



Advances in Spotted Hyena Optimizer: A Comprehensive Survey

Shafih Ghafari¹ · Farhad Soleimani Gharehchopogh¹

Received: 25 April 2021 / Accepted: 25 June 2021 / Published online: 5 July 2021
© CIMNE, Barcelona, Spain 2021

Abstract

Metaheuristic algorithms are widely used in various fields of optimization engineering. These algorithms have become popular because of their ability to explore and exploit solutions in various problem areas. The Spotted Hyena Optimizer (SHO) algorithm is a metaheuristic algorithm inspired by the life of spotted hyenas, introduced by Dhiman and Kumar (2017) to solve continuous optimization problems. Various studies have been performed based on changes in the SHO algorithm to solve various problems due to its effectiveness and success in solving continuous problems. This paper aims to comprehensively survey the application of the SHO algorithm in solving various optimization problems. In this paper, SHO algorithms are categorized based on hybridization, improvement, SHO variants, and optimization problems. This study invites researchers and developers of meta-heuristic algorithms to employ the SHO algorithm for solving diverse problems since it is a simple and robust algorithm for solving intricate and NP-hard problems. Based on the studies, it was concluded that the SHO algorithm had been used more in optimization problems. The purpose of optimization problems is to find optimal solutions and finding global points in the problem environment. Also, the SHO algorithm establishes a good trade-off between the exploration and extraction stages. Based on the done studies and investigations, properties and factors of the SHO algorithm are better than another meta-heuristic algorithms, which has increased its adaptability and flexibility in different fields.

1 Introduction

Although mathematical programming methods have stable and logical-mathematical foundations for solving optimization problems, these methods cannot solve non-convex, non-linear, and intermittent problems with high dimensions and high complexity. These methods have a long computation time and often do not converge to the absolute optimal solution. They do not even allow us to find a definite optimal solution. Mathematical optimization techniques suffer from getting stuck in local optima. The algorithm assumes a locally optimal solution as a globally optimal solution and fails to find the global optimum. Also, mathematical methods can often not be used for unknown derivatives or non-differentiable [1, 2].

It is challenging to solve complex optimization problems with optimization algorithms. On the other hand, there is no algorithm capable of solving all optimization problems. It means that a method may show good performance on a

problem set but cannot find optimal solutions to different problems. Such problems are addressed through the widespread use of metaheuristic algorithms and population-based optimization methods. Due to their simplicity, the low number of parameters, the high degree of dependence on the type of problem, and lack of complexity, metaheuristic algorithms achieve optimal solutions [3–5]. Metaheuristic algorithms are intelligent methods generally inspired by nature and the evolution of animals and plants. Over the last few decades, these algorithms have been used in conjunction with rapid advances in complex problem solving and optimization problems [6–8]. Although they may not definitively guarantee finding optimal solutions, these methods increase the likelihood of finding a globally optimal solution or a near-optimal solution due to the mechanisms and operators used by directing the search operation to utility [9, 10]. Figure 1 shows the most critical advantages of metaheuristic algorithms compared to mathematical programming methods.

Recent decades have seen an increase in the evaluation and use of metaheuristic algorithms to solve optimization issues. Metaheuristic algorithms include Ant Colony Optimization (ACO) [11], Symbiotic Organisms Search (SOS) [12], Whale Optimization Algorithm (WOA) [13], Krill Herd Algorithm (KHA) [14], Crow Search Algorithm (CSA)

✉ Farhad Soleimani Gharehchopogh
bonab.farhad@gmail.com

¹ Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

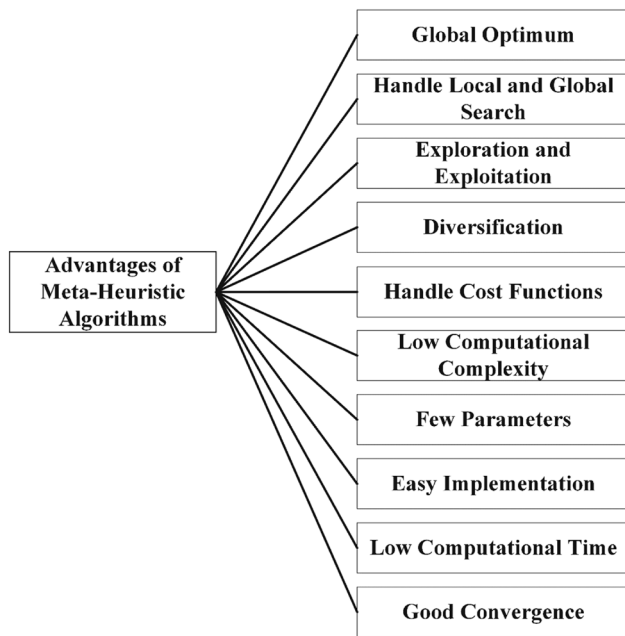


Fig. 1 Advantages of metaheuristic algorithms compared to mathematical programming methods

