JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-19, No.-01, January (2024) pp 27-46 ISSN (Print) 0973-8975

A NOVEL HYBRID HARMONY SEARCH (HS) WITH WAR STRATEGY OPTIMIZATION (WSO) FOR SOLVING OPTIMIZATION PROBLEMS

Sameerah Khaleel[1], Hegazy Zaher[2], Naglaa Ragaa Saeid[3]

[1]PhD student, Department Of Operations Research, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt.

[2]Professor, Department of Statistics, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt.

[3]Assoociate Professor, Department of Operations Research, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt.

Email: samerakhaleel212@gmail.com

https://doi.org/10.26782/jmcms.2024.01.00003

(Received: November 22, 2023; Revised: January 10, 2024; Accepted: January 15, 2024)

Abstract

The usage of nature-inspired meta-heuristic algorithms is increasing due to their simplicity and versatility. These algorithms are widely used in numerous domains, especially in scientific fields such as operations research, computer science, artificial intelligence, and mathematics. Based on the core principles of exploration and exploitation, they provide flexible problem-solving abilities. This study presents a novel method to improve the effectiveness of the War Strategy Optimization (WSO) algorithm for optimization issues. The suggested approach combines the WSO technique with the Harmony Search (HS) algorithm, resulting in a hybrid algorithm called H-WSO. The aim is to enhance the overall optimization performance by leveraging the capabilities of both algorithms through the integration of swarm intelligence approaches.

In order to assess the effectiveness of the recently suggested H-WSO algorithm, a set of experiments was carried out on 50 benchmark test functions. These functions included both unimodal and multimodal functions and spanned across different dimensions. The findings from these studies clearly showed a notable enhancement in the efficiency of the H-WSO algorithm when compared to the original WSO algorithm. Various metrics were utilized to evaluate the effectiveness of the proposed algorithm, including the optimal fitness function value (Mean), Standard Deviation (St.d), and Median. The H-WSO algorithm regularly shows higher efficiency than the WSO algorithm, making it a promising and practical approach for addressing complicated optimization challenges.

Keywords : Meta-heuristic Algorithms, War Strategy Optimization algorithm, Harmony Search algorithm, Hybrid method.

I. Introduction

Optimization problems include identifying the most favorable solution from a range of possible solutions, typically to maximize or minimize an objective function while satisfying particular constraints. The issues encompass a wide range of disciplines, including mathematics, computer science, engineering, economics, and operations research. There are two main types of optimization problems: continuous optimization problems, which include linear programming, quadratic programming, and nonlinear programming, and discrete optimization problems, which include the traveling salesman problem, the knapsack problem, and the graph-coloring problem. To address these problems, one must utilize a range of algorithms and techniques, including gradient descent, linear programming solvers, evolutionary algorithms, dynamic programming, and branch and bound methods. The choice of the most appropriate optimization method depends on the specific attributes of the problem, including the types of variables, constraints, and objective function. Optimization is crucial for improving efficiency, production, and overall effectiveness [XXVI].

Meta-heuristic algorithms are optimization techniques that are inspired by natural events, aiming to mimic the optimization strategies observed in biological systems, physics, or chemistry. These algorithms are specifically tailored for problem domains that lack clear-cut or intricate answers. The objective is to achieve a harmonious equilibrium between identifying an appropriate resolution and the duration necessary for the investigation. While it may not provide the most optimal option, it does provide a satisfactory compromise. The process uses rules or criteria to choose the optimal solution from a set of solutions through numerous iterations or cycles, progressively improving the outcomes [XII].

Swarm Intelligence Algorithms (SIA) are a type of meta-heuristics that are specifically influenced by the collective behavior of social insect colonies, such as ants, bees, or birds. These algorithms imitate the decentralized and self-organized characteristics of social insect colonies, with a focus on promoting cooperation and interaction among individuals in the swarm. Structured Intelligent Agents (SIAs) possess notable benefits like parallelism, robustness, and adaptability. They possess the ability to proficiently resolve intricate optimization and decision-making issues. The efficacy of a particular Social Impact Assessment (SIA) is contingent upon the inherent characteristics of the issue being addressed and the meticulous calibration of its variables. The user's text is "[XXIX]." Various issues may necessitate distinct swarm intelligence methodologies. Several approaches, including Particle Swarm Optimization (PSO), exist [VII]. The topic being referred to is the Genetic Algorithm [XVI]. The Firefly Algorithm is a computational optimization technique [XVII].The user's text is "[I].". The following optimization algorithms are mentioned: Cat Swarm Optimization (CSO)[XVIII], Ant Colony Optimization (ACO)[X], Harmony Search (HS)[XXXIV], Artificial Bee Colony (ABC)[IV], Flower Pollination Algorithm (FPA) [XXVII] [XIV], Cuckoo Search Algorithm (CSA)[XXII], and Hunger Game Search (HGS) [XXX].

Exploration and exploitation are fundamental components of meta-heuristic techniques. The exploration entails the systematic investigation of potential solutions over the entirety of the solution space, encompassing both known and uncharted territories. Exploration facilitates the identification of novel and potentially advantageous solutions that have the potential to enhance overall performance. It mitigates the algorithm's tendency to become trapped in local optima, whereas Exploitation concentrates on enhancing the search in regions that are probable to harbor solutions near the optimal ones. After identifying places with potential, the algorithm focuses its search in those areas to move closer to the optimal answers. Achieving a balance between exploration and exploitation is a crucial obstacle when it comes to building efficient meta-heuristic algorithms[III].

