

REVIEW

Open Access



# Hunter–Prey Optimization Algorithm: a review

Pappu Soundarya Lahari<sup>1\*</sup> and Varaprasad Janamala<sup>1</sup>

\*Correspondence:  
pappusoundarya.lahari@res.  
christuniversity.in

<sup>1</sup>Electrical and Electronics  
Engineering Department, School  
of Engineering and Technology,  
Christ Deemed to Be  
University, Kengeri Campus,  
Bangalore 560074, India

## Abstract

The Hunter–Prey Optimization Algorithm (HPO) is a nature-inspired optimization technique influenced by the predator–prey relationships observed in nature. Over the years, HPO has gained attention as a promising method for solving complex optimization problems. This review article provides a comprehensive analysis and a bibliographic study of the Hunter–Prey Optimization Algorithm. It explores its origins, underlying principles, applications, strengths, weaknesses, and recent developments in detail. By delving into various facets of HPO, this review aims to shed light on its effectiveness and potential, inspiring the researchers to address real-world optimization challenges.

**Keywords:** Optimization techniques, Hunter–Prey Optimization, Improved Hunter–Prey Optimization

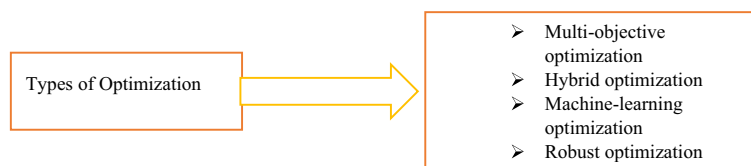
## Introduction

Low-emission technologies like distribution generation and electric vehicles have increased adoption in the distribution networks. These technologies are widely promoted to meet the growing power demand and economic and environmental factors. Various control schemes have been embraced in the active distribution networks to overcome the drawback of power quality issues from these penetrations and determine the suitable site and size of penetration. Mathematical optimization is the observed and broadly implemented key technology in validating control scheme technologies. Optimization has also been continuing in the limelight and is most focused on by researchers in various fields of engineering problems. Rapid depletion of existing sources and profit maximization could be the reasons for the same [1]. This work highlights the role of one such optimization algorithm, the Hunter–Prey Optimization Algorithm (HPO), in addressing the challenges posed by these low-emission technologies.

## Background classification of optimization methods

The types of optimization are studied under four types, and some of the developed optimization techniques to address the earlier discussed issues [2], Fig. 1 depicting the same, are as follows:

- *Multi-objective optimization:* This approach simultaneously considers two or more divergent objectives, such as power loss reduction and voltage stability



**Fig. 1** Types of optimization methods

enhancement. Multi-objective optimization can yield a series of optimal solutions that constitute a deal between the individual objectives, letting decision-makers opt for the solution that aptly fits their needs.

- *Hybrid optimization:* This technique involves combinations of various meta-heuristic algorithms to overcome individual deficiencies and improve the algorithm's strength, convergence speed, and accuracy.
- *Machine learning optimization:* This technique is widely used for large-scale systems to extract data and predict a much more accurate optimal solution at a faster rate.
- *Robust optimization:* This technique considers the sensitivity of system parameters. It deals with uncertainties in the optimization problem like renewable energy introduction, sudden load demand, etc.

On the other hand, optimization methods can be broadly classified under the categories of classical optimization techniques, sensitive index methods, meta-heuristic methods, and mixed techniques [2]. Classic optimization techniques involve the traditional methods of mathematics, namely Newton–Raphson (NR) [3], Linear Programming (LP) [4], Non-Linear Programming (NLP) [5], Mixed-Integer Programming (MIP) [6], Mixed-Integer Linear Programming (MILP) [7], Mixed-Integer Non-Linear Programming (MINLP) [8, 9], Dynamic Programming (DP) [7].

Sensitive index methods involve the calculation of sensitivity index factors for optimization solutions. LFI (Loss Factor Index), PLI (Power Loss Index), LSI (Loss Sensitivity Index), and voltage indices like VSI (Voltage Stability Index) and AVDI (Average Voltage Deviation Index) are a few of them that aid in finding an optimization solution considering the sensitivity parameters.

Mixed methods combine traditional mathematical and meta-heuristic methods with a sensitivity indices approach for a better solution than individual performances. Hybrid works of PSO-GSA [10], GA-GSA [11], fuzzy-GA [12], and fuzzy-DE [13] are a few of them.

The meta-heuristic methods of optimization approach can be classified under the categories of:

1. Biology-based optimization
2. Physics-based optimization
3. Geography-based optimization
4. Other population-based optimization

### **Biology-based optimization**

These types of methods, which are inspired by biological and natural activities [14], can be further classified into:

- a. Evolutionary algorithms
- b. Swarm-based algorithms

#### ***(a) Evolutionary algorithms***

A global adaptive search optimization algorithm named Genetic Algorithm (GA) based on natural selection [15], another technique called Evolutionary Programming (EP) put forward by D. B. Fogel in 1990 [16, 17], Evolution Strategy (ES) inspired by the biological principles of evolution [18], developed by Storny et al. [19], Differential Evolutionary (DE) algorithm, a widely accepted algorithm to handle nonlinear, multimodal, non-differentiable cost functions, swiftly [20] come under evolutionary algorithms. GA employs binary coding to represent the problem parameters, while DE uses float point, making it more accurate with more adaptive and control parameters [21, 22].

#### ***(b) Swarm-based algorithms***

Swarm-based algorithms are highly emerging nature-inspired techniques that mimic the interactional behavior of large, homogenous species among themselves and their environment. Bird flocks, ant colonies, fish shoals, and honeybees form a few agents of these swarm-based algorithms. Particle-Swarm Optimization (PSO), proposed by Kennedy et al. [23], inspired by flocks of birds, is a broadly accepted swarm technique. The Bacteria Foraging Optimization Algorithm (BFO), developed by Passino [24], emulates *Escherichia coli* bacteria's foraging behavior, the Cuckoo Search Algorithm (CS) by Yang et al. [25] mimics the breeding practice of cuckoo birds, Ant Colony Optimization (ACO) rested on the foraging behavior and pheromones-based communication of ants, Firefly Algorithm (FA) is motivated by the flashing behavior of the fireflies and developed by Yang [26], the home search strategy of pigeons inspired the Pigeon-Inspired Optimization (PIO) algorithm and its improved version, proposed by Duan et al. [27, 28], are various swarm-based algorithms. Other swarm-based optimizations include the Coral Reef Optimization algorithm (CRO) by Salcedo-Sanz et al. [29], an Artificial Immune System (AIS) based on clone generation and maturation [30], Whale Optimization Algorithm (WOA) [31], Cat Swarm Optimization (CaSO) [32], Crow Search Algorithm (CSA) [33], Moth-Flame Optimization Algorithm (MFO) [34], Grey Wolf Optimizer (GWO) [35], Flower Pollination Algorithm (FPA) [36], Honey Bee Mating Optimization (HBMO) [37, 38], symbiotic organism search algorithm, Butterfly Optimization Algorithm (BOA) [39, 40], Camel Search Algorithm (CA) [41], Bat Algorithm (BA) [42], and Grasshopper Optimization Algorithm (GOA) [43] sum up the group.

#### ***Physics-based optimization algorithms***

These optimization algorithms are inspired by the laws of physics, the physical behavior of matter, or its physical properties. The Gravitational Search Algorithm

(GSA), rooted in the gravitational law and physics laws of motion by Rashedi et al. [44], Simulated Annealing (SA) follows the physical process of annealing that is used for crystallization [45, 46], Magnetic Optimization Algorithm (MOA) based on the attraction and repulsion principle of magnets by Tayarani et al. [47], Intelligent Water Drop (IWD) algorithm inspired from the river flow introduced by Hosseini [48] can be classified under the physics-based algorithms. Multiverse Optimization (MVO) [49], Atom Search Optimization (ASO) algorithm [50], Curved Space Optimization (CuSO) [51], Galaxy-Based Search Algorithm (GBSA) [52], Water Cycle Algorithm (WCA) [53], Black Hole (BH) algorithm [54], Harmony Search (HS) Algorithm [55] add up to the list.

#### ***Geography-based algorithms***

The Imperialistic Competition Algorithm (ICA), pioneered by Gargari et al. [56], involves countries' colonies and imperialists as population; Tabu Search (TS) following a search-escape pattern put forward by Glover [57] comes under the geography-based algorithms.

