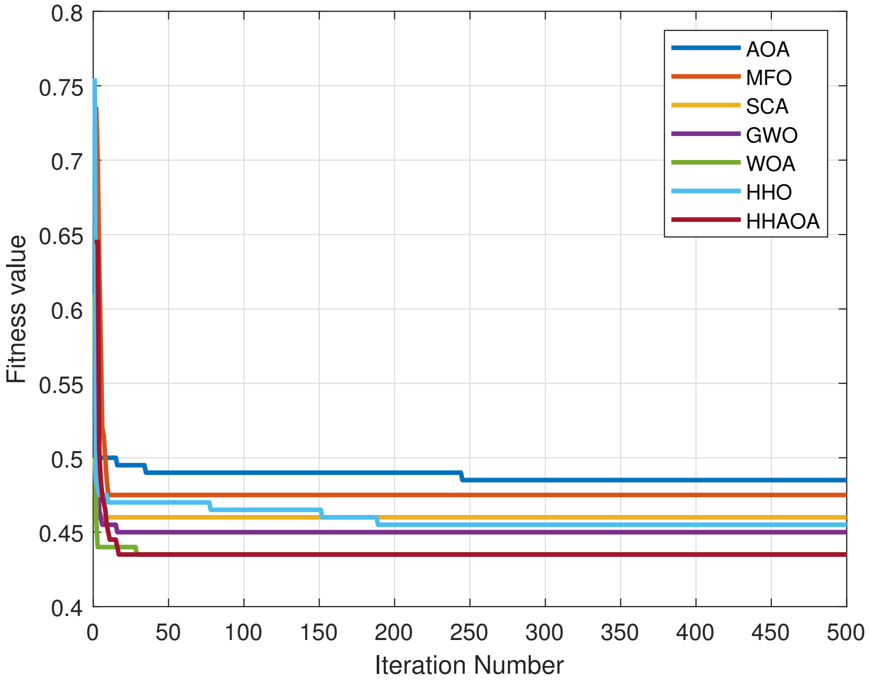
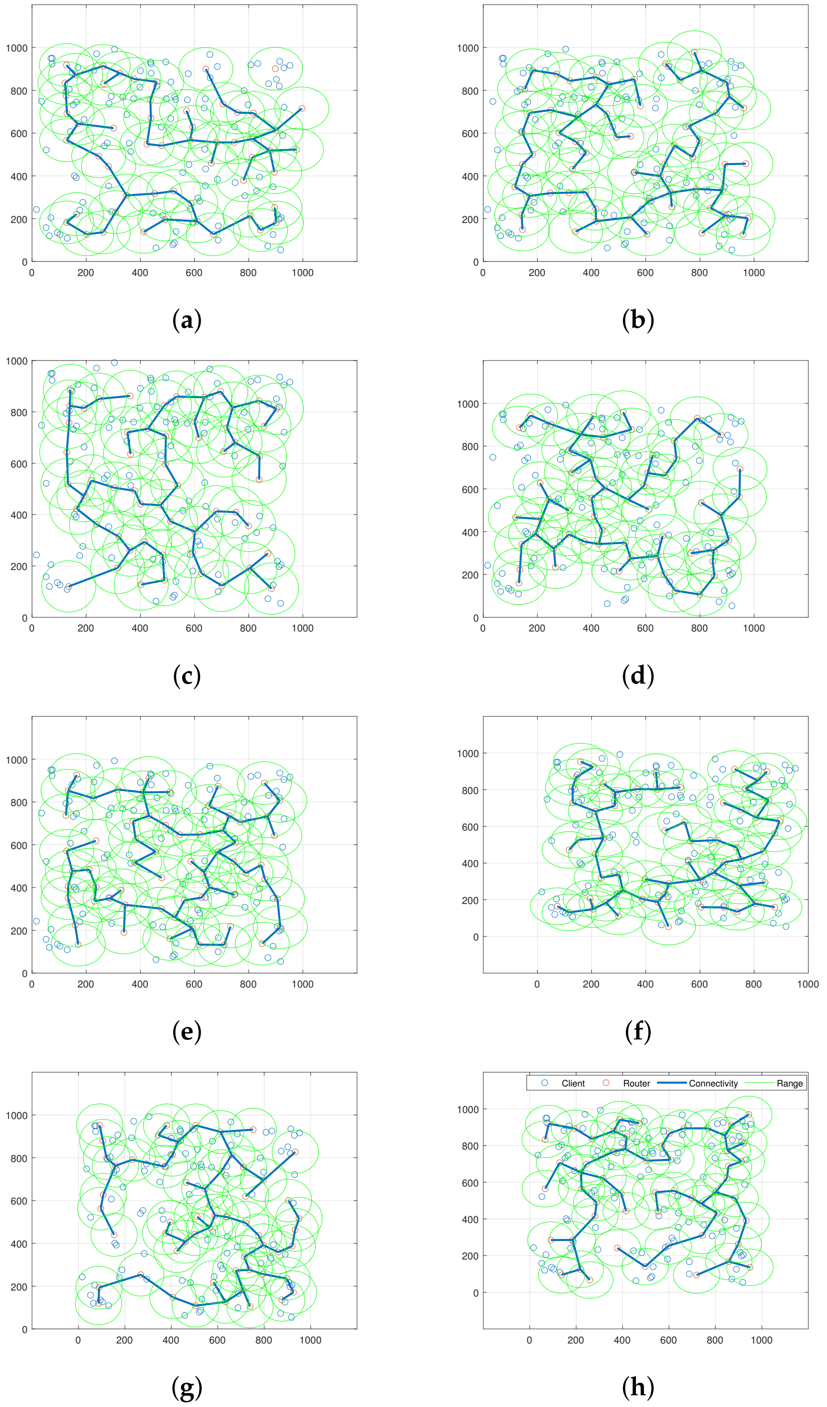