II. War Strategy Optimization (WSO)

A novel meta-heuristic optimization technique named WSO was introduced by Ayyarao et al. in 2022 [XXIII], taking inspiration from historical military tactics. This algorithm integrates two primary military strategies: offensive and defensive.

The core principle of war strategy revolves around the synchronized deployment of forces by the monarch and military leaders, to effectively reach designated targets and accomplish predetermined goals. War strategy is a fluid and progressive procedure, in which the military forces cooperate and participate in conflicts against the enemy. This technique demonstrates flexibility in response to evolving circumstances throughout the course of the battle. The positions of the king and commanders exert a constant influence on the status of the troops in the army. The flags positioned on the king's and army commander's chariots function as conspicuous markers of their positions, easily seen by all soldiers [XXV]. Soldiers get training in tactics that rely on aural cues, such as the reverberations of drums or other musical apparatus. Upon the death of a military leader, there is a shift in the plan, requiring successive commanders to acquire the knowledge and skills necessary to reconstruct and sustain the fundamental aspects of the war strategy. The king's ultimate objective is to vanquish the enemy leader, while the troops of the army strive to launch an attack on the opposing faction and ascend in their military hierarchy.

III. Mathematical Model of WSO

During each cycle, all troops have an equal probability of being designated as either the King or Commander, based on their combat strength (Fitness Value). Both the King and the Commander serve as leaders on the battlefield. The strategic maneuvers of the King and the Commander on the battlefield will direct the actions of the remaining soldiers. Either the King or the Commander may encounter formidable opposition from the enemy's soldier (Local Optima) who possesses sufficient power to ensnare the Leaders. To prevent this, soldiers in conflict will adhere to the guidance provided by the King or Commander's position, as well as their collective movement methods.

a. Attack Strategy

It has devised two military strategies. Under the initial approach, each soldier adjusts their position according to the coordinates of both the King and the Commander. The process of upgrading the attack model is illustrated in Figure 1. The monarch strategically positions himself to initiate a formidable assault on the enemies. The soldier with the highest level of assault force or physical condition is considered to be the king.

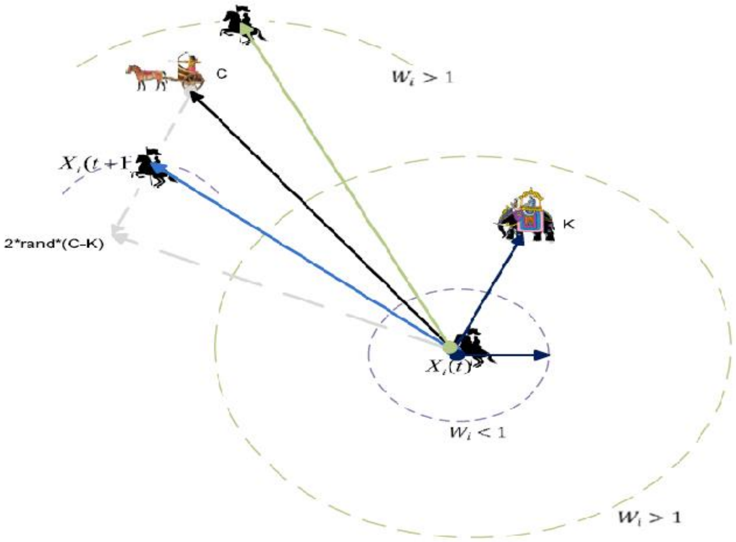

Fig.1. Attack Strategy [XXIII]

At the commencement of the conflict, all soldiers will possess identical ranks and weights. Should the soldier effectively implement the approach, his rank will increase. Nevertheless, as the fight advances, the rankings and burdens of all warriors will be revised by the triumph of the strategy. As the war draws to a close, the King, Army commander, and soldiers maintain a closely aligned position as they approach the goal.

$$\mathcal{X}_{\mathrm{i}}\,(t+1) = \mathcal{X}_{\mathrm{i}}(t) + 2 \times \rho \times (C - K) + rand \times \big(W_{\mathrm{i}} \times K - \mathcal{X}_{\mathrm{i}}(t)\big) \quad (1)$$

Where,

$\mathcal{X}_{\mathrm{i}}\,(t+1)$: is a new position.

$\mathcal{X}_{\mathrm{i}}$: is the previous C position of the commander.

K: is the position of the king.

$W_{\mathrm{i}} \times K - \mathcal{X}_{\mathrm{i}}(t)$ based on the king position. If >1, then the position of

The product of W_i and K, subtracted by $X_i(t)$, exceeds the position of the monarch, resulting in the soldier's position being further than that of the commander. If the value of W_i is less than 1, then the position of W_i multiplied by K minus $X_i(t)$ is located between the position of the monarch and the current position of the soldier. As W_i approaches zero, the soldier's updated position converges towards the commander's position, signifying the culmination of the war.

b. Rank and Weighing Updating

The update of each search agent's position is determined by the interplay between the positions of the King, the Commander, and the rank of each soldier. The ranking of each soldier is determined by their past performance on the battlefield, as controlled by equation (4), which in turn affects the weighting factor W_i. The rank of each soldier corresponds to the proximity of the soldier (search agent) to the objective (fitness value). If the fitness of the attack force at the new position (Fn) is lower than that of the prior position (Fp), the soldier will choose to return to the previous position.