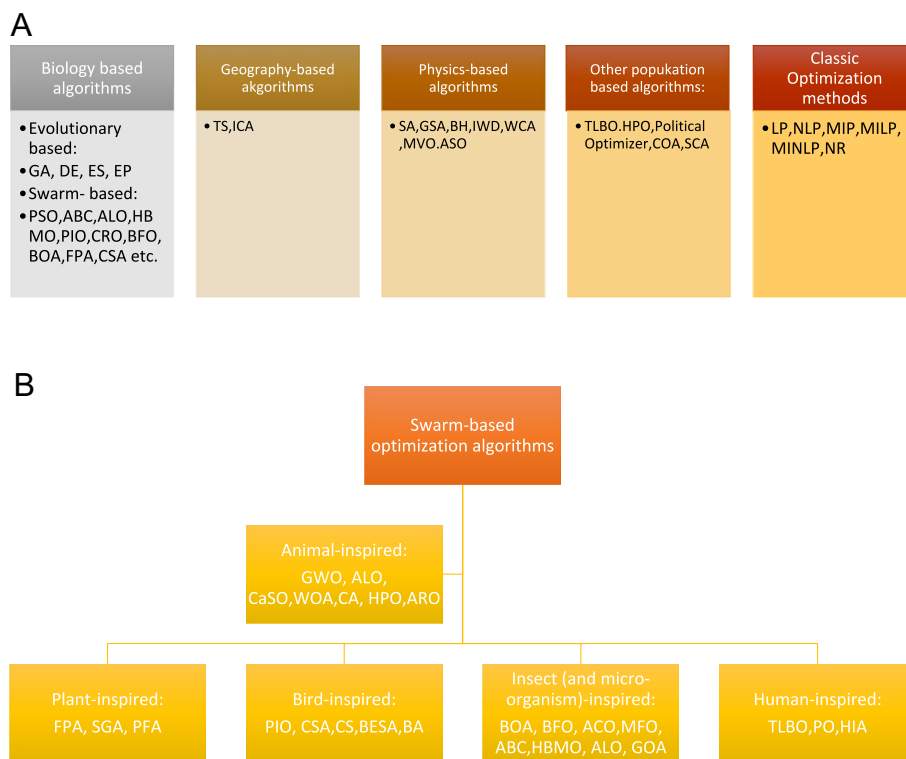
#### ***Other population-based algorithms***

Sine Cosine Algorithm (SCA) [58], Parallel Seeker Optimization Algorithm (PSOA) [59], Artificial Rabbit Optimization (ARO) algorithm [32], Hunter–Prey Optimization (HPO) [60], Teaching–Learning-Based Optimization Algorithm (TLBO) [61], Bald Eagle Search Algorithm (BESA) [62], Chaotic Optimization Algorithm (COA) [63], political optimizer [64], Paddy Fields Algorithm (PFA) [65], Saplings Growth Algorithm (SGA) [66], Human-Inspired Algorithm (HIA) [67] are the other population-based optimization algorithms.

Figure 2a is on the classification of algorithms and 2b gives another classification based on the nature of inspiration. The algorithms can also be classified based on their inspiration. Since the nature-inspired optimization algorithms are inspired by a group or flock of animals' behavior, they can conveniently be classified into a) animal inspired, b) bird inspired, c) insect inspired, d) plant inspired, and e) human inspired.

#### **Literature on Hunter–Prey Optimization**

HPO is a nature-inspired population-based optimization algorithm by Naurei et al. [60] 2022 to address optimization problems in different engineering fields. Many researchers have employed the algorithm to solve various issues, as in [68], the authors applied the HPO algorithm for the optimal positioning of PV-STATCOM with energy loss depreciation and improving the voltage profile. Active power loss minimization, greenhouse gas emission reduction, and improving the hosting capacity of PV and the voltage profile are the objectives in [69] applying the HPO algorithm. Article [70] studies the algorithm for optimal PV placement with actual power loss reduction and voltage profile enhancement. A combined algorithm of HPO-HDL is put forward in [71] for fake news detection; HDL stands for Hybrid Deep Learning, an AI technique. HPO is used to identify the parameters of solar PV cells of R.T.C. France and STM-6/120 models [72]. A tabularized representation of the same is shown in Table 1.



**Fig. 2** a Classification of optimization algorithms. b Classification based on mode of inspiration

There are articles on improvised versions of the standard HPO (IHPO) [73] defined an enhanced HPO algorithm for optimal FACTS and wind DG placement with power loss reduction, cost reduction, and voltage enhancement as objectives. IHPO with extreme machine learning to predict the wind power output and accuracy is studied in [74]. A hybrid combination of IHPO and Convolution Neural Networks (CNN) to speculate the structural damages in buildings and construction is proposed in [75]. [76] uses the IHPO to plan a robot path finding algorithm in unknown surroundings. A table representing the discussed works is depicted in Table 2.

**HPO motivation**

HPO’s elegance lies in its relative simplicity. It uses a small set of intuitive rules to traverse and exploit the search space effectively, making it computationally efficient and potentially applicable to many optimization problems. Like other algorithms, HPO also has exploration and exploitation phases after initialization, and there is a difference in effectively balancing the exploration and exploitation phases. On the other hand, reactive power can be optimally dispatched by finding the prime locations of the desired devices using the proposed algorithm. Hence, applying the newly emerging algorithm in the engineering field to handle the optimization problem motivates this article.

**Research gap and challenges**

Several optimization algorithms have been proposed recently, and many are successfully in use, while many are in the developing and testing stages. At this point, developing a new

**Table 1** Works on Hunter–Prey Optimization (HPO)

Ref	Algorithm	Objectives	Observations made	Future scope
[68]	HPO	Lessen the electrical energy losses	HPO surpasses the Differential Evolutionary (DE) method, Artificial Rabbits Algorithm (ARA), Particle Swarm Optimization (PSO), and golden search optimizer (GSO) in allocating PV-STATCOM devices in distribution systems in convergence as well as loss minimization and voltage enhancement	The work can be extended with a network reconfiguration, introducing custom power devices, a hybrid DG technology, or EVCS to highlight the proposed HPO algorithm
[69]	HPO	Boost up the voltage profile To minimize the active power losses	The obtained results from HPOA are compared with the Teacher–Learner-Based Optimization (TLBO), Butterfly Optimization Algorithm (BOA), Bald Eagle Search (BES), and Cuckoo Optimization Algorithm (COA) The proposed algorithm shows better results in improving HC of PV, loss reduction, and voltage improvement	A combined study with PV and WT can be an expected future work. EVs and FACTS integration can also pave the way for the following study
[70]	HPO	To reduce the GHC emissions To improve the voltage profile To improve the Hosting Capacity (HC) of the PV array Optimal PV placement	The supremacy of the HPO algorithm is highlighted by comparing it with other existing algorithms for active power loss reduction	Simultaneous EV and PV optimal placement using the HPO
[71]	HPO+HDL	Active loss reduction Voltage improvement Fake news detection	The combination of HPO and hybrid deep learning (HDL) is presented to detect fake news	Combination with other AI techniques for improved performance
[72]	HPO	Parameters of solar PV panel models of STM-6/120 and P.T.C. France are extracted using the HPO for single, double, and triple diode models	A better convergence and parameter values are obtained using the HPO	It applies to other optimizations in power systems, such as optimal power flow, synchronous motor designs, AC–DC power system operation, combined heat, and power dispatch

**Table 2** Works on improved Hunter–Prey Optimization (IHPO)

Ref	Algorithm	Objectives	Observations	Future scope
[73]	EHPO	Optimal power flow  Loss minimization Cost reduction Voltage deviation minimization and voltage enhancement	The efficiency of the preferred EHPO is apt to solve real-world complex power system problems	EV and PV placement, large test-systems
[74]	IHPO	Wind output prediction  Increase the efficiency of the predicted output	Improved HPO can boost the convergence speed and search efficiency of the HPO	It can be studied with PV and a hybrid DG model
[75]	IHPO & CNN	To identify the structural damages quickly and accurately	There is a two-stage approach, with CNN at stage one to identify the damage and IHPO at the second stage to estimate its severity	The same approach is used in two stages to study the electrical distribution networks with DG integration
[76]	IHPO	Path designing of a mobile robot in an unknown environment, refraining from obstacles	The IHPO is used along with an obstacle-detection strategy and a search strategy, and it is found to be more effective	Application to a real-time robot is the paper's proposed future work

algorithm to showcase its supremacy could be challenging. However, as an answer to the question of ‘What is the need for a new algorithm?’ the NFL theory has been put forward by [77]. According to the No Free Launch (NFL), no one can present an algorithm that can solve all problems of optimization, and researchers are allowed to suggest new optimization algorithms or improve the existing techniques to solve a subset of problems in various fields. The new algorithm HPO can handle optimization problems in different engineering and non-engineering fields.

#### Contribution of the article

- This article reviews the HPO algorithm, demonstrating its working phases: initialization, exploration, and exploitation, along with the parameters deciding the algorithm.
- The paper also details the newer versions of the algorithm, explaining the improvements made to the standard algorithm.
- The application of HPO in electrical engineering for optimal DG and capacitor bank placement is showcased. Overall, the paper briefs the new nature-inspired HPO algorithm, its variants, and applications.

Further, the paper is categorized, detailing the 2. Standard HPO, 3. Improved versions of HPO, 4. Discussions, and 5. Conclusions.

## Standard Hunter–Prey Optimization Algorithm (HPO)

### Inspiration

Hunter–Prey Optimization (HPO) takes a captivating approach to problem-solving, drawing inspiration from the dynamic world of predator–prey interactions. It mimics predators' strategies to hunt and capture their prey. The scenario of a hunter searching for prey, and since prey is usually grouped, the hunter chooses prey far from the flock (average herd position). After the hunter finds his prey, he chases and hunts it. At the same time, the prey searches for food, escapes the predator's attack, and reaches a safe place, which is the fitness function. HPO is a class of swarm intelligence algorithms and falls under the broader category of meta-heuristic algorithms used for optimization problems.

### Mathematical model of the algorithm

Naruei et al. [60] proposed HPO, a new intelligent optimization algorithm with fast convergence and a higher optimization potentiality. The general structure of any algorithm begins with the population in initialization,  $(\vec{x}) = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$ . For every member, the objective function is computed. The positions of the hunter and the prey are updated at every iteration, evaluating the objective function till the algorithm stops. The initial position of the member is given by Eq. (1) from [60],

$$x_i = \text{rand}(1, d) * (u - l) + l \quad (1)$$

where  $x_i$  is the initial position of hunter or prey,  $\text{rand}(1, d)$  is any random number between  $[0, 1]$ ,  $u$  and  $l$  are the upper and lower boundaries, and  $d$  is the dimension of the problem. The objective function is then evaluated.  $OF = f(\vec{x})$ . The exploration and exploitation are the following stages after initialization. These stages involve a search mechanism that pilots the search agents toward the optimal solution. Equation (2) defines the hunter's position as

$$x_{i,j}(t+1) = x_{i,j}(t) + 0.5 \left[ (2CZP_{\text{pos}} - x_{i,j}(t)) + (2(1-C)Z\mu_i - x_{i,j}(t)) \right] \quad (2)$$

where  $x_{i,j}(t)$  defines the current position of the hunter and  $x_{i,j}(t+1)$  for the next iteration.  $P_{\text{pos}}$  represents the position of the prey, and  $C$  and  $Z$  are the balance and adaptive parameters, respectively.  $\mu$  is the mean of the locations and is evaluated using Eq. (3)

$$\mu = \frac{1}{n} \sum_{i=1}^n \vec{x}_i \quad (3)$$

$n$  is the number of iterations, and  $x_i$  gives the position at the iteration  $t$ .