Figure 8. WMN convergence for different algorithms.

In the numerical analysis, the HHAOA method yielded the best outcomes with the least mean value of 0.499. Comparatively, the AOA came in second place with a mean value of 0.531. While the HHAOA method is 6.412% faster than AOA, it is worth noting that the WOA method is lagging with a minimal difference of 0.007. This resulted in WOA reaching a third place with a mean value of 0.456, which is 8.617% higher than the HHAOA method. The GWO, HHO, SCA, and MFO methods were followed in the remaining ranking order, which makes it clear that the proposed HHAOA method produced better results in the numerical analysis.

Figure 9 shows the network connectivity and coverage region for the WMN for all the algorithms and the initial WMN topology. Figure 9a illustrates that the initial iteration resulted in insufficient mesh client coverage and high network congestion due to multiple overlapping mesh routers, which increased deployment costs. Even with redundant mesh routers, their connectivity did not reach the maximum capacity of 100%. Figure 9b–h shows the optimized WMN connectivity and coverage of the AOA, MFO, SCA, GWO, WOA, HHO, and the proposed HHAOA, respectively. In these, SCA has the maximum number of disconnected clients and more number of overlapped router placements. It has a 150% less efficient connectivity than the HHO, which secured second place, with four unconnected clients in its optimized WMN. Additionally, the AOA has eight disconnected clients, followed by MFO, WOA, and GWO, with seven, five, and four disconnected clients in the WMN network. However, Figure 9h shows that the proposed HHAOA has dramatically improved the client coverage and optimally deployed the mesh routers to achieve full network connectivity with the least number of routers and had only one disconnected client. Likewise, the proposed HHAOA significantly reduced network congestion by reducing the number of mesh routers by 31.7% while maintaining high coverage and connectivity. Furthermore, other comparing algorithms produced network topologies where mesh routers overlapped, resulting in more significant interference. In contrast, the topology formed by the HHAOA approach was more widely dispersed, resulting in improved client coverage.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 9. WMN network connectivity and its coverage area for various algorithms. (**a**) Initial Network; (**b**) AOA Optimized WMN; (**c**) MFO Optimized WMN; (**d**) SCA Optimized WMN; (**e**) GWO Optimized WMN; (**f**) WOA Optimized WMN; (**g**) HHO Optimized WMN; (**h**) HHAOA Optimized WMN.

4.3. Performance Evaluation on Pressure Process Control

In the numerical analysis, the comparison was carried out for the process rise time (t_r), settling time before (t_{s1}) disturbance, overshoot ($\%OS$), and settling time after (t_{s2}) disturbance injection. In order to evaluate the ability of the FOPPI to decrease the stochastic disturbance and to track the set-point effectively, an external disturbance of 35% was injected at 100 s in the process feedback loop. Table 9 shows the quantitative analysis of the optimized FOPPI controller using different optimization techniques, and the comparative analysis is shown in Figure 10. Here, controller parameters K_p, T_i and λ were obtained based on the ITAE value and integral time was obtained by $T_i = \frac{K_p}{K_i}$, where K_i is the integral gain.

Table 9. Experimental quantitative analysis of various optimized FOPPI controller on the pressure process.

Optimization Type	K_p	K_i	λ	t_r	t_{s1}	t_{s1}	%OS
AOA	1.207	0.836	0.97	0.8404	59.6059	142.2151	2.8723
MFO	2.713	1.183	0.98	0.7552	57.5289	140.1284	18.9260
SCA	1.881	1.062	0.96	1.0371	66.9394	151.0140	23.3440
GWO	1.672	0.714	0.96	5.1078	52.9052	135.1132	2.5131
WOA	2.015	0.615	0.99	0.9381	54.2561	137.2006	2.6118
HHO	2.427	1.031	0.99	0.8411	49.2073	130.0259	4.8653
HHAOA	2.359	0.901	0.98	1.4504	37.2265	119.2158	4.7061