$$\mathcal{X}_i(t+1) = \left(\mathcal{X}_i(t+1)\right) \times (F_n \geq F_p) + \left(\mathcal{X}_i(t)\right) \times (F_n < F_p) \quad (2)$$

If the soldier updates the position successfully, the rank R_i of the soldier will be upgraded.

$$R_i = (R_i + 1) \times (F_n \geq F_p) + (R_i) \times (F_n < F_p) \quad (3)$$

Based on the rank, a new weight is calculated as:

$$W_i = W_i \times \left(1 - \frac{R_i}{Max\_iter}\right)^{\alpha} \quad (4)$$

c. Defense Strategy

The second strategic position update relies on the coordinates of the King, the army commander, and a randomly selected soldier. The ranking and weight updating procedures remain unchanged.

$$\mathcal{X}_i(t+1) = \mathcal{X}_i(t) + 2 \times \rho \times \left(K - \mathcal{X}_{ran}(t)\right) + rand \times W_i \times \left(C - \mathcal{X}_i(t)\right) \quad (5)$$

d. Replacement or Relocation of Weak Soldiers

During each cycle, the soldiers with the lowest fitness are detected as weak and then replaced by superior soldiers. An uncomplicated method is substituting the feeble soldier with a randomly selected soldier, as indicated by the following equation.

$$\mathcal{X}w(t+1) = Lb + rand \times (Ub - Lb) \quad (6)$$

The operation of the WSO algorithm can be delineated as follows:

Step 1: Set the parameters to their initial values.

Specify the magnitude of the troop population (soldier size).

Specify the spatial dimension of the war (problem dimension).

Specify the minimum and maximum limits of the search area.

Set the initial positions of the King (K) and the Army Commander (C).

Specify the offensive troops under the command of the King and the Army Commander.

Set the size of the soldier to S.

Step 2. Set the initial values for the variables:

Define R as a zero vector with dimensions (1, soldier size).

Define vector W as a one-dimensional array of size (1, soldier size), where each element is equal to 2.

Step 3. Disperse the soldiers in the war space randomly and uniformly (random attack).

Step 4. Iterate through each soldier in the population using a for loop.

Determine the offensive strength of each soldier.

Step 5. Arrange the soldiers' fitness (attack force) in order.

Step 6. Choose the soldier with the highest level of physical fitness to be the King, and the soldier with the second-highest attack force to be the Army Commander.

Step 7. Instantiate a counter variable t and assign it a value of 0.

Step 8. Commence the primary iteration loop while t is below the maximum number of iterations. Maximum temperature: Perform a repetitive action for each soldier in the population using a for loop:

Produce a random value for the variable ρ.

If the value of ρ is below a specified threshold ρr (expressed as a percentage signal), execute the following actions (exploration):

Revise the location of every soldier utilizing equation (5).

Alternatively, carry out the following actions (Exploitation):

Revise the location of every soldier utilizing equation (1).

Determine the magnitude of the offensive power for every individual soldier.

Rank the physical condition of each soldier.

Revise the location of each soldier by considering the offensive strength of both their current and former places, utilizing equation (2).

Revise the ranking and weighting of each soldier according to their achievements utilizing equation (3).

The soldier iteration cycle has concluded.

Determine the soldier who exhibits the lowest level of physical fitness.

Reposition the feeble soldier by selecting an appropriate relocation alternative.

Revise the locations of the King and the Commander.

Increase the value of the counter t by 1.

Step 9. Termination of the primary iteration loop.

IV. HS Optimization

HS Optimization is a meta-heuristic method that seeks to achieve the optimal solution for a problem by harmoniously combining many components, much like composing a harmonious musical piece. The method achieves a balance between exploration and exploitation through the utilization of memory consideration, pitch altering, and randomization. It iteratively enhances the harmonies stored in memory until the most favorable solution is achieved.

The HS algorithm operates in the following manner [XXVIII]:

Step 1: Initialization Specify the problem to be resolved, encompassing the objective function, decision variables, and restrictions. Specify the algorithm's parameters, including the number and range of variables, as well as the maximum number of iterations. Create an initial population of potential solutions either by random selection or by employing a particular methodology.

Step 2. Harmony Memory Consideration (HMCR): Choose one or several candidate solutions from the memory that accurately represent the harmony memory. This stage involves the process of choosing musical notes or chords from a stored memory, much like selecting them.

Step 3. Pitch Adjusting (PAR): Modify the chosen values according to the chosen memory. This stage involves altering the chosen notes or chords in order to generate novel harmonies. The adjustment can be achieved through the implementation of various tactics such as randomness, improvisation, or drawing from memory.

Step 4. Randomization: Apply a randomization process to select and assign values to certain decision variables within the New Harmony.

This stage introduces exploration into the search process and serves to avoid the algorithm from prematurely converging on a local optimum.

Step 5. Assess and Revise: Calculate the value of the objective function for the updated harmony. If the new harmonic surpasses all existing harmonies in quality, update the memory by substituting the weakest harmony with the new one.

Step 6. Termination: Continuously execute stages 2-5 until a termination condition is satisfied, such as reaching the maximum allowable number of iterations or attaining a suitable solution. Output the minimum and maximum values of hm.

V. Hybrid Method

A hybrid method refers to a methodology that combines many strategies or algorithms to solve a problem or improve a solution. Within the domain of swarm intelligence algorithms, this approach entails combining swarm intelligence approaches with supplementary optimization or search algorithms to enhance their efficiency and overcome inherent constraints.