For a random vector  $P$  with 0 and 1 values,

$$P = \vec{R}_1 < C, \text{IDX} = (P == 0) \quad (4)$$

For  $R_1$  being a random vector,  $\text{IDX}$  defines the index value of the  $R_1$  vector at  $P = 0$ . The adaptive parameter  $Z$  can be evaluated from Eq. (5) with  $R_2$  and  $R_3$  as random vectors between  $[0, 1]$ ;  $\otimes$  denotes the element-wise multiplication.



$$Z = \vec{R}_2 \otimes IDX + \vec{R}_3 \otimes (\sim IDX) \quad (5)$$

The balance parameter between the stages of exploration and exploitation,  $C$ , can be obtained from Eq. (6)

$$C = 1 - \text{itr} \left( \frac{0.98}{\text{itr}_{\max}} \right) \quad (6)$$

Itr is the current iteration, and  $\text{itr}_{\max}$  is the maximum iteration count. During iterations, the value of  $C$  decreases from 1 to 0.02.

The prey's position  $P_{\text{pos}}$  is calculated using the average value  $\mu$  from Eq. (3) and the Euclidean distance obtained from Eq. (7).

$$D_{Eu(i)} = \sqrt[2]{\sum_{j=1}^d (x_{i,j} - \mu_j)^2} \quad (7)$$

From Eq. (8), the member at the maximum distance is considered prey.

$$P_{\text{pos}} = \vec{x}_i \mid i \text{ is the index of Max(end)sort}(D_{Eu}) \quad (8)$$

The hunter easily mocks down the animal away from the group, then goes for its next, and this continues. Also, there would be a late convergence if the search agent is considered at a longer distance from the mean every time. So, to avoid this situation, Eq. (9) is defined for  $N$  number of search agents.

$$k\text{best} = \text{round}(C \times N) \quad (9)$$

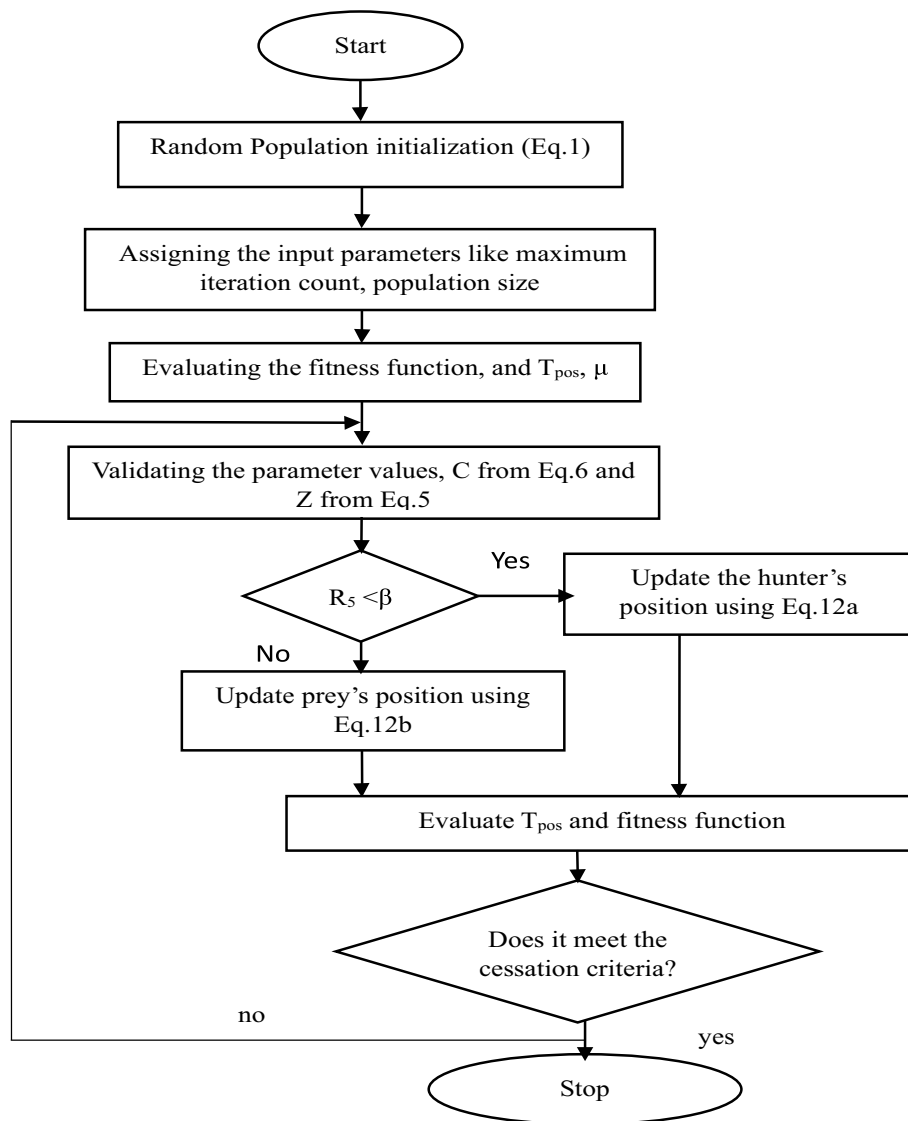
Now, the new prey position can be given by

$$P_{\text{pos}} = \vec{x}_i \mid i \text{ is sorted } D_{Eu}(k\text{best}) \quad (10)$$

The  $k\text{best}$  value is  $N$  at the start but gradually decreases with iterations and gets its value equal to the first member. This is because the hunter chooses the prey at a farther distance each time, and thus, the  $k\text{best}$  values decrease at each iteration. In the hunting scenario, the prey tries to escape from the attacker and reach its herd; hence, it can be said that the safe position of the prey is the optimal solution. The prey escape phase equation can be given as

$$x_{i,j}(t+1) = T_{\text{pos}(j)} + CZ \cos(2\pi R_4) \times (T_{\text{pos}(j)} - x_{i,j}(t)) \quad (11)$$

Here, in Eq. (11),  $x_{i,j}(t)$  and  $x_{i,j}(t+1)$  are the current and the next prey's positions.  $Z$  and  $C$  are calculated from Eqs. 5 and 6, respectively.  $R_4$  is the random vector between  $[-1,1]$ , and  $T_{\text{pos}}$  is the global optimum position. The significance of the  $\cos$  function is that it decides the next prey's location from the global maximum using the input parameters at various radii and angles. To distinguish the hunter and prey mathematically, by combining equations 2 and 11 and defining another random vector  $R_5$  ranging in  $[0,1]$ , a regulating parameter  $\beta = 0.1$ ,



**Fig. 3** Flow chart for HPO

$$x_{i,j}(t+1) = \begin{cases} x_{i,j}(t) + 0.5 \left[ (2CZP_{\text{pos}} - x_{i,j}(t)) + (2(1-C)Z\mu_{(j)} - x_{i,j}(t)) \right] & 12a \\ T_{\text{pos}(j)} + CZ \cos(2\pi R_4) \times (T_{\text{pos}(j)} - x_{i,j}(t)) & 12b \end{cases} \quad (12)$$

Equation 12a gives the next position of the hunter for  $R_5 < \beta$ , and Eq. 12b updates the prey's position for  $R_5 > \beta$ . The flow chart of HPO is shown in Fig. 3.

The following are the suppositions made by the author [60] for the HPO algorithm to provide appropriate solutions:

- Random selection of hunter and prey assures the search for space exploration. Also, there will be a low probability of getting stuck in the local maximum.

- Search space exploration by selecting the farthest member as prey and the mechanism of the mean distance reduction with every iteration ensure the convergence of the algorithm and its exploitation.
- The adaptive parameter escalates the population divergence, reduces the severity of the hunter and prey position, and guarantees the algorithm's convergence.
- The adjustment parameters of the HPO algorithm are smaller in number and are non-gradient algorithms.

### Parameters of HPO

The parameters of HPO can be identified as population-related and movement-related parameters. Population size and maximum iterations ( $itr_{max}$ ) are population-based parameters, and regulating parameter  $\beta$  and balance parameter  $C$  are the movement-related parameters. The fitness function can be regarded as the other parameter of HPO. Table 3 summarizes the parameters of a few referred algorithms:

### Complexity analysis of HPO

The complexity of an optimization algorithm refers to the computational resources it requires to solve a problem. There are two main aspects to consider:

**Table 3** Parameters of various algorithms

Algorithm	Parameters	Values
ALO [78]	$r$ (walking variable)	[0,2]
BOA [39]	$a$ (power exponent of modality)	0.1
	$c$ (sensory modality)	0.01
	$p$ (switching probability)	0.6
GWO [35]	$a$ (internal parameter)	[0,2]
HHO [79]	$\beta$ (probability parameter)	[0,1] Usually 0.5
	$E$ (Energy parameter)	[-1,1]
HPO [60]	$C$ (balance parameter)	[1-0.02]
	$B$ (adjusting parameter)	0.1
MPA [73]	$a$ (power exponent of modality)	0.1
	$c$ (sensory modality)	0.01
	$p$ (switching probability)	0.6
PSO [23]	$C1, C2$ (social and cognitive coefficients)	2
	$\omega_1$ and $\omega_2$ (inertia function)	[0.9-0.2]
SCA [58]	$a$ (constant)	2
	$r$ (balance parameter)	2
TS [57]	$T$ (Tabu memory parameter)	[0,4] (minimum and maximum values)
WOA [31]	$a$ (internal parameter)	[0,2]

**Time complexity**

This measures the time it takes for the algorithm to execute, typically expressed in terms of the number of basic operations it performs. Here are some common factors influencing time complexity:

- Population size (N): Many optimization algorithms work with a population of solutions. The complexity often increases as the population grows, as the algorithm must evaluate and update each solution.
- Number of iterations (T): Most algorithms iterate through a loop, refining their solutions. The complexity increases with the number of iterations required for convergence.
- Problem dimensionality (D) refers to the number of optimized variables. For some algorithms, the complexity increases with the number of variables as the search space increases.
- Function complexity: The complexity of the objective function (what the algorithm is trying to optimize) can also play a role. Evaluating a more complex function in each iteration can increase the overall complexity of the time.