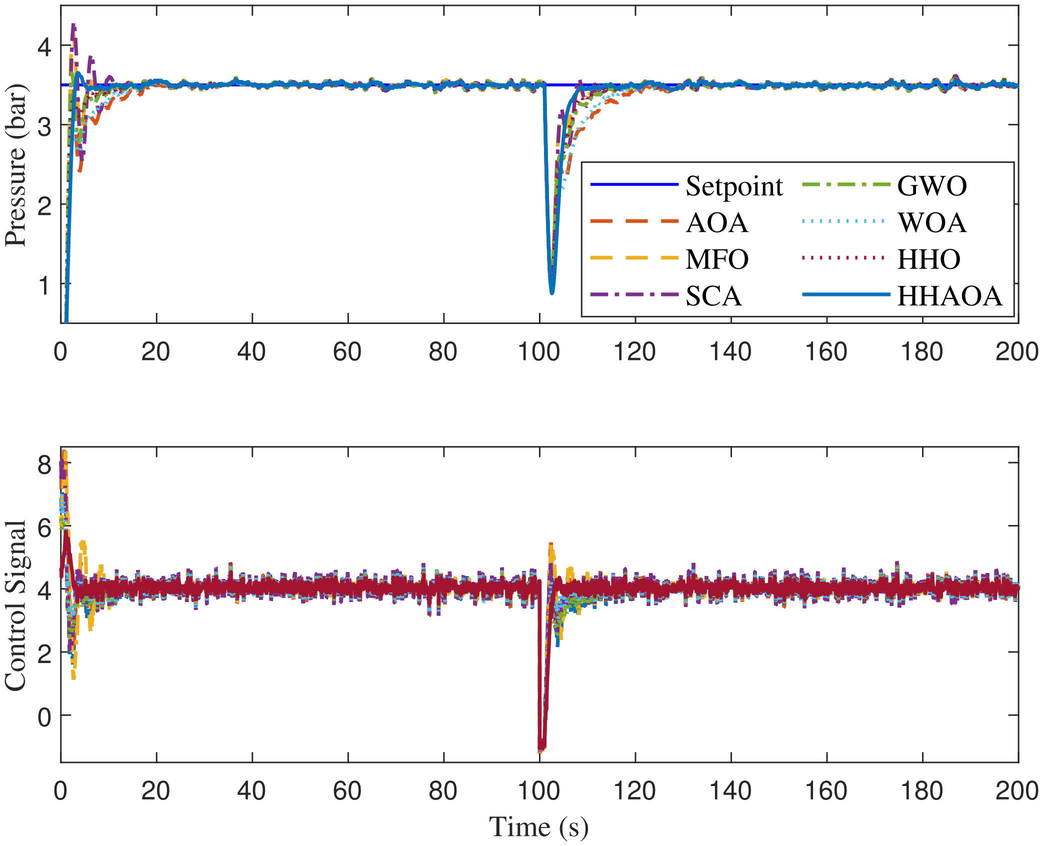


Figure 10. Set-point tracking and disturbance rejection analysis for optimal FOPPI controller.

The numerical results from the table show that the MFO has the fastest rise time of 0.7552 s, followed by the AOA, HHO, WOA, and SCA with values of 0.8404, 0.8411, and 0.9381 s, respectively. Here the GWO has the slowest rise time of 5.1078 s which is 576.351% slower performance than the fastest MFO. The proposed HHAOA optimized FOPPI had the second-last rise time of 1.4504 s which is almost two times slower than the MFO.

During the settling time before disturbance (t_{s1}), even with the slowest rise time, the HHAOA-optimized FOPPI managed to settle faster at 37.2265 s which is 29.7129 s faster than the slowest settled SCA with the settling at 66.9394 s. The result shows that the proposed HHAOA has 79.816% faster performance, which is twice as fast as SCA. The HHO comes second fastest, settling at 49.2073 s, followed by GWO, WOA, MFO, and AOA with the respective values of 52.9052, 54.2561, 57.5289, and 59.6059 s. While observing the settling time t_{s2} after disturbance injection performance, once again, the

HHAOA settled faster at 119.2158, which is 34.7982 s ahead of SCA (151.0140s). The same settling time trend is also observed in this, with the HHO settling at 130.0259 s followed by GWO settling at 135.1132 s. The second last rank is for AOA settling at 142.2151 s. In the peak overshoot performance, the HHAOA has 4.7601%, corresponding to a 396.037% reduction of the value, while comparing the SCA at 23.3340%. Amazingly, the GWO has the least peak overshoot value of 2.5131%, followed by WOA, AOA, HHO, and HHAOA, respectively. In this numerical performance analysis, in most of the cases, the Friedman ranking order is observed.