The combination of swarm intelligence algorithms with other approaches, such as classical optimization algorithms or search strategies, results in a hybrid strategy that offers several benefits. The benefits include improved solutions, faster convergence towards optimal or nearly optimal solutions, and better balancing of exploration-exploitation trade-offs. The user's text is "[XXIV]." Several hybrid algorithms exist, including a hybrid approach combining Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) [V]. This is a hybrid algorithm that combines Ant Colony Optimization (ACO) and Differential Evolution (DE) [VIII]. This is a hybrid resulting from the crossbreeding of a domestic cat with a wildcat species known as the Geoffroy's cat (Cat-GA) [II]. This is a hybrid algorithm that combines Ant Colony Optimization (ACO) with Simulated Annealing (SA) [XV, XIII]. This is a hybrid algorithm that combines the Cat Swarm Optimization (Cat-PSO) algorithm with another method.

VI. Proposed Hybrid H-WSO Algorithm

The H-WSO algorithm is a robust meta-heuristic optimization method that leverages the advantages of both the HS and WSO algorithms. HS is an optimization technique that is based on the idea of generating harmonies, whereas WSO is influenced by old military techniques. The H-WSO algorithm seeks to achieve an optimal solution by combining and harmonizing the exploration and exploitation aspects of these two algorithms. It accomplishes this by integrating memory-based randomization and pitch adjustment from HS, which consistently enhances the harmonies stored in the memory. WSO successfully combines the exploration and exploitation processes, leading to the finding of optimal solutions.

The H-WSO algorithm is particularly suitable for addressing optimization issues by combining the advantageous features of HS and WSO. This enables it to systematically search the solution space and effectively utilize the identified solutions to achieve the optimal solution.

6. Computational Experiments

To assess the effectiveness of the proposed H-WSO, fifty benchmark test functions are employed for verification. This study presents unimodal and multimodal functions, which are illustrated in Tables 1-2.

Table 1: Details of unimodal benchmark functions.

B.F.	Expression	F_{min}	Range	Dim.
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	0	[-100,100]	50
F2	$f(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1]$	0	[-1.28,1.28]	50
F3	$f(x) = \sum_{i=1}^{D} \lvert Xi \rvert^{i+1}$	0	[- 1,1]	50
F4	$f(x) = \sum_{i=1}^{n} \lvert Xi \rvert$	0	[-100,100]	50
F5	$f(x) = max_i \sum_{i=1}^{n} \lvert Xi \rvert$, $1 \leq i \leq n$	0	[-100,100]	50

F6	$f(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	0	[- 100,100]	50
F7	$f(x) = 25 + \sum_{i=1}^{n}([Xi])$	25-6n	[-5.12,5.12]	50
F8	$f(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	0	[-100,100]	50
F9	$f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	0	[-100,100]	50
F10	$f(x) = \sum_{i=1}^{n} x_i^{10}$	0	[-10,10]	50
F11	$f(x) = \sum_{i=1}^{n-1}[(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2]$	0	[-30,30]	50
F12	$f(x) = \sum_{i=1}^{n-1}(x_i^2)^{(x^2_{i+1}+1)} + (x^2_{i+1})^{x_{i+1}}$	0	[-1,4]	50
F13	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_{i\_1})^2$	0	[-10,10]	50
F14	$f(x) = \sum_{I=1}^{\frac{D}{4}}(x_{4i\_3} + 10x_{4i\_2})^2 + 5(x_{4i\_1} - x_{4i})^2 + (x_{4i\_2} - x_{4i\_1})^4 + 10(x_{4i\_3} - x_{4i})^4$	0	[-4,5]	50
F15	$f(x) = \sum_{i=1}^{n}(x_i)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$	0	[-5,10]	50
F16	$f(x) = \exp(-\sum_{i=1}^{n}(x_i/\beta)^{2m}) - 2$	-1	[-20,20]	50
F17	$f(x) = \sum_{i=1}^{d}[\sum_{i=1}^{d}(j+\beta)(x_j^i - \frac{1}{j^i})]^2$	0	[-d_i, d_i]	5
F18	$f(x)= 2x_1^2 - 1.05x_1^4 + \frac{x_i^6}{6} + x_1x_2 + x_2^2$	0	[-5, 5]	2
F19	$f(x)=(1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x^1 + x_1x_2^3)^2$	0	[-4.5,4.5]	2
F20	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	0	[-10,10]	2
F21	$f(x) = (x_1 + 10)^2 + (x_2 + 10)^2 + e^{-x_1^2 - x_2^2}$	0	[-10,10]	2
F22	$f(x) = 0.26(x^{12} + x^{22}) - 0.48x_1x_2$	0	[-10,10]	2
F23	$f(x) = 0.5 + \frac{cos^2(\sin(\lvert x^2 - y^2\rvert)) - 0.5}{(1 + 0.001(x^2 + y^2))^2}$	0.29257	[-100,100]	2

F24	$f(x) = 2\frac{x^{13}}{3} - 8x^{12} + 33x^1 - x^1x^2 + 5 + [(x^1 - 4)^2 + (x^2 - 5)^2 - 4]^2$	21.35	[-500,500]	2
F25	$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	0	[-1.2,1.2]	2

Table 2: Details of multimodal benchmark functions.