**Space Complexity**

This refers to the amount of memory required by the algorithm to run. Here's what typically contributes to space complexity:

- Storing solutions: The algorithm needs to store information about each solution in the population, including its position in the search space. This memory usage scales with the population size (N) and problem dimension (D).
- Additional data structures: Some algorithms might use additional data structures like sorting mechanisms or temporary variables. These can contribute to the overall complexity of space, but their impact is usually smaller than that of storing solutions.

The space complexity is almost similarly computed for most algorithms, such as  $O(N \times D)$ , where N is the population size and D is the dimension.

$O(N)$ , the most commonly used notation of complexity, denotes linear complexity, meaning the execution time grows linearly with the population size or number of iterations.

Typically, the complexity of the HPO algorithm depends on four components: initialization, updating of the hunter, updating prey, and fitness evaluation. Note that the initialization process's computational complexity with N search agents is  $O(N)$ . The computational complexity of the updated process can be given as  $O(T \times N) + O((1 - \beta) \times T \times N \times D) + O(\beta \times T \times N \times D)$ , T denotes the maximum number of iterations (earlier referred to as  $iter_{max}$ ), D denotes the number of problem variables, and  $\beta$  is a regulatory parameter with a value of 0.1 [60]. Hence, the total complexity is  $O(N \times (T + (1 - \beta)TD + \beta TN + 1))$ .

The complexity of PSO is  $O(D \times N \times iter_{max})$ . The computation time grows linearly with the problem dimensionality D, swarm size N, and maximum iterations  $iter_{max}$ .

The complexity of the standard GWO algorithm is  $O(N * d * Tmax)$ ,  $N$  is the size,  $d$  is the population, and  $Tmax$  is the maximum number of iterations. ALO's time complexity is generally considered to be  $O(D * N * T)$ , where  $D$  represents the dimensionality of the problem (number of variables being optimized),  $N$  represents the population size (number of antlions), and  $T$  represents the maximum number of iterations. The total time complexity of WOA is generally considered to be  $O(M * N * D + f(N))$ , where  $M$  represents the maximum number of iterations,  $N$  represents the population size,  $D$  represents the number of dimensions in the problem, and  $f(N)$  represents the time required to evaluate the fitness function for  $N$  individuals. The complexity of HHO is generally considered to be  $O(N(D + T))$ ,  $T$  is the maximum iteration count,  $D$  is the dimension, and  $N$  is the population.  $O(N*T)$  gives the complexity of SCA. Here,  $T$  represents the total number of iterations the algorithm runs for and is the population size. Compared to HHO, SCA avoids the additional complexity of interactions between solutions (like hunting behavior), making it slightly more efficient. In general, Tabu Search is considered not to have a polynomial time complexity as determining a specific time complexity for Tabu Search is observed to be complicated.

### Improved versions of the Hunter–Prey Optimization Algorithm

This section discusses various recent works with improvised versions of the HPO algorithm in varied fields of engineering.

#### Improvising regulating parameter

[76] proposes Improved HPO for robot path planning with an upgraded adjusting (regulating) parameter  $\beta$  and introducing a new parameter called changing parameter (CP). The new  $\beta$  is given as

$$\beta = 2 \times \text{rand} - 1 \quad (13)$$

The changing parameter addresses the absence of a transfer parameter in the standard HPO. It increases the exploration speed and thus a faster pace for exploitation. Equation (14) gives the CP value and is followed by the refined position-defining equations.

$$CP = \sin\left(C - \frac{t}{T}\right) \quad (14)$$

$$x_{ij}(t+1) = \begin{cases} x_{ij}(t) + 0.5[(\alpha 2CZP_{\text{pos}} - x_{ij}(t)) + 2(1-C)Z\mu_j - x_{ij}(t)] & 15a \\ T_{\text{pos}(j)} + \alpha CZ \cos(2\pi R_4) \times (T_{\text{pos}(j)} - x_{ij}(t)) & 15b \end{cases} \quad (15)$$

where  $\alpha = \beta \times CP$ .

In this version of IHPO, the author upgrades the regulatory parameter for randomization and prevents early convergence, which also helps alter the search direction. The proposed algorithm is juxtaposed with other contemporary algorithms of Particle Swarm Optimization (PSO), Salp Swarm Algorithm (SSA), Fitness-Dependent Optimizer (FDO), conventional COOT and HPO algorithms for 13 benchmark criteria, and 30-dimension functions that are widely used by the researchers. The IHPO performs its best when compared with other algorithms. The work uses the nature-inspired

algorithm and local search and block detection strategies for robot path planning in an unrecognized environment. The author gives a future scope for real Turtlebot robot testing. Also, there is a reach to implement the algorithm in the electrical distribution networks with renewable source integration.

#### Improvising initialization phase

[75] presents a developed version of HPO, which will be upgraded at the initialization stage. Instead of random initialization in the conventional method, this work uses tent chaotic mapping for initialization and Cauchy distribution for random variables to clear the periodic points.

$$y_{i+1} = \begin{cases} \eta y_i, & 0 \leq y_i \leq 0.5 \\ \eta(1 - y_i), & 0.5 \leq y_i \leq 1 \end{cases} \quad (16)$$

$$y_{i+1} = \begin{cases} \mu y_i + \text{cauchy}(0, 1) \times \frac{1}{N}, & 0 \leq y_i \leq 0.5 \\ \mu(1 - y_i) + \text{cauchy}(0, 1) \times \frac{1}{N}, & 0.5 \leq y_i \leq 1 \end{cases} \quad (17)$$

Equations (16) and (17) give the tent mapping and Cauchy's distribution with  $N=2$ , called the chaotic parameter. Now, initialization Eq. (1) is modified to

$$x_i = y_i \times (u - l) + l \quad (18)$$

The author also suggests a linear combination of prey position, global optimum, and mean position, updating Eq. (12) as

$$x_{i,j}(t+1) = \begin{cases} x_{i,j}(t) + 0.5 \left[ \left( 2CZ \left( \frac{P_{\text{pos}(j)} - T_{\text{pos}(j)}}{2} \right) - x_{i,j}(t) \right) + 2(1 - C)Z \left( \frac{\mu_j - T_{\text{pos}(j)}}{2} \right) - x_{i,j}(t) \right] 19a \\ T_{\text{pos}(j)} + CZ \cos(2\pi R_4) \times \left( \left( \frac{g^{\text{best}(j)} - T_{\text{pos}(j)}}{2} \right) - x_{i,j}(t) \right) 19b \end{cases} \quad (19)$$

The performance of the IHPO is compared to Differential Evolution (DE), Cuckoo Search Algorithm (CSA), Particle Swarm Optimization (PSO), Gray Wolf Optimizer (GWO), and Moth-Flame Optimization (MFO), and the convergence speed of the improved HPO is much faster than the rest. It also surpasses the rest of the algorithms' convergence efficiency, accuracy, and optimization ability. In the referred article, the algorithm and Convolutional Neural Network (CNN) are implemented to identify the structural damages in buildings and constructions.

[74] Also, the standard HPO can be advanced to IHPO by upgrading the initialization phase. To increase population diversity, the stochastic reverse learning technique has been introduced as

$$X_{\text{rand}} = l + u - r \cdot X \quad (19)$$

$l$  and  $u$  are the lower and upper boundaries,  $r$  is an arbitrary value between  $[0,1]$ ,  $X$  belongs to  $[l,u]$ , and  $X_{\text{rand}}$  gives the random reverse solution. Stochastic reverse learning focuses on producing a random inverse response from the present iteration during the population search, collating the objective function values within the two solutions, and then picking out the prime solution to proceed to the next iteration.

Also, the paper introduces a weighing function,  $\omega$ , to improvise the prey’s position equation; Eq. (11) is updated as Eq. (20).

$$x_{i,j}(t + 1) = \omega \cdot T_{\text{pos}(j)} + CZ\cos(2\pi R_4) \times (T_{\text{pos}(j)} - x_{i,j}(t)) \tag{20}$$

$$\omega = \omega_{\min} \left( \frac{\omega_{\max}}{\omega_{\min}} \right)^{\frac{1}{\text{itr}_{\max}}} \tag{21}$$

$\text{itr}_{\max}$  gives the maximum number of iterations,  $t$  is the current iteration,  $c$  is the adjustment parameter, and  $\omega_{\min}$  and  $\omega_{\max}$  are the weight regulating parameters. It can be observed that a similar parameter has been introduced in earlier referred work [74] in improvising the prey’s position equation. The IHPO is compounded with Extreme Machine Learning (ELM) to estimate wind power and is found effective, providing scope to improve wind power prediction accuracy. The scope can be further extended for a hybrid DG placement.