Based on the comparison results, it is evident that the FOPPI controller optimized by HHAOA is more efficient in producing a stable control signal and rejecting disturbances. However, it is worth noting that the HHAOA may have a slower rise time when tracking the initial set-point, which is noticeable. In the same period, the HHAOA has the fastest settling ahead of others with minimal peak overshoot, are shown in Figure 11, section A. Later, in the external disturbance injection at 100 s, the HHAOA optimized FOPPI had the fastest disturbance recovery and better set-point tracking ability (see, Figure 11, section B). The control actions of the FOPPI for different algorithms are shown in Figure 11, sections C and D. Here, the HHAOA has the robust and effectively generated control signal at 4.3 itself, while others are starting at above 6.0. This is also true in the case of the after-disturbance injection scenario. It is clear from the results that finding the optimized controller parameters for the real-time process plant is essential in order to improve its performance.

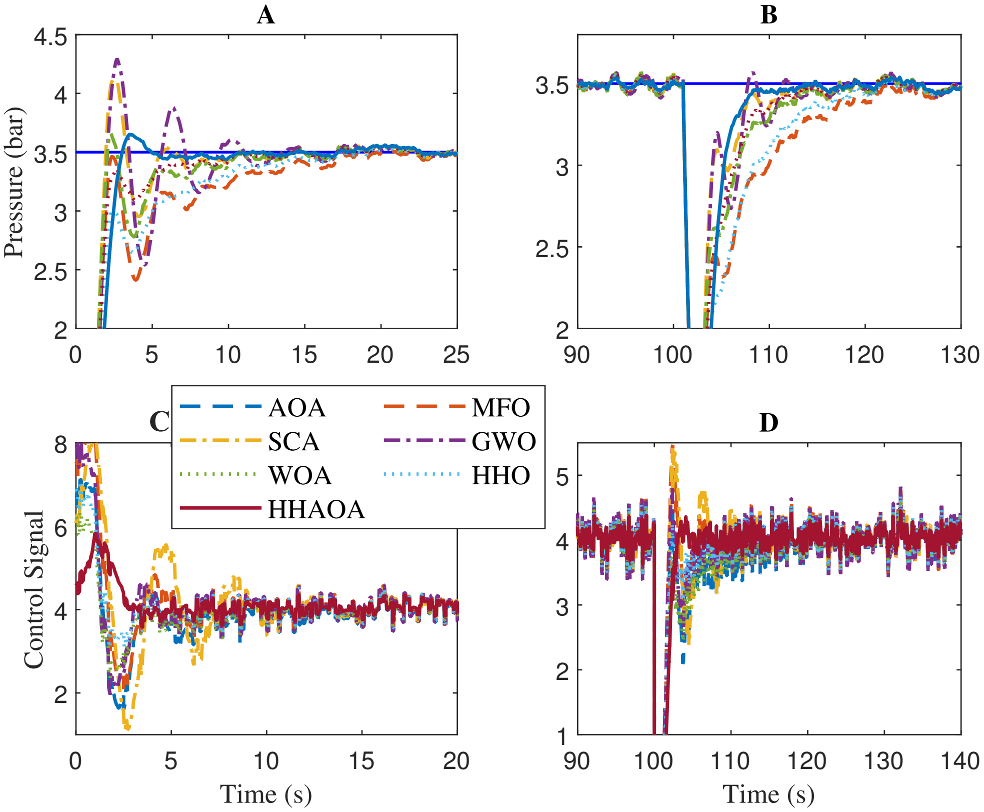


Figure 11. Zoomed view of Figure 10 (**A**) Initial set-point tracking (**B**) Disturbance rejection performance (**C**) Control signal during initial set-point (**D**) Control signal during disturbance rejection.

5. Summary and Conclusions

This section presents the overall summary, which includes the proposed techniques and their advancements given in the first part. Later, the future scope of the current research and the concluding remarks are given.

5.1. Summary

This study analyzed different optimization techniques: AOA, MFO, SCA, GWO, WOA, HHO, and the proposed HHAOA. The tests were conducted on various benchmark functions and optimal router placement and connectivity for WMN benchmark functions. Additionally, the optimized FOPPI controller for a real-time pressure process plant was examined. The HHAOA technique outperformed all other algorithms in every comparison analysis, including the rigorous Friedman ranking test. This statistical evidence unequivocally proves the superiority of the HHAOA method over existing techniques. Moreover, it is crucial to highlight the notable contributions that further emphasize the effectiveness of the proposed technique.

1. Utilizing a range of multi-hopping methodologies, the proposed HHAOA can effectively detect the global minima with fewer attempts throughout most benchmark functions, resulting in a more efficient and accurate optimization process. These can be evidenced by its consistently higher mean, best, and standard deviation scores in benchmark functions testing.
2. Through various multi-hopping techniques, the proposed HHAOA is able to identify the global minima in significantly fewer attempts in most of the benchmark functions.
3. In the WMN, the HHAOA demonstrated a highly competent desired objective searching technique, producing the most optimal path for routing network traffic. The algorithm significantly reduces congestion by minimizing the data transmitted across the network, resulting in improved network performance.
4. HHAOA carefully selects the appropriate access points to connect to the network, ensuring that clients are always connected to nearby routers. This makes it an essential optimization technique for successfully deploying wireless mesh networks for the best performance.
5. Notable improvements in producing the robust and smoother control signal of the pressure process were achieved by optimizing the FOPPI controller parameters using HHAOA. These include smaller peak overshoot, dynamic set-point tracking, and practical disturbance rejection ability.