B.F.	Expression	F_{min}	Range	Dim.
F26	$f(x) = 418.9829n - \sum_{i=1}^{n} -x_i sin\sqrt{x_i}$	0	[-500,500]	50
F27	$f(x) = \sum_{i=1}^{n}[x^2{}_i - 10\cos(2\pi x_i) + 10]$	0	[-5.12,5.12]	50
F28	$f(x) = 1+ \sum_{i=1}^{n} sin^2(x_i) - 0.1e^{(\sum_{i=1}^{n} x_i^2)}$	0.9	[-10,10]	50
F29	$f(x) = \sum_{i=1}^{n}(x^2 - i)^2$	0	[-500,500]	50
F30	$f(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	0	[-10,10]	50
F31	$f(x) = \sum_{i=1}^{n} \in_i \lvert x_i \rvert^i$	0	[-5,5]	50
F32	$f(x) = -20\, e^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)} - e^{\left(\frac{1}{n\sum_{i=1}^{n}\cos(2\pi x_i)}\right)} + 20 + e$	0	[-32,32]	50
F33	$f(x) = \sum_{i=1}^{n} 8sin^2(\,7\{x_i - 0.9\}^2) + 6sin^2(14\{x_i - 0.9\}^2) + (x_i - 0.9)^2$	1	[-500,500]	50
F34	$f(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$	0	[-100,100]	50
F35	$f(x) = 1/2\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)$	-39.16599	[-5,5]	50
F36	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{\sqrt{i}}}\right) + 1$	0	[-100,100]	50
F37	$f(x) = (\sum_{i=1}^{n} sin^2(x_i) - e^{\sum_{i=1}^{n} x_i^2})e^{\sum_{i=1}^{n} sin^2(\sqrt{x_i})}$	-1	[-10.10]	50

F38	$f(x) = (\sum_{i=1}^{n} \lvert x_i \rvert) e^{-\sum_{i=1}^{n} x_i^2}$	0	[-2π,2π]	50
F39	$f(x) = 0.1\left[sin^2(3\pi x_i) + \sum_{i=1}^{n} (x_i - 1)^2\{1 + sin^2(3\pi x_i + 1)\} + (x_n - 1)^2\{1 + sin^2(2\pi x_n)\}\right] + \sum_{i=1}^{n} u(x_i, 5,100,4)$	0	[-50,50]	50
F40	$f(x) = \pi/n\left[10sin(\pi x_i) + \sum_{i=1}^{n} (x_i - 1)^2\{1 + 10sin^2(\pi x_{i+1})\} + (x_n - 1)^2\right] + \sum_{i=1}^{n} u(x_i, 5,100,4)$	0	[-50,50]	50
F41	$f(x) = x^2+y^2+ 25(sin^2(x)+sin^2(y))$	0	[-5.5]	2
F42	$f(x) = -200e^{2\sqrt{x^2+y^2}} + 5e^{\cos(3x)+\sin(3y)}$	-195.629	[-32,32]	2
F43	$f(x) = Cos(x)sin(y) - \frac{x}{y^2+1}$	-2.02181	[-1,2]	2
F44	$f(x) = \sin(x)\, e^{(1-\cos(y))^2} + \cos(y)e^{(1-\cos(x))^2}$	- 106.7645	[-2π,2π]	2
F45	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3} x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	-1.0316	[-5,5]	2
F46	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 - \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	0.3978873	[-5,10]	2
F47	$f(x) = -\sum_{i=1}^{4} c_i e^{(\sum_{j=1}^{3} a_{ij}\{x_j - p_{ij}\}^2)}$	- 3.862782	[0,1]	3
F48	$f(x) = \sum_{i=1}^{4} c_i e^{(\sum_{j=1}^{6} a_{ij}\{x_j - p_{ij}\}^2)}$	-3.32237	[0,1]	6
F49	$f(x) = -0.0001(\lvert \sin(x)\sin(y)\exp(100 - \frac{\sqrt{x^2+y^2}}{\pi}) \rvert + 0.1)^{0.1}$	-2.062	[-10,10]	2
F50	$f(x) = \lvert x^2+y^2+xy \rvert + \lvert \sin(x) \rvert + \lvert \cos(y) \rvert$	1	[-500,500]	2

VII. Setting of Parameters

To assess the performance of H-WSO, we compare it to the original WSO. A performance comparison is conducted between WSO and H-WSO using 10 independent runs for each benchmark function. Each test consists of a 50-population size and a maximum of 1000 iterations. Three performance evaluation measures employed in H-WSO are the mean of the best fitness function value, the standard deviation, and the median for fitness functions. Table 3-4 displays the data that establish the superiority of H-WSO over WSO.