**Updating the step size**

The step size is upgraded in two stages, high-velocity and low-velocity ratios inspired by the Marine Predator Algorithm (MPA) [73, 80], to avoid the optimum values trapped in local maxima. The first stage is defined for a higher step size, reflecting Brownian motion mathematically:

$$it < \frac{1}{3} \text{maxit} \tag{22}$$

$$S = \vec{R}_B \otimes (E - \vec{R}_B \otimes x_{i,j}(t)) \tag{23}$$

$$x_{i,j}(t + 1) = x_{i,j}(t) + P \cdot \vec{R}_B \otimes S \tag{24}$$

For  $P=0.5$ ,  $R_b$  is the vector representing Brownian motion, and  $E$  is the fittest solution matrix;  $it$  is the number of iterations and  $maxit$ : maximum iterations. For a lower step size, the equations considered are

$$it > \frac{1}{3} \text{maxit} \tag{25}$$

$$S = \vec{R}_L \otimes (\vec{R}_L \otimes x_{i,j}(t) - E) \tag{26}$$

$$x_{i,j}(t + 1) = E + P \cdot CF \otimes S \tag{27}$$

The optimum solution matrix  $E$  is given as

$$E = \begin{bmatrix} Xb_{1,1}^t & \dots & Xb_{1,d}^t \\ \vdots & \ddots & \vdots \\ Xb_{1,n}^t & \dots & Xb_{n,d}^t \end{bmatrix} \tag{28}$$

For  $n$  search agents,  $d$  dimensions  $X_b$  is the best solution.

The vector  $R_L$  implements the Levy method in the second stage. The optimal power flow problem is addressed for wind energy and FACTS-integrated distribution systems using the Enhanced HPO (EHPO). The works conclude that EHPO is an efficient tool to solve real-world complex power system problems, with scope for analyzing the same for large-scale systems and systems incorporating PV\_DG and EV technologies.

### Improved HPO

[81] proposes an improved version of HPO, IHPO, to overcome the local maximum trap and improve the accuracy of the conventional algorithm. Two steps are involved in studying the IHPO; firstly, the initial population set is generated using the tent chaotic mapping like in [75], followed by an Enhanced Sine Cosine Algorithm (ESCA) adaption and Cauchy's strategy of mutation for the exploration and exploitation stages.

### Improvised initialization phase

The initialization stage is reformed using chaotic mapping, which is characterized by an increase in population diversity. Mathematically, Eq. (16) follows in the other way,

$$y_{i+1} = \begin{cases} \frac{y_i}{u}, 0 \leq y_i < u \\ \frac{1-y_i}{1-u}, u < y_i < 1 \end{cases} \quad (29)$$

$u=0.5$ . The next equation denotes the Tent mapping for controllable randomness of variables to avoid unstable fixed points and trapping into the small periods during iterations. For  $d$  dimensions and  $N$  search agents,

$$y_{i+1}^d = \begin{cases} 2\left(y_i^d + \text{rand}(0, 1) \times \frac{1}{N}\right), 0 \leq y_i^d < 0.5 \\ 2\left(1 - y_i^d + \text{rand}(0, 1) \times \frac{1}{N}\right), 0.5 \leq y_i^d < 1 \end{cases} \quad (30)$$

Hence, the initialization equation updates as follows:

$$x_i^d = l + (u - l) \times y_i^d \quad (31)$$

### Adaption of ESCA

The better global search ability of the SCA is implemented to the standard HPO considering

$$P_v = \exp\left(\mu \times \left(\frac{t}{T}\right)\right)^3 \quad (32)$$

$P_v$  gives the population position, with  $t$  being the ongoing iteration number and  $T$  being the maximum number of iterations.  $\mu$  is the conversion factor. For  $\text{rand} < P_v$ , the population position is updated using the ESCA; if  $\text{rand} > P_v$ , the position is updated using HPO.  $\mu=0.01$  by [82], and in [81],  $\mu=-10$ . The standard SCA is enhanced by initiating hyperbolic sine regulating factor and dynamic cosine wave weight coefficient [82].

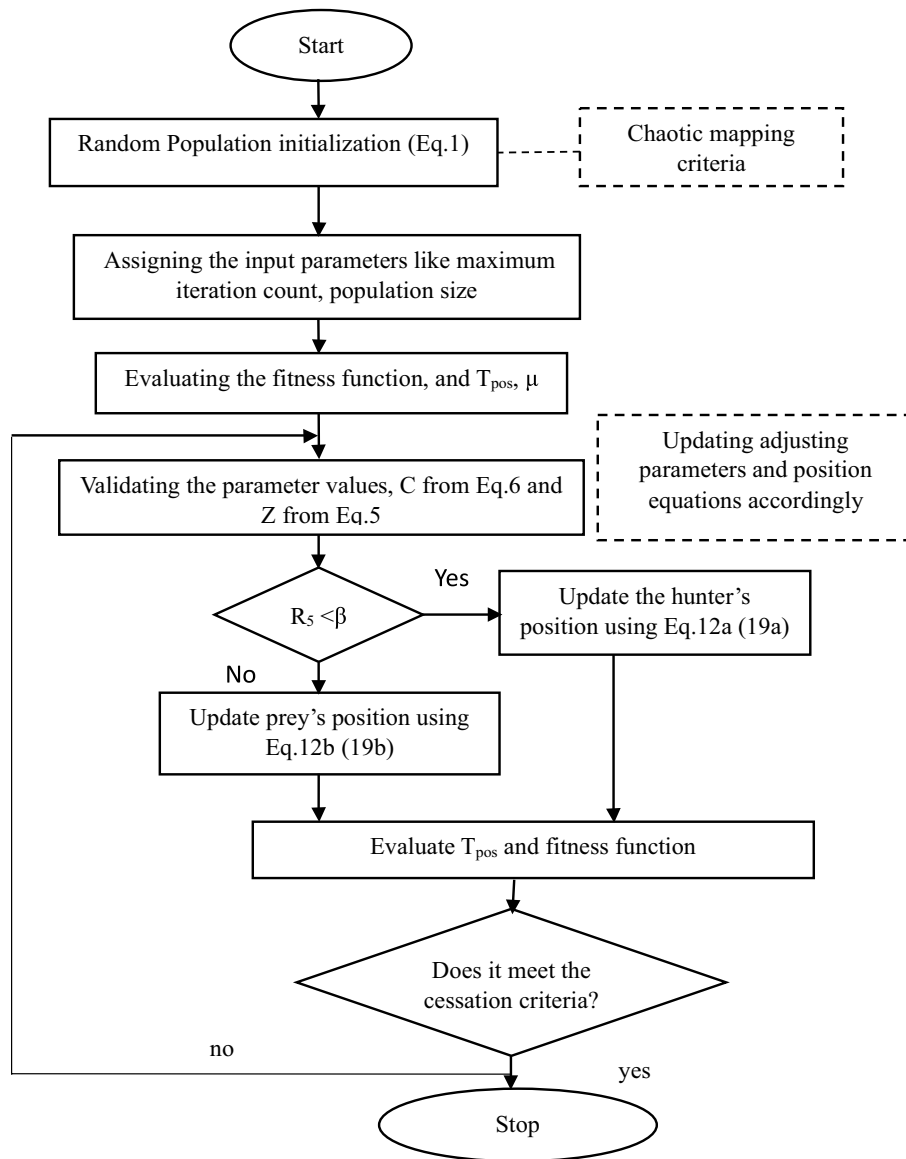


**Cauchy's strategy**

Cauchy's mutation strategy is introduced to improve the convergence speed and convergence accuracy and avoid local maximum trapping. The updated population is given as

$$x_i^d(t + 1) = r_1 \otimes x_i^d(t) + \text{Cauchy} \otimes (x_{\text{best}}^d(t) - x_i^d(t)) \tag{33}$$

$x_i^d(t)$  is the current individual position,  $x_i^d(t + 1)$  is the position after Cauchy mutation,  $r_1$  is a random value ranging in [0,1]. The Cauchy mutation is applied to avoid the early occurrence of optimization. It is referred from [83], where the condition for Cauchy mutation is defined for  $stdY(t) > Cstd$ . The algorithm converges at a good rate, and for  $stdY(t) \leq Cstd$ , there is a chance of a local maxima trap and Eq. (33) coming into the picture. In general, Cstd, the maximum value of the variation coefficient is 0.1, and the



**Fig. 4** Flow chart of IHPO

mutation strategy is applied at  $T/5$  iterations ( $T$  is the maximum iteration count).  $stdY(t)$  is the standard deviation of three consecutive iterations.

The efficacy of the proposed algorithm is tested for eight classic test functions with Wild Horse Optimization (WHO), Gray Wolf Optimization (GWO), Sine Cosine Algorithm (SCA), and HPO. The practical application of the algorithm is the proposed scope of the referred article. Figure 4 illustrates the flow chart of improved HPO.

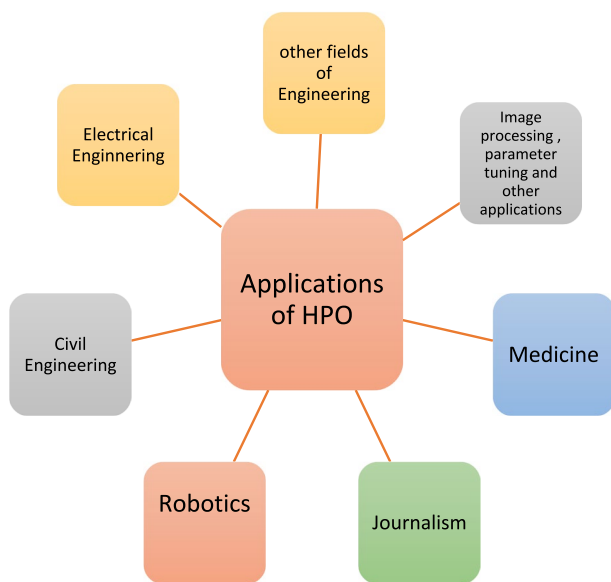
**Inferences and discussions**

From the above literature, HPO is a new nature-inspired meta-heuristic algorithm widely implemented in different engineering fields as an optimization technique. HPOA is applicable in complex systems, construction fields, and image processing because it handles multimodal and non-convex optimization problems. Figure 5 gives a pictorial representation of applications of the HPO algorithm.