5.2. Conclusions

This research article introduced a novel optimization technique called hybrid HHAOA, which combines two existing algorithms, HHO and AOA, to achieve a better performance. In order to assess the efficacy of the HHAOA algorithm, tests were conducted on a total of 33 optimization benchmark functions. The analysis involved a comparison of performance results, which were based on various measures, including mean, global best, worst, and standard deviation. The convergence performance of the HHAOA algorithm is faster at achieving global minima with fewer iterations than other algorithms. The comparison results were evaluated using Friedman ranking, which showed that the proposed HHAOA algorithm significantly outperforms various algorithms with a 181.342% increased performance ranking in terms of the final mean value. In addition, the best connectivity, network overlapping minimization, and optimal router placement for WMN using the proposed HHAOA were simulated for 500 iterations with 100 search agents. The HHAOA produced the most satisfactory desired performance by creating the best client router connectivity with only one client disconnected in the network. Additionally, the network overlapping was significantly reduced, with a 31.7% reduction in the WMN routers substantially minimizing the operational cost. Experimentation was conducted on a real-time pressure process to further demonstrate the HHAOA algorithm's effectiveness. The findings showed that the proposed algorithm performed best in the dynamic processes. Furthermore, using HHAOA-optimized FOPPI resulted in a more reliable, smooth, and robust control signal, leading to quicker settling and reduced peak overshoot which, in turn, significantly minimizes the wear and tear on the control valve. As part of future research, newer evolutionary algorithms with different mathematical operators will be investigated to enhance the algorithm's performance. Additionally, attempts will be made to hybridize the HHAOA algorithm with other metaheuristic optimization algorithms to widen its applicability to more complex, real-time industrial and engineering problems.

Author Contributions: Conceptualization, R.I. and M.O.; methodology, P.A.M.D.; software, P.A.M.D.; validation, P.A.M.D., K.B. and H.A.; formal analysis, P.A.M.D.; investigation, P.A.M.D. and K.B.; resources, R.I. and M.O.; data curation, P.A.M.D. and H.A.; writing—original draft preparation, P.A.M.D. and K.B.; writing—review and editing, P.A.M.D. and K.B.; visualization, P.A.M.D. and H.A.; supervision, R.I. and M.O.; project administration, R.I. and M.O.; funding acquisition, R.I. and M.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Yayasan Universiti Teknologi PETRONAS-Prototype Research Grant 015PBC-001.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors would like to acknowledge the support from Universiti Teknologi PETRONAS for providing the research facilities.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ALO	Ant Lion Optimizer
AOA	Arithmetic Optimization Algorithm
ABC	Artificial Bee Colony
BMVO	Binary Multi-Verse Optimizer
BSSA	Binary Salp Swarm Algorithm
BLPSO	Biogeography-based Learning Particle Swarm Optimization
BOA	Butterfly Optimization Algorithm
DESMAOA	Deep Ensemble of Slime Mold Arithmetic Optimization Algorithm
DE	Differential Evolution
FOPPI	Fractional-order Predictive PI
GA	Genetic Algorithm
GOA	Grasshopper Optimisation Algorithm
GSA	Gravitational Search Algorithm
GWO	Grey Wolf Optimization
HART	Highway Addressable Remote Transducer
HHO	Harris Hawk Optimization
HHSS	Harris Hawk and Salp Swarm
HH-AOA	Harris Hawk-Arithmetic Optimization Algorithm
HMOHHO	Hybrid Mutation Operator Harris Hawks Optimization
IIAOA	mproved Arithmetic Optimization Algorithm
IANN	Improved Artificial Neural Network
ISA	International Society of Automation
JOS	Joint Opposite Selection
MPC	Model Predictive Controller
MHHO	Modified Harris Hawk Optimizer
MFO	Moth Flame Optimization
MOHHO	Multi-Objective Harris Hawks Optimization
MVO	Multi-Verse Optimizer
NCS	Network Control Systems
NGSA	Non-dominated Sorting Genetic Algorithm
PPPSO	Predator Prey Particle Swarm Optimizations
SSA	Salp Swarm Algorithm

SADE	Self Adaptive Differential Evolution
SCA	Sine Cosine Algorithm
TLBO	Teaching Learning Based Optimization
WOA	Whale Optimization Algorithm
WMN	Wireless Mesh Networks
WIA-PA	Wireless network for Industrial Automation-Process Automation