Table 3: Comparison between WSO results and H-WSO results on unimodal benchmark test functions

Function	WSO Algorithm			H-WSO Algorithm		
	Mean	**St.d**	**Median**	**Mean**	**St.d**	**Median**
F1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F2	7.00E-05	5.21E-05	3.88E-05	**2.14E-06**	**3.59E-05**	**1.73E-05**
F3	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F4	6.31E-259	**0.00E+00**	2.53E-263	**3.87E-261**	2.81E-261	**2.31E-264**
F5	9.27E-260	**0.00E+00**	3.53E-262	**1.27E-267**	3.37E-261	**2.18E-264**
F6	1.51E-08	4.26E-08	8.21E-10	**1.03E-09**	**1.28E-09**	**1.25E-11**
F7	-2.75E+02	**0.00E+00**	-2.75E+02	**-3.97E+02**	**0.00E+00**	**-2.41E+04**
F8	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F9	6.47E-258	**0.00E+00**	**8.72E-261**	**5.50E-259**	**0.00E+00**	3.74E-260
F10	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	1.15E-07	2.61E-07	**1.95E-09**	**1.11E-11**	**2.31E-11**	4.53E-08
F12	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F13	2.46E-01	**0.00E+00**	2,49E-01	**5.95E-03**	**0.00E+00**	**3.53E-03**
F14	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F15	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F16	0.00E+00	**0.00E+00**	**0.00E+00**	**-1.29E-15**	**0.00E+00**	**0.00E+00**
F17	4.39E-01	9.23E-01	**1.14E-01**	**4.29E-01**	**8.34E-01**	1.42E-01
F18	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F19	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F20	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F21	1.38E-87	2.35E-103	1.38E-87	**2.38E-89**	**2.27E-107**	**1.48E-89**
F22	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F23	2.93E-01	4.53E-17	2.93E-01	**2.82E-01**	**3.12E-17**	**2.81E-01**
F24	1.91E+01	**4.53E-17**	**1.91E+01**	**1.57E+01**	4.56E-17	1.92E+01
F25	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

Table 4: Comparison between WSO results and H-WSO results on multimodal benchmark test functions

Functions	WSO Algorithm			H-WSO Algorithm		
	Mean	St.d	Median	Mean	St.d	Median
F26	1.27E-05	4.04E-06	1.27E-05	**2.82E-08**	**3.13E-06**	**1.05E-05**
F27	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F28	**9.00E-01**	**1.17E-16**	**9.00E-01**	**9.00E-01**	**1.17E-16**	**9.00E-01**
F29	3.83E+03	**6.02E+02**	3.66E+03	**3.09E+03**	6.11E+02	**3.58E+03**
F30	2.30E-259	**0.00E+00**	2.22E-261	**3.05E-261**	1.62E-02	1.67E-258
F31	7.77E-39	2.46E-38	3.15E-70	**8.56E-45**	**6.08E-47**	**2.89E-71**
F32	-8.88E-16	**0.00E+00**	-8.88E-16	**-3.09E-15**	**0.00E+00**	**-8.56E-14**
F33	**1.00E+00**	4.90E-03	**1.00E+00**	**1.00E+00**	**4.29E-08**	**1.00E+00**
F34	2.73E-42	8.65E-42	4.09E-106	**4.09E-54**	**3.09E-48**	**4.44E-109**
F35	**-1.96E+03**	5.02E-08	**-1.96E+03**	**-1.96E+03**	**3.33E-16**	**-1.96E+03**
F36	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F37	**-1.00E+00**	**0.00E+00**	**-1.00E+00**	**-1.00E+00**	**0.00E+00**	**-1.00E+00**
F38	1.22E-20	2.52E-22	1.21E-20	**1.76E-31**	**8.93E-27**	**2.21E-29**
F39	2.32E-07	2.72E-07	9.05E-08	**2.03E-12**	**4.81E-11**	**7.09E-10**
F40	8.81E-11	2.28E-10	5.34E-12	**8.43E-15**	**2.17E-10**	**2.45E-12**
F41	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F42	**-1.96E+02**	3.00E-14	**-1.96E+02**	**-1.96E+02**	**3.09E-18**	**-1.96E+02**
F43	**-2.02E+00**	4.68E-16	**-2.02E+00**	**-2.02E+00**	2.43E-09	**-2.02E+00**
F44	-1.07E+02	9.47E-15	**-1.07E-15**	**-1.92E+02**	**5.43E-15**	**-1.07E-15**
F45	-1.03E+00	**0.00E+00**	**-1.03E+00**	**-1.23E+01**	1.87E-12	**-1.03E+00**
F46	3.98E-01	**0.00E+00**	**3.98E-01**	**9.80E-02**	3.65E-13	**3.98E-01**
F47	3.00E+00	1.32E-15	3.00E+00	**-3.76E+00**	**1.44E-17**	**2.31E+00**
F48	-3.86E+00	9.00E-16	**-3.86E+00**	**-3.76E+01**	**5.89E-18**	**-3.09E+00**
F49	-3.27E+00	6.14E-02	-3.32E+00	**-3.76E+00**	**6.45E-03**	**-3.62E+02**
F50	-2.06E+00	4.68E-16	**-2.06E+00**	**-2.82E+00**	**6.33E-17**	**-2.06E+00**

When comparing H-WSO to WSO, it becomes evident that there are clear performance benefits in several benchmark test functions.

• The benchmark test functions show that H-WSO performs much better than WSO in the unimodal functions F2, F4, F5, F6, F7, F9, F11, F13, F16, F17, F21, F23, and F24. Nevertheless, both algorithms produce comparable outcomes in the remaining operations.

• H-WSO demonstrates superior performance than WSO in F26, F29, F30, F31, F32, F34, F38, F39, F40, F44, F45, F46, F47, F48, F49, and F50 in the multimodal benchmark test functions, while exhibiting comparable performance in the remaining functions.

In summary, the results indicate that the suggested H-WSO improves both the process of exploring new opportunities and the process of using existing resources. This enhancement enables meta-heuristics to explore the search space with more precision. Figures 2-7 provide comprehensive metrics comparisons between H-WSO and WSO.