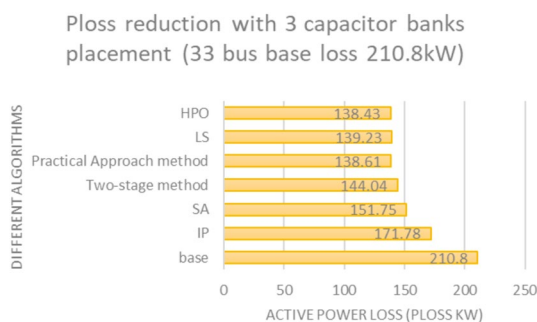
**Electrical engineering applications**

The proposed algorithm HPO finds its application in electrical distribution systems to find the optimal placements of capacitor banks, custom power devices, and renewable energy devices for the system’s loss reduction and voltage enhancement. The following figures accentuate the supremacy of HPO in determining the optimal locations of preferred devices with active loss and energy loss redemption. [84] studies the HPO for optimal capacitor banks in IEEE-33 and 69 bus systems. The comparative graph for power loss reduction for various algorithms highlighting HPO is depicted below:

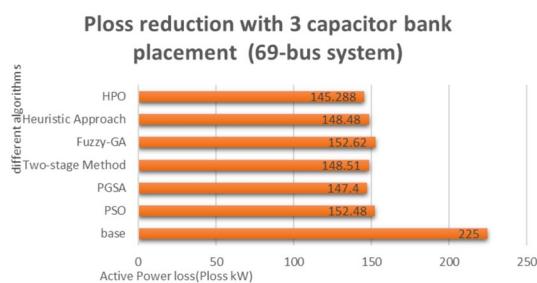
Figures 6 and 7 portray the active loss reduction comparison using different algorithms for IEEE-33 and 69 bus systems with an optimal capacitor bank placement. [69] address the optimal PV placement for active loss reduction; Fig. 8 shows the same for prime locations of PV in an IEEE-33 bus system.



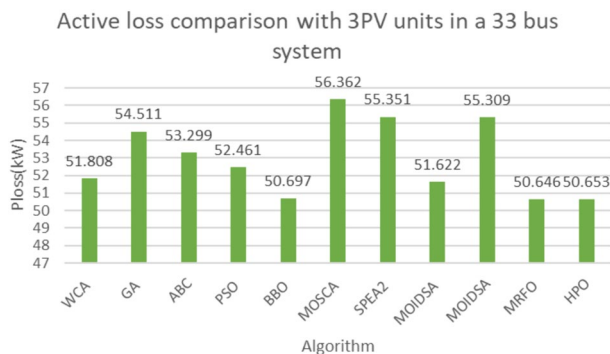
**Fig. 5** Applications of HPO



**Fig. 6** Graphical representation of active loss reduction by capacitor bank placement in IEEE-33 bus system using HPO



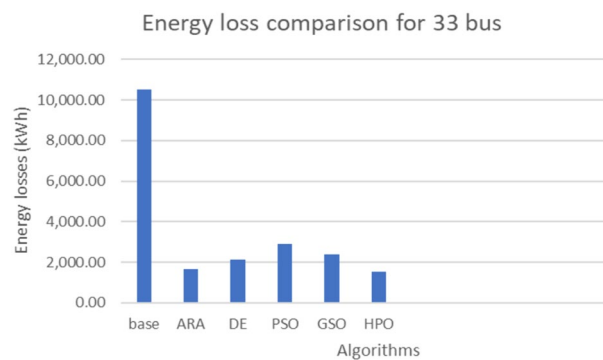
**Fig. 7** Graphical representation of active loss reduction by capacitor bank placement in IEEE-69 bus system using HPO



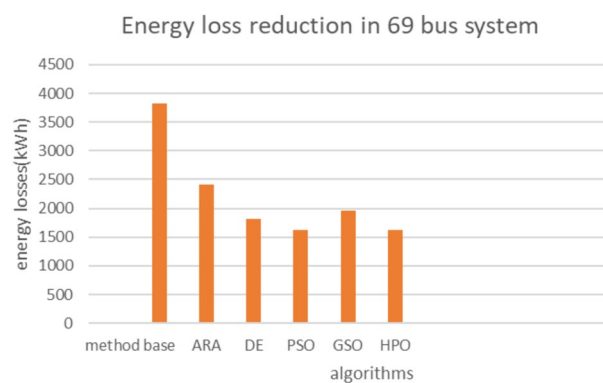
**Fig. 8** Graphical representation of active loss reduction by PV placement in IEEE-33 bus system using HPO.

HPO is analyzed on IEEE-33 and 69 buses for PV\_STATCOM placement in article [68], comparing the energy losses. Figures 9 and 10 graphically depict the superiority of HPO in energy loss reduction collating with the rest.

From the graphical analysis in the above figures, the put-forward algorithm HPO outperforms the other algorithms and is considered effective in tackling electrical engineering problems. Some referred works focus on enlightening the HPO in various established fields other than electrical engineering.



**Fig. 9** Energy loss graphical depiction using various algorithms for an IEEE-33 bus system



**Fig. 10** Energy loss graphical depiction using various algorithms for an IEEE-69 bus system

### Journalism

[71] applies HPO in journalism for fake news detection. The authors proposed HPO with hybrid deep learning (HDL) to identify the false news from the obtained information.

### Civil engineering

An improved version of HPO is used to identify the multi-storey building damages in [75]. The process of damage detection by HPO is followed by the neural network technique, CNN. The two-stage approach is found effective in the respective field.

### Robotics

[76] put forward an improvised HPO for robot path planning. The work highlights using the proposed algorithm combined with a search strategy to help robots search an obstacle-free path in unknown surroundings.

### Rotor dynamics

[85] uses HPO to optimize the squeeze film damper's key parameters to reduce the rotor's vibration performance. The paper highlights the HPO over PSO in convergence speed and accuracy.

### Cloud security

HPO finds its application in cloud security in the article [86], where the algorithm is used to enhance cloud data security by optimizing the multi-objective function with the preservation of information ratio, the ratio of hiding, and the modification degree performed at the critical generation stage as objective functions.

### Medicine

A hybrid combination of HPO with ladybug beetle optimization algorithm is used in ophthalmoscopy to identify Diabetic Retinopathy (DR) [87].

HPO, along with other methodologies or its improved versions, is used in vast fields for the betterment of the desired output. Many works continue to explore the algorithm's efficacy; the advantages and disadvantages of the algorithm are discussed below.

### Highlights of HPO

- **Diverse Exploration:** The predator–prey dynamics encourage the algorithm to cover multiple regions of the solution space simultaneously, aiding in fleeing from local optima.
- **Flexibility:** HPOA can quickly adapt to different problem domains by tuning its parameters and mechanisms.
- **Convergence Speed:** The algorithm often demonstrates faster convergence due to the dynamic interactions between hunters and prey than the other existing algorithms [60].

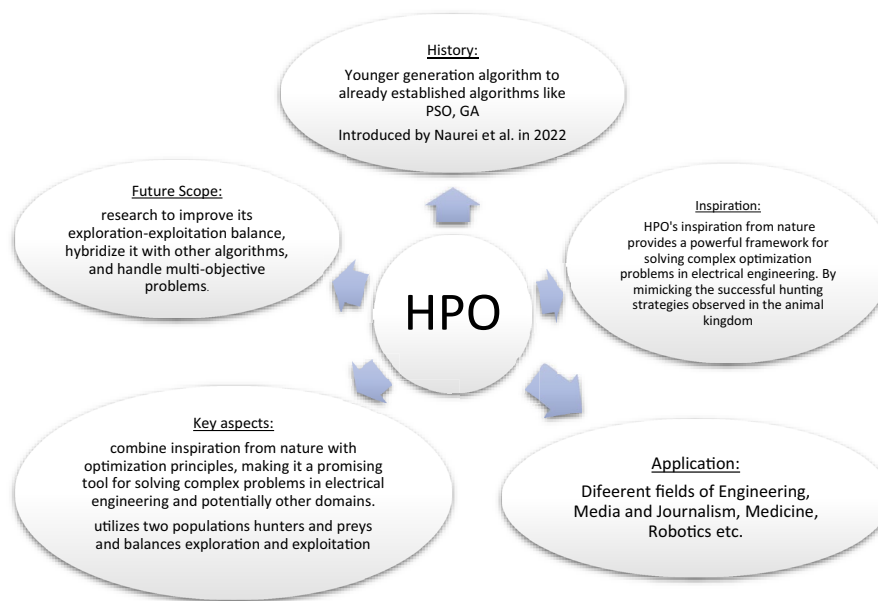
### Drawbacks of HPO

- **Parameter Sensitivity:** Effective parameter tuning is crucial for optimal performance, and improper settings can lead to premature convergence or stagnation. This problem is addressed in [75] work and introduces Cauchy's distribution to avoid the same.
- **Ham in local maxima:** Before getting entrapped in the local maximum and improvised versions, inscribing the same is referred from the literature.