References

1. Zhang, X.M.; Han, Q.L.; Ge, X.; Ding, D.; Ding, L.; Yue, D.; Peng, C. Networked control systems: A survey of trends and techniques. *IEEE/CAA J. Autom. Sin.* **2019**, *7*, 1–17. [CrossRef]
2. Abdulrab, H.; Hussin, F.A.; Abd Aziz, A.; Awang, A.; Ismail, I.; Devan, P.A.M. Reliable fault tolerant-based multipath routing model for industrial wireless control systems. *Appl. Sci.* **2022**, *12*, 544. [CrossRef]
3. Devan, P.A.M.; Hussin, F.A.; Ibrahim, R.; Bingi, K.; Khanday, F.A. A Survey on the application of WirelessHART for industrial process monitoring and control. *Sensors* **2021**, *21*, 4951. [CrossRef] [PubMed]
4. Tran, D.C.; Ibrahim, R.; Hussin, F.A.; Omar, M. Energy-efficient superframe scheduling in industrial wireless networked control system. In *Proceedings of the 11th National Technical Seminar on Unmanned System Technology (NUSYS'19)*; Springer: Singapore, 2021; pp. 1227–1242.
5. Park, P.; Ghadikolaei, H.S.; Fischione, C. Proactive fault-tolerant wireless mesh networks for mission-critical control systems. *J. Netw. Comput. Appl.* **2021**, *186*, 103082. [CrossRef]
6. Hirata, A.; Oda, T.; Saito, N.; Kanahara, K.; Hirota, M.; Katayama, K. Approach of a solution construction method for mesh router placement optimization problem. In Proceedings of the 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), Kobe, Japan, 13–16 October 2020; pp. 467–468.
7. Devan, P.A.M.; Ibrahim, R.; Omar, M.B.; Bingi, K.; Abdulrab, H.; Hussin, F.A. Improved Whale Optimization Algorithm for Optimal Network Coverage in Industrial Wireless Sensor Networks. In Proceedings of the 2022 International Conference on Future Trends in Smart Communities (ICFTSC), Kuching, Malaysia, 1–2 December 2022; pp. 124–129.
8. Chai, Y.; Zeng, X.J. Regional condition-aware hybrid routing protocol for hybrid wireless mesh network. *Comput. Netw.* **2019**, *148*, 120–128. [CrossRef]
9. Devan, P.A.M.; Hussin, F.A.; Ibrahim, R.; Bingi, K.; Abdulrab, H. Design of fractional-order predictive PI controller for real-time pressure process plant. In Proceedings of the 2021 Australian & New Zealand control conference (ANZCC), Gold Coast, QLD, Australia, 25–26 November 2021; pp. 86–91.
10. Obeidat, H.; Shuaieb, W.; Obeidat, O.; Abd-Alhameed, R. A review of indoor localization techniques and wireless technologies. *Wirel. Pers. Commun.* **2021**, *119*, 289–327. [CrossRef]
11. Abdulrab, H.; Hussin, F.A.; Awang, A.; Ismail, I.; Devan, P.A.M.; Shutari, H. Optimal Node Placement and Congestion Reduction in an Industrial Wireless Mesh Network using HHO Algorithm. In Proceedings of the IEEE 2022 International Conference on Future Trends in Smart Communities (ICFTSC), Kuching, Malaysia, 1–2 December 2022; pp. 164–169.
12. Taleb, S.M.; Meraihi, Y.; Mirjalili, S.; Acheli, D.; Ramdane-Cherif, A.; Gabis, A.B. Mesh Router Nodes Placement for Wireless Mesh Networks Based on an Enhanced Moth–Flame Optimization Algorithm. *Mob. Netw. Appl.* **2023**. [CrossRef]
13. Taleb, S.M.; Meraihi, Y.; Gabis, A.B.; Mirjalili, S.; Zaguia, A.; Ramdane-Cherif, A. Solving the mesh router nodes placement in wireless mesh networks using coyote optimization algorithm. *IEEE Access* **2022**, *10*, 52744–52759. [CrossRef]
14. Devan, P.A.M.; Hussin, F.A.B.; Ibrahim, R.; Bingi, K.; Abdulrab, H.Q. Fractional-order predictive PI controller for dead-time processes with set-point and noise filtering. *IEEE Access* **2020**, *8*, 183759–183773. [CrossRef]
15. Joseph, S.B.; Dada, E.G.; Abidemi, A.; Oyewola, D.O.; Khammas, B.M. Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon* **2022**, *8*, e09399. [CrossRef]
16. Kazikova, A.; Pluhacek, M.; Senkerik, R. Why tuning the control parameters of metaheuristic algorithms is so important for fair comparison? In Proceedings of the 38th International Conference on Mathematical Methods in Economics 2020 (MME 2020), Online, 9–11 September 2020; Volume 26, pp. 9–16.
17. Ulusoy, S.; Nigdeli, S.M.; Bekdaş, G. Novel metaheuristic-based tuning of PID controllers for seismic structures and verification of robustness. *J. Build. Eng.* **2021**, *33*, 101647. [CrossRef]
18. Golilarz, N.A.; Addeh, A.; Gao, H.; Ali, L.; Roshandeh, A.M.; Munir, H.M.; Khan, R.U. A new automatic method for control chart patterns recognition based on ConvNet and harris hawks meta heuristic optimization algorithm. *IEEE Access* **2019**, *7*, 149398–149405. [CrossRef]
19. Devan, P.; Hussin, F.A.; Ibrahim, R.B.; Bingi, K.; Nagarajapandian, M.; Assaad, M. An arithmetic-trigonometric optimization algorithm with application for control of real-time pressure process plant. *Sensors* **2022**, *22*, 617. [CrossRef]
20. Yousri, D.; AbdelAty, A.M.; Said, L.A.; Elwakil, A.; Maundy, B.; Radwan, A.G. Parameter identification of fractional-order chaotic systems using different meta-heuristic optimization algorithms. *Nonlinear Dyn.* **2019**, *95*, 2491–2542. [CrossRef]
21. Omar, M.B.; Bingi, K.; Prusty, B.R.; Ibrahim, R. Recent advances and applications of spiral dynamics optimization algorithm: A review. *Fractal Fract.* **2022**, *6*, 27. [CrossRef]