The suggested optimization method can be utilized in many branches of computer science and engineering to enhance their performance [VI, IX, XI, XIX, XX, XXI, XXXI, XXXII, XXXIII].

Fig. 2. Comparison of Fitness Function Value (Mean) between WSO and H-WSO for Unimodal functions

Fig. 3. Comparison of Fitness Function Value (Mean) between WSO and H-WSO for Multimodal functions

Fig. 4. Comparison of St.d Value between WSO and H-WSO for Unimodal functions

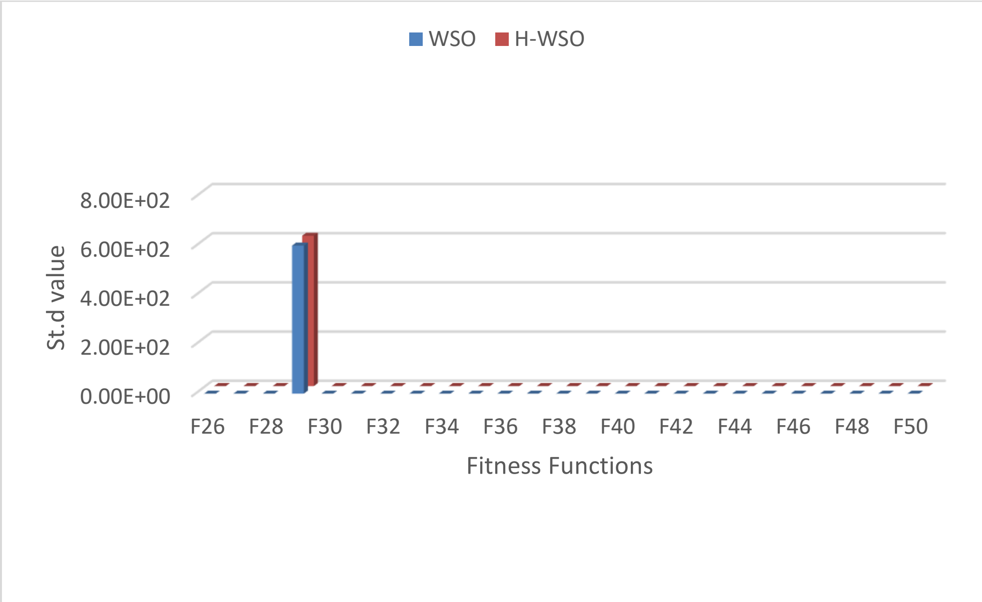

Fig. 5. Comparison of St.d Value between WSO and H-WSO for Multimodal functions

Fig. 6. Comparison of Median Value between WSO and H-WSO for Unimodal functions

Fig. 7. Comparison of Median Value between WSO and H-WSO for Multimodal functions

VIII. Conclusion

A hybrid technique has been presented to enhance the effectiveness of the (WSO) algorithm in addressing optimization difficulties. This paper presents the combination of WSO with HS, referred to as H-WSO. To evaluate the algorithm's performance, it is subjected to testing on fifty standardized benchmark test functions. The results demonstrate that H-WSO exhibits significant improvements compared to the original WSO algorithm. When compared to WSO, it demonstrates enhanced performance by achieving higher fitness function values (Mean), standard deviation, and Median. This work highlights the effectiveness of the hybrid H-WSO algorithm in optimizing solutions for various test functions, providing evidence of its superiority over WSO in terms of efficiency and overall performance.

Conflict of Interest:

The authors declares that there was no conflict of interest regarding this paper.

References

I. A. Prof. Sabah Manfi, et al., “Use the Firefly Algorithm to Find the Profits of the Airlines at Baghdad International Airport,” no. 2018, pp. 137–149.

II. A. Kekli, "Hybrid Cat Swarm Optimization with Genetic Algorithm to solve the Open Shop Scheduling with Vehicle Routing Problem Hybrid Cat Swarm Optimization with Genetic Algorithm to solve the Open Shop," pp. 0–34, 2023.

III. D. M. El-Sadek and others, "Solve global optimization problems based on metaheuristic algorithms," *Bull. Fac. Sci. Zagazig Univ.*, vol. 2022, no. 3, pp. 29–42, 2022.

IV. D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *J. Glob. Optim.*, vol. 39, pp. 459–471, 2007.

V. D. T. Nguyen, J. R. Ho, P. C. Tung, and C. K. Lin, "A Hybrid PSO–GWO Fuzzy Logic Controller with a New Fuzzy Tuner," *Int. J. Fuzzy Syst.*, vol. 24, no. 3, pp. 1586–1604, 2022, doi: 10.1007/s40815-021-01215-6.

VI. F. Abayaje, S. A. Hashem, H. S. Obaid, Y. S. Mezaal, and S. K. Khaleel, "A miniaturization of the UWB monopole antenna for wireless baseband transmission," *Periodicals of Engineering and Natural Sciences*, vol. 8, no. 1, pp. 256–262, 2020.

VII. H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Appl. Math. Comput.*, vol. 274, pp. 292–305, 2016.

VIII. H. Li, X. Zhang, S. Fu, and Y. Hu, "A hybrid algorithm based on ant colony optimization and differential evolution for vehicle routing problem," *Eng. Lett.*, vol. 29, no. 3, pp. 1201–1211, 2021.

IX. H. Bećirspahić, H. Basarić, T. Namas, and B. Durakovic, "Minimum viable product: A robot solution to EOD operations", Defense and Security Studies, vol. 3, pp. 22–31, Jun. 2022.