### Conclusion

The Hunter–Prey Optimization Algorithm presents an intriguing approach to optimization inspired by nature's predator–prey interactions. While it has demonstrated success in various applications, it is essential to recognize its strengths and weaknesses when considering its implementation. As research in this field continues to evolve, a deeper understanding of the algorithm's behavior and its potential to solve complex optimization problems will emerge.

In summary, this comprehensive analysis highlights the significance of the Hunter–Prey Optimization Algorithm as a promising optimization technique; Fig. 11 gives an



**Fig. 11** HPO algorithm

overview of the HPO algorithm. Inspired by nature, its unique approach to exploration and exploitation provides a fresh perspective on optimization. The benefits of the review can be addressed below.

For Researchers:

- Consolidated knowledge: The paper summarizes the current state of the art and key concepts of the HPO algorithm.
- Future research directions: By highlighting areas for future work, as suggested previously, the paper can guide researchers toward promising avenues for further development and application of HPO.
- Comparative analysis: A well-structured review paper can compare HPO with other optimization algorithms, highlighting its advantages, limitations, and suitable use cases. This can help researchers choose the most appropriate optimization technique for their problem.

For developers and practitioners:

- Understanding HPO's potential: The paper can showcase the capabilities of HPO for solving various optimization problems. This can encourage developers to explore its application in their projects.
- Practical implementation guidance: The review paper can provide practical insights for implementing HPO with existing algorithms. This can save developers time and effort during implementation.

For the field of optimization:

- Advancement of HPO: The review paper can stimulate further research and development efforts on HPO, improving its performance, adaptability, and theoretical understanding.
- Comparison with existing techniques: By comparing HPO with established optimization algorithms, the paper can contribute to a wider understanding of the strengths and limitations of different approaches. This can guide researchers toward developing more powerful and versatile optimization tools.
- Promoting broader use: A well-disseminated review paper can raise awareness of HPO within the optimization community, potentially leading to its broader adoption in various scientific and engineering fields.

#### Future scope of HPO:

The potential areas where the review can have its future scope can be identified as:

##### Enhancing the HPO algorithm:

- Parameter tuning: While HPO avoids excessive parameters, explore the impact of fine-tuning existing ones like initial population size, levy flight parameters, or the balance factor between exploration and exploitation. [88] enhances the standard HPO by parameter tuning.
- Hybridization with other algorithms: [89, 90] suggests hybridizing with WHO and SFO for improved performance. Works on HPO hybridized with other algorithms like PSO and GA can be a suggested scope in hybridization.
- Multi-objective optimization: [84] applies HPO for multi-objective functions, paving the way for many more works in the same direction.

The future outlook in theoretical analysis can be:

- Investigating whether the current HPO algorithm accurately reflects real-world predator–prey dynamics and exploring potential improvements by incorporating biological concepts can be a direction of future extension.

Additional considerations like the parallelization of HPO for large-scale applications and its adaptiveness to a dynamic environment could also be expected.

#### Abbreviations

ABC	Artificial Bee Colony Algorithm
ACO	Ant Colony Optimization
AIS	Artificial Immune System
ARO	Artificial Rabbits Optimization
ASO	Atom Search Optimization
AVDI	Average Voltage Stability Index
BBO	Biogeography-Based Optimization Algorithm
BESA	Bald Eagle Search Algorithm
BFO	Bacteria Foraging Optimization Algorithm
BH	Black Hole Algorithm
BOA	Butterfly Optimization Algorithm
CaSO	Cat Swarm Optimization
CNN	Convolved Neural Network
COA	Chaotic Optimization Algorithm
CRO	Coral Reef Optimization
CSA	Crow Search Algorithm
CuSO	Curved Space Optimization
DE	Differential Evolutionary

DP	Dynamic Programming
EHPO	Enhanced Hunter–Prey Optimization
EP	Evolutionary Programming
ES	Evolutionary Strategy
FA	Firefly Algorithm
GA	Genetic Algorithm
GBSA	Gravity-Based Search Optimization
GOA	Grasshopper Optimization Algorithm
GSA	Gravitational Search Algorithm
GWO	Grey Wolf Optimizer
HBMO	Honey Bee Mating Optimization
HDL	Hybrid Deep Learning
HHO	Harris Hawk Optimization
HIA	Human-Inspired Algorithm
HPO	Hunter–Prey Optimization
HS	Harmony Search
ICA	Imperialistic Competition Algorithm
IHPO	Improved Hunter–Prey Optimization
IWD	Intelligent Water Drops Algorithm
LFI	Loss Factor Index
LP	Linear Programming
LSI	Loss Sensitivity Index
MFO	Moth-Flame Optimization
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Non-Linear Programming
MIP	Mixed-Integer Programming
MOA	Magnetic Optimization Algorithm
MOIDSA	Multi-Objective Improved Differential Search Algorithm
MOSCA	Multi-Objective Sine Cosine Algorithm
MPA	Marine Predator Algorithm
MRFO	Manta Ray Foraging Optimization Algorithm
MVO	Multiverse Optimization
NLP	Non-Linear Programming
NR	Newton Raphson
PFA	Paddy Fields Algorithm
PGSA	Plant Growth Simulation Algorithm
PIO	Pigeon-Inspired Optimization
PLI	Power Loss Index
PSO	Particle Swarm Optimization
PSOA	Parallel Seeker Optimization Algorithm
SA	Simulated Annealing
SGA	Saplings Growth Algorithm
SPEA	Strength Pareto Evolutionary Algorithm
TLBO	Teaching–Learning-Based Optimization
TS	Tabu Search
VSI	Voltage Stability Index
WCA	Water Cycle Algorithm
WOA	Whale Optimization Algorithm

**Acknowledgements**

Not applicable.

**Author contributions**

Manuscript writing and editing by PSL and VJ helped with literature and the meta-heuristic algorithm. All authors read and approved the manuscript.

**Funding**

The authors received no funding from any source.

**Availability of data and materials**

Not applicable.

**Declarations****Competing interests**

The authors declare no financial or non-financial interests.

Received: 1 December 2023 Accepted: 23 May 2024

Published online: 03 June 2024



## References

1. Behera S, Sahoo S, Pati BB (2015) A review on optimization algorithms and application to wind energy integration to grid. *Renew Sustain Energy Rev* 48:214–227
2. Marouani I, Guesmi T, Alshammari BM, Alqunun K, Alshammari AS, Albadrin S, Rahmani S (2023) Optimized FACTS devices for power system enhancement: applications and solving methods. *Sustainability* 15(12):9348
3. Padmavathi SV, Sahu SK, Jayalaxmi A (2013) Modeling and simulation of static var compensator to enhance the power system security. In: 2013 IEEE Asia Pacific conference on postgraduate research in microelectronics and electronics (PrimeAsia), pp 52–55. IEEE
4. Jordehi AR, Jasni J (2012) Approaches for FACTS optimization problem in power systems. In: 2012 IEEE international power engineering and optimization conference Melaka, Malaysia, pp 355–360. IEEE
5. Momoh JA (2008) Electric power system applications of optimization. CRC, New York, NY
6. Yousefi A, Nguyen TT, Zareipour H, Malik OP (2012) Congestion management using demand response and FACTS devices. *Electr Power Energy Syst* 37:78–85
7. Rezaee Jordehi A, Jasni J (2012) Approaches for FACTS optimization problem in power systems. In: Proceedings of the IEEE international power engineering and optimization conference (PEOCO2012), Melaka, Malaysia, 6–7 June 2012
8. Minguez R, Milano F, Minano RZ, Conejo AJ (2007) Optimal network placement of SVC devices. *IEEE Trans Power Syst* 22:1851–1860
9. Jian Y, Qiang L (2023) Optimal deployment in moving target defense against coordinated cyber-physical attacks via game theory. *Electronics* 12:2484
10. Inkollu SR, Kota VR (2016) Optimal setting of FACTS devices for voltage stability improvement using PSO adaptive GSA hybrid algorithm. *Eng Sci Technol Int J* 19:1166–1176
11. Ram S, Amarnath J (2013) Optimal setting of IPFC for voltage stability improvement using (GA-GSA) hybrid algorithm. In: Proceedings of the Nirma university international conference
12. Phadke AR, Fozdar M, Niazi KR (2012) A new multi-objective fuzzy-GA formulation for optimal placement and sizing of shunt FACTS controller. *Int J Electr Power Energy Syst* 40:46–53
13. Biplab Bhattacharyya B, Kumar VG, Kumar S (2014) Fuzzy-DE approach for the optimal placement of FACTS devices to relief Congestion in a power system. In: Proceedings of the international conference on control, instrumentation, energy & communication, Kolkata, India, 31 January–2 February 2014
14. Yang XS (2005) Ch 32: biology-derived algorithms in engineering optimization. In: Handbook of bio-inspired algorithms and applications. Chapman & Hall/CRC Computer & Information Science Series; 2005. pp 585–96
15. Saad MS, Jamaluddin H, Darus IZ (2012) PID controller tuning using evolutionary algorithms. *WSEAS Trans Syst Control* 7(4):139–149
16. Sinha N, Chakrabarti R, Chattopadhyay PK (2003) Fast evolutionary programming techniques for short-term hydro-thermal scheduling. *Electr Power Syst Res* 66(2):97–103
17. Bäck T, Schwefel H-P (1993) An overview of evolutionary algorithms for parameter optimization. *Evol Comput* 1(1):1–23
18. Hansen N, Arnold DV, Auger A (2013) Evolution strategies; 5 Apr 2013
19. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
20. Jayakumar T (2008) Steady state security and economic power dispatch using multiobjective evolutionary programming. *Int J Appl Eng Res* 3(8):1073–1085
21. Storn R (1996) On the usage of differential evolution for function optimization. In: Proceedings of the fuzzy information processing society. Berkeley (CA, USA); pp 519–23
22. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10(6):646–657
23. Kennedy J, Eberhart R (1995) Particle swarm optimization. *Proc IEEE Int Conf Neural Netw* 4(2):1942–8
24. Passino KM (2010) Bacterial foraging optimization. *Int J Swarm Intell Res* 1(1):1–16
25. Yang XS, Deb S (2010) Engineering optimisation by cuckoo search. *Int J Math Model Numer Optim* 1(4):330–343
26. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Proceedings of the stochastic algorithms: foundation and applications, SAGA, 5792; 2009. pp 169–78
27. Duan H, Qiao P (2014) Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *Int J Intell Comput Cybern* 7(1):24–37
28. Sun H, Duan H (2014) PID controller design based on prey-predator pigeon-inspired optimization algorithm. In: Proceedings of 2014 IEEE international conference on mechatronics and automation. Tianjin (China); August 3–6, pp 1416–20
29. Salcedo-Sanz S, DelSer J, Landa-Torres I, Gil-López S, Portilla-Figueras A (2014) The coral reefs optimization algorithm: a novel meta-heuristic for solving hard optimization problems. *Sci World J* 1–15
30. Basu M (2011) Artificial immune system for fixed head hydrothermal power system. *Energy* 36(1):606–612
31. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
32. Wang L et al (2022) Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng Appl Artif Intell* 114:105082
33. Hussien AG et al (2020) Crow search algorithm: theory, recent advances, and applications. *IEEE Access* 8:173548–173565
34. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89:228–249
35. Mirjalili S, Seyed MM, Andrew L (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
36. Yang XS (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. Springer, Berlin
37. Fathian M, Amiri B, Maroosi A (2007) Application of honey-bee mating optimization algorithm on clustering. *Appl Math Comput* 190(2):1502–1513