22. Wasim, M.S.; Amjad, M.; Habib, S.; Abbasi, M.A.; Bhatti, A.R.; Muyeen, S. A critical review and performance comparisons of swarm-based optimization algorithms in maximum power point tracking of photovoltaic systems under partial shading conditions. *Energy Rep.* **2022**, *8*, 4871–4898. [CrossRef]
23. Wolpert, D.H. What is important about the No Free Lunch theorems? In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*; Springer: Cham, Switzerland, 2021; pp. 373–388.
24. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
25. Abdulrab, H.Q.; Hussin, F.A.; Ismail, I.; Assaad, M.; Awang, A.; Shutari, H.; Devan, P.A.M. Hybrid Harris Hawks With Sine Cosine for Optimal Node Placement and Congestion Reduction in an Industrial Wireless Mesh Network. *IEEE Access* **2023**, *11*, 2500–2523. [CrossRef]
26. Kamboj, V.K.; Nandi, A.; Bhadoria, A.; Sehgal, S. An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Appl. Soft Comput.* **2020**, *89*, 106018. [CrossRef]
27. Elgamal, Z.M.; Yasin, N.B.M.; Tubishat, M.; Alswaitti, M.; Mirjalili, S. An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field. *IEEE Access* **2020**, *8*, 186638–186652. [CrossRef]
28. Alabool, H.M.; Alarabiat, D.; Abualigah, L.; Heidari, A.A. Harris hawks optimization: A comprehensive review of recent variants and applications. *Neural Comput. Appl.* **2021**, *33*, 8939–8980. [CrossRef]
29. Yousri, D.; Babu, T.S.; Fathy, A. Recent methodology based Harris Hawks optimizer for designing load frequency control incorporated in multi-interconnected renewable energy plants. *Sustain. Energy Grids Netw.* **2020**, *22*, 100352. [CrossRef]
30. Li, C.; Li, J.; Chen, H.; Jin, M.; Ren, H. Enhanced Harris hawks optimization with multi-strategy for global optimization tasks. *Expert Syst. Appl.* **2021**, *185*, 115499. [CrossRef]
31. Too, J.; Abdullah, A.R.; Mohd Saad, N. A new quadratic binary harris hawk optimization for feature selection. *Electronics* **2019**, *8*, 1130. [CrossRef]
32. Dokeroglu, T.; Deniz, A.; Kiziloz, H.E. A robust multiobjective Harris' Hawks Optimization algorithm for the binary classification problem. *Knowl.-Based Syst.* **2021**, *227*, 107219. [CrossRef]
33. Yousri, D.; Allam, D.; Eteiba, M.B. Optimal photovoltaic array reconfiguration for alleviating the partial shading influence based on a modified harris hawks optimizer. *Energy Convers. Manag.* **2020**, *206*, 112470. [CrossRef]
34. Hussain, K.; Zhu, W.; Salleh, M.N.M. Long-term memory Harris' hawk optimization for high dimensional and optimal power flow problems. *IEEE Access* **2019**, *7*, 147596–147616. [CrossRef]
35. Houssein, E.H.; Saad, M.R.; Hussain, K.; Zhu, W.; Shaban, H.; Hassaballah, M. Optimal sink node placement in large scale wireless sensor networks based on Harris' hawk optimization algorithm. *IEEE Access* **2020**, *8*, 19381–19397. [CrossRef]
36. Srinivas, M.; Amgoth, T. EE-hHHSS: Energy-efficient wireless sensor network with mobile sink strategy using hybrid Harris hawk-salp swarm optimization algorithm. *Int. J. Commun. Syst.* **2020**, *33*, e4569. [CrossRef]
37. Munagala, V.K.; Jatoth, R.K. Design of fractional-order PID/PID controller for speed control of DC motor using Harris Hawks optimization. In *Intelligent Algorithms for Analysis and Control of Dynamical Systems*; Springer: Singapore, 2021; pp. 103–113.
38. Arini, F.Y.; Chiewchanwattana, S.; Soomlek, C.; Sunat, K. Joint Opposite Selection (JOS): A premiere joint of selective leading opposition and dynamic opposite enhanced Harris' hawks optimization for solving single-objective problems. *Expert Syst. Appl.* **2022**, *188*, 116001. [CrossRef]
39. Fu, W.; Lu, Q. Multiobjective optimal control of FOPID controller for hydraulic turbine governing systems based on reinforced multiobjective Harris Hawks optimization coupling with hybrid strategies. *Complexity* **2020**, *2020*, 1–17. [CrossRef]
40. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
41. Hu, G.; Zhong, J.; Du, B.; Wei, G. An enhanced hybrid arithmetic optimization algorithm for engineering applications. *Comput. Methods Appl. Mech. Eng.* **2022**, *394*, 114901. [CrossRef]
42. Premkumar, M.; Jangir, P.; Kumar, B.S.; Sowmya, R.; Alhelou, H.H.; Abualigah, L.; Yildiz, A.R.; Mirjalili, S. A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: Diversity analysis and validations. *IEEE Access* **2021**, *9*, 84263–84295. [CrossRef]
43. Khatir, S.; Tiachacht, S.; Le Thanh, C.; Ghandourah, E.; Mirjalili, S.; Wahab, M.A. An improved Artificial Neural Network using Arithmetic Optimization Algorithm for damage assessment in FGM composite plates. *Compos. Struct.* **2021**, *273*, 114287. [CrossRef]
44. Omar, M.B.; Ibrahim, R.; Mantri, R.; Chaudhary, J.; Ram Selvaraj, K.; Bingi, K. Smart Grid Stability Prediction Model Using Neural Networks to Handle Missing Inputs. *Sensors* **2022**, *22*, 4342. [CrossRef]
45. Salehuddin, N.F.; Omar, M.B.; Ibrahim, R.; Bingi, K. A Neural Network-Based Model for Predicting Saybolt Color of Petroleum Products. *Sensors* **2022**, *22*, 2796. [CrossRef]
46. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of COVID-19 ct images. *Processes* **2021**, *9*, 1155. [CrossRef]
47. Zheng, R.; Jia, H.; Abualigah, L.; Liu, Q.; Wang, S. Deep ensemble of slime mold algorithm and arithmetic optimization algorithm for global optimization. *Processes* **2021**, *9*, 1774. [CrossRef]
48. Elsisi, M.; Tran, M.Q.; Hasanien, H.M.; Turky, R.A.; Albalawi, F.; Ghoneim, S.S. Robust model predictive control paradigm for automatic voltage regulators against uncertainty based on optimization algorithms. *Mathematics* **2021**, *9*, 2885. [CrossRef]