X. J. Handl, J. Knowles, and M. Dorigo, "Ant-based clustering and topographic mapping," *Artif. Life*, vol. 12, no. 1, pp. 35–62, 2006.

XI. J. K. Ali and Y. S. Miz'el, "A new miniature Peano fractal-based bandpass filter design with 2nd harmonic suppression," in *2009 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, 2009.

XII. K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2191–2233, 2019, doi: 10.1007/s10462-017-9605-z.

XIII. K. A. Santoso, M. B. Kurniawan, A. Kamsyakawuni, and A. Riski, "Hybrid Cat-Particle Swarm Optimization Algorithm on Bounded Knapsack Problem with Multiple Constraints," *Proc. Int. Conf. Math. Geom. Stat. Comput. (IC-MaGeStiC 2021)*, vol. 96, pp. 244–248, 2022, doi: 10.2991/acsr.k.220202.045.

XIV. O. Edin and S. Manfi, "Semi-parametric regression function estimation for environmental pollution with measurement error using artificial flower pollination algorithm," vol. 13, no. 1, pp. 1375–1389, 2022.

XV. P. Stodola, K. Michenka, J. Nohel, and M. Rybansk\`y, "Hybrid algorithm based on ant colony optimization and simulated annealing applied to the dynamic traveling salesman problem," *Entropy*, vol. 22, no. 8, p. 884, 2020.

XVI. S. M. Redha, A. Abdullatif, and I. A. Hussain, "Using Particle Swarm Optimization Algorithm to Address the Multicollinearity Problem," *ARPN J. Eng. Appl. Sci.*, vol. 14, no. 10, pp. 3345–3353, 2019, doi: 10.36478/JEASCI.2019.3345.3353.

XVII. S. M. Redha and A. T. A. Hadia, "Employment of the genetic algorithm in some methods of estimating survival function with application," *Period. Eng. Nat. Sci.*, vol. 8, no. 1, pp. 481–490, 2020, doi: 10.21533/pen.v8i1.1161.g530.

XVIII. S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7-11, 2006 Proceedings 9*, 2006, pp. 854–858.

XIX. S. Ahmadov, "Enhancing BYOD mobile device security in a hybrid environment", Sustainable Engineering and Innovation, vol. 5, no. 2, pp. 247-260, 2023.

XX. S. Kubura, S. T. Imeci, and A. F. Uslu, "Rectangular small patch antenna", Sustainable Engineering and Innovation, vol. 6, no. 1, pp. 17-24, 2024.

XXI. S. Sarac and B. Durakovic, "Analysis of student performances in online and face-to-face learning: A case study from a Bosnian public university", Heritage and Sustainable Development, vol. 4, no. 2, pp. 87–94, Dec. 2022.

XXII. S. Feng, C. Cheng, and L. Mo, "An Improved Cuckoo Search Algorithm and Its Application in Function Optimization," *Commun. Comput. Inf. Sci.*, vol. 1768 CCIS, no. 5, pp. 439–455, 2023, doi: 10.1007/978-981-99-0272-9_30.

XXIII. T. S. L. V Ayyarao *et al.*, "War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization," *IEEE Access*, vol. 10, pp. 25073–25105, 2022.

XXIV. T. O. Ting, X. S. Yang, S. Cheng, and K. Huang, "Hybrid metaheuristic algorithms: Past, present, and future," *Stud. Comput. Intell.*, vol. 585, no. December, pp. 71–83, 2015, doi: 10.1007/978-3-319-13826-8_4.

XXV. U. N. I. V Er, S. Of, K. Library, and P. O. Rfof, "U N IV ER SITY OF NAIROBI KIKUYU LIBRARY P O RfOf 97 KIKUYU," 2012.

XXVI. X. S. Yang, S. Deb, Y. X. Zhao, S. Fong, and X. He, "Swarm intelligence: past, present and future," *Soft Comput.*, vol. 22, no. 18, pp. 5923–5933, 2018, doi: 10.1007/s00500-017-2810-5.

XXVII. X.-S. Yang, “Flower pollination algorithm for global optimization,” in *International conference on unconventional computing and natural computation*, 2012, pp. 240–249.

XXVIII. X. Yang, “Music-Inspired Harmony Search Algorithm,” *Music. Harmon. Search Algorithm*, no. May, 2009, doi: 10.1007/978-3-642-00185-7.

XXIX. Y. Ding, L. Chen, and K. Hao, *Bio-inspired optimization algorithms*, vol. 118. 2018. doi: 10.1007/978-981-10-6689-4_8.

XXX. Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, “Hunger games search Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts,” *Expert Syst. Appl.*, vol. 177, p. 114864, 2021.

XXXI. Y. S. Mezaal and H. T. Eyyuboglu, “A new narrow band dual-mode microstrip slotted patch bandpass filter design based on fractal geometry,” in *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*, IEEE, 2012, pp. 1180–1184.

XXXII. Y. S. Mezaal and S. F. Abdulkareem, “New microstrip antenna based on quasi-fractal geometry for recent wireless systems,” in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 2018.

XXXIII. Y. S. Mezaal, H. H. Saleh, and H. Al-saedi, “New compact microstrip filters based on quasi fractal resonator,” *Adv. Electromagn.*, vol. 7, no. 4, pp. 93–102, 2018.

XXXIV. Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: harmony search,” *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.