38. Kawambwa S, Mnyanghwalo D (2023) A multi-agent-based symbiotic organism search algorithm for DG coordination in electrical distribution networks. *J Electr Syst Inf Technol* 10:6. <https://doi.org/10.1186/s43067-023-00072-7>
39. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23:715–734
40. Injeti SK, Thunuguntla VK (2020) Optimal integration of DGs into radial distribution network in the presence of plug-in electric vehicles to minimize daily active power losses and to improve the voltage profile of the system using bio-inspired optimization algorithms. *Prot Control Mod Power Syst* 5:3. <https://doi.org/10.1186/s41601-019-0149-x>
41. Ibrahim MK, Ali RS (2016) Novel optimization algorithm inspired by camel traveling behavior. *Iraqi J Electr Electron Eng* 12(2):167–177
42. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Studies in computational intelligence, vol 284 (pp. 65–74). Berlin, Heidelberg: Springer.
43. Meraihi Y et al (2021) Grasshopper optimization algorithm: theory, variants, and applications. *IEEE Access* 9:50001–50024
44. Rashedi E, Nezamabadi-pour H, Saryzadi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248
45. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Sci J* 220(4598):671–680
46. Deeb N (1992) Simulated annealing in power systems. In: *Proceedings of IEEE international conference on man systems and cybernetics; 1992*. 1086–9
47. Tayarani NMH, Akbarzadeh-T, MR (2008) Magnetic optimization algorithms a new synthesis. In: *Proceedings of IEEE congress on evolutionary computation, CEC 2008 (IEEE world congress on computational intelligence); 2008*. pp 2659–64
48. Shah-Hosseini H (2008) Optimization with the nature-inspired intelligent water drops algorithm. *Int J Intell Comput Cybern* 1(2):193–212
49. Devarapalli R, Bhattacharyya B (2021) Power and energy system oscillation damping using multi verse optimization. *SN Appl Sci* 3:383
50. Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl-Based Syst* 163:283–304
51. Moghaddam FF, Reza FM, Mohamed C (2012) Curved space optimization: a random search based on general relativity theory. *arXiv preprint arXiv:1208.2214*
52. Shah-Hosseini H (2011) Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *Int J Comput Sci Eng* 6(1–2):132–140
53. Eskandar H et al (2012) Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110:151–166
54. Kumar S, Deepanwita D, Sanjay KS (2015) Black hole algorithm and its applications. In: *Computational intelligence applications in modeling and control*, pp 147–170
55. Geem ZM (2009) *Music-inspired harmony search algorithm: theory and applications*. Springer, Berlin
56. Gargari EA, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: *Proceedings of IEEE congress on evolutionary computation*; pp 4661–7
57. Glover F, Taillard E, de Werra D (1993) A user's guide to tabu search. *Ann Oper Res* 41(1):3–28
58. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96:120–133
59. Shafik MB et al (2019) Adaptive multi objective parallel seeker optimization algorithm for incorporating TCSC devices into optimal power flow framework. *IEEE Access* 7:36934–36947
60. Naruei I, Farshid K, Amir SM (2022) Hunter–prey optimization: algorithm and applications. *Soft Comput* 26(3):1279–1314
61. Rao RV (2016) *Teaching-learning-based optimization algorithm*. Springer International Publishing, Berlin
62. Sayed GI, Soliman MM, Hassanien AE (2021) A novel melanoma prediction model for imbalanced data using optimized SqueezeNet by bald eagle search optimization. *Comput Biol Med* 136:104712
63. Yan XF, Chen DZ, Hu SX (2003) Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. *Comput Chem Eng* 27(10):1393–1404
64. Geem ZM (2009) *Music-inspired harmony search algorithm: theory and applications*. Springer, Berlin
65. Premaratne U, Samarabandu J, Sidhu T (2009) A new biologically inspired optimization algorithm. In: *2009 international conference on industrial and information systems (ICIIS)*, pp 279–284. IEEE
66. Karci A (2007) Theory of saplings growing up algorithm. In: *International conference on adaptive and natural computing algorithms*, Springer, pp 450–460
67. Zhang LM, Dahmann C, Zhang Y (2009) Human-inspired algorithms for continuous function optimization. In: *2009 IEEE international conference on intelligent computing and intelligent systems, IEEE*, pp 318–321
68. Shaheen AM et al (2023) Optimal allocation of PV-STATCOM devices in distribution systems for energy losses minimization and voltage profile improvement via Hunter-Prey-based algorithm. *Energies* 16(6):2790
69. Inkollu SR, et al. (2022) An application of Hunter-Prey optimization for maximizing photovoltaic hosting capacity along with multi-objective optimization in radial distribution network. *Int J Intell Eng Syst* 15(4)
70. Lahari PS, Varaprasad J (2022) Hunter Prey optimization for optimal allocation of photovoltaic units in radial distribution system for real power loss and voltage stability optimization. In: *Congress on intelligent systems*. Singapore: Springer Nature Singapore, 2022.
71. Alshahrani HJ, et al. (2023) Hunter Prey optimization with hybrid deep learning for fake news detection on Arabic corpus. *Comput Mater Continua* 75(2)
72. Elshahed M et al (2022) An innovative hunter-prey-based optimization for electrically based single-, double-, and triple-diode models of solar photovoltaic systems. *Mathematics* 10(23):4625
73. Hassan MH, et al (2023) An enhanced hunter-prey optimization for optimal power flow with FACTS devices and wind power integration. *IET Gen Trans Distrib*
74. Wang X et al (2023) Short-term wind power prediction by an extreme learning machine based on an improved Hunter–Prey optimization algorithm. *Sustainability* 15(2):991

75. Xiang C et al (2022) Structural damage identification based on convolutional neural networks and improved Hunter-Prey optimization algorithm. *Buildings* 12(9):1324
76. Abdulsahab JA, Kadhim DJ (2023) Multi-objective robot path planning using an improved hunter prey optimization algorithm. *Int J Intell Eng Syst* 16(2)
77. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82. <https://doi.org/10.1109/4235.585893>
78. Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83:80–98
79. Heidari AA et al (2019) Harris hawks optimization: algorithm and applications. *Fut Gen Comput Syst* 97(2019):849–872
80. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. *Expert Syst Appl* 152:113377
81. Fu M, Qiang L (2022) An improved Hunter-prey optimization algorithm and its application. In: 2022 IEEE international conference on networking, sensing and control (ICNSC). IEEE
82. He Qing Xu, Qinshuai WK (2019) Optimization of wireless sensor node deployment based on improved sine cosine algorithm. *Comput Appl* 39(07):2035–2043
83. Li Jianfeng Lu, Di LH (2022) An improved atomic search algorithm. *J Syst Simul* 34(03):490–502
84. Pappu SL, Janamala V (2023) A multi-objective hunter-prey optimization for optimal integration of capacitor banks and photovoltaic distribution generation units in radial distribution systems. *ITEGAM-JETIA* 9(43):42–53
85. Xu Y et al (2024) A comprehensive multi-parameter optimization method of squeeze film damper-rotor system using hunter-prey optimization algorithm. *Tribol Int* 194:109538
86. Sugitha G, Ayshwarya LS (2024) A multi-objective privacy preservation model for cloud security using hunter prey optimization algorithm. *Peer-to-Peer Networking and Applications* 1–13
87. Sagvekar V et al (2024) Hybrid hunter-prey ladybug beetle optimization enabled deep learning for diabetic retinopathy classification. *Biomed Signal Process Control* 95:106346
88. Qu C, Lu Z, Peng X, Lin G (2023) A Hunter-Prey algorithm coordinating mutual benefit and sharing and interactive learning for high-efficiency design of photovoltaic models. *Int J Intell Syst* 30:2023
89. Aluri L, Latha D. HSFO: Hunter sail fish optimizer enabled deep learning for single document abstractive summarization based on semantic role labelling for Telugu text.
90. Ramadan HA, Khan B, Diab AA (2022) Accurate parameters estimation of three diode model of photovoltaic modules using hunter-prey and wild horse optimizers. *IEEE Access* 16(10):87435–87453

### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.