49. Zheng, R.; Jia, H.; Abualigah, L.; Liu, Q.; Wang, S. An improved arithmetic optimization algorithm with forced switching mechanism for global optimization problems. *Math. Biosci. Eng.* **2022**, *19*, 473–512. [CrossRef]
50. Agushaka, J.O.; Ezugwu, A.E. Advanced arithmetic optimization algorithm for solving mechanical engineering design problems. *PLoS ONE* **2021**, *16*, e0255703. [CrossRef] [PubMed]
51. Elkasem, A.H.; Khamies, M.; Magdy, G.; Taha, I.; Kamel, S. Frequency stability of AC/DC interconnected power systems with wind energy using arithmetic optimization algorithm-based fuzzy-PID controller. *Sustainability* **2021**, *13*, 12095. [CrossRef]
52. Xu, Y.P.; Tan, J.W.; Zhu, D.J.; Ouyang, P.; Taheri, B. Model identification of the proton exchange membrane fuel cells by extreme learning machine and a developed version of arithmetic optimization algorithm. *Energy Rep.* **2021**, *7*, 2332–2342. [CrossRef]
53. Ewees, A.A.; Al-qaness, M.A.; Abualigah, L.; Oliva, D.; Algamal, Z.Y.; Anter, A.M.; Ali Ibrahim, R.; Ghoniem, R.M.; Abd Elaziz, M. Boosting arithmetic optimization algorithm with genetic algorithm operators for feature selection: Case study on cox proportional hazards model. *Mathematics* **2021**, *9*, 2321. [CrossRef]
54. Selvam, A.M.D.P.; Hussin, F.A.; Ibrahim, R.; Bingi, K.; Nagarajapandian, M. *Optimal Fractional-order Predictive PI Controllers: For Process Control Applications with Additional Filtering*; Springer Nature: Singapore, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.