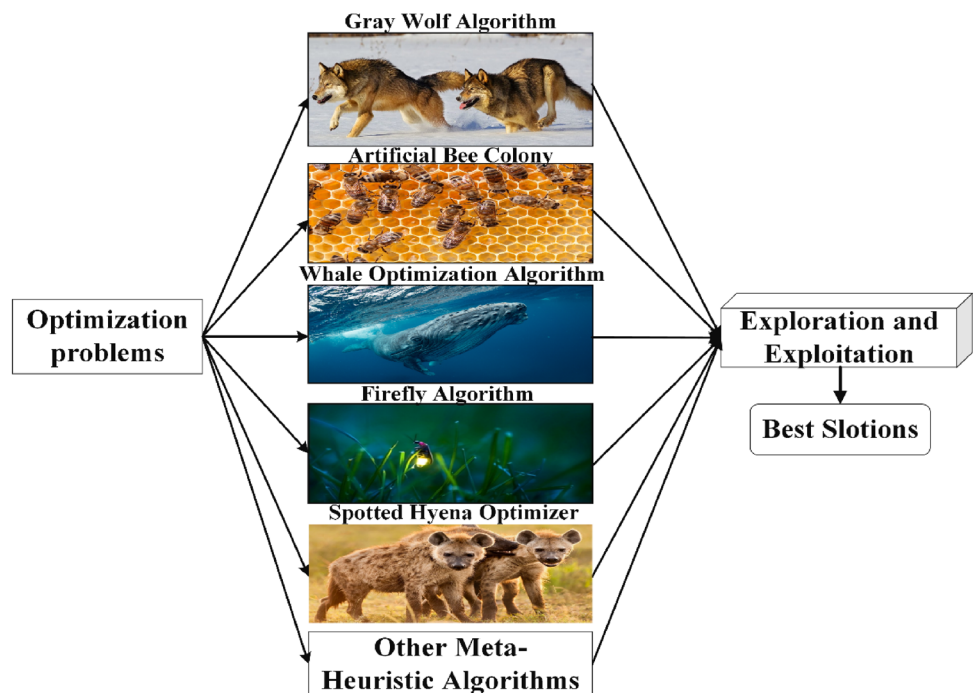
[15], Social Spider Optimization (SSO) [16], Water Wave Optimization (WWO) [17], Firefly Algorithm (FA) [18], Dragonfly Algorithm (DA) [19], Emperor Penguin Optimizer (EPO) [20], Antlion Optimizer (ALO) [21], Butterfly Optimization Algorithm (BOA) [22], Moth-Flame Optimization (MFO) [23], Rain optimization algorithm (ROA)

[24], Horse Optimization Algorithm (HOA) [25], Search Group Algorithm (SGA) [26], Sine Cosine Algorithm (SCA) [27], Water Strider Algorithm (WSA) [28], and so on.

Optimization is finding value for problem variables to optimize the objective function. Optimization models are generally divided into two categories: Single-Objective Optimization Problems (SOPs) [29] and Multi-Objective Optimization Problems (MOPs) [30]. In SOPs, the goal is to optimize one criterion (for example, calculating the minimum cost), while in MOPs, several criteria are optimized simultaneously (for example, calculating the minimum cost and maximum accuracy).

As shown in Fig. 2, the most critical factor that controls a metaheuristic algorithm’s performance and accuracy is the trade-off between exploration and extraction. Exploration refers to a metaheuristic algorithm’s ability to search different search environment areas to discover the optimal solution [31, 32]. On the other hand, extraction is the skill to centralize search within the optimal range to extract the desired solution. A metaheuristic algorithm balances these two conflicting goals well. In any metaheuristic algorithm or the improved version, performance is improved by controlling the afore-mentioned parameters. In the initial iterations, extraction power should usually be increased, and extraction power should be gradually emphasized [33, 34]. It means that, in the initial iterations, metaheuristic algorithms must search the problem environment in various ways and search the areas found with greater accuracy in the final iterations. The main goal is to find the best trade-off between exploration and extraction

Fig. 2 Extraction and extraction agents in metaheuristic algorithms



to ensure optimal solutions are found. Exploration is used to discover new solutions and prevent getting stuck in local convergence, while extraction is used to find the best solutions [35, 36]. In the extraction phase, the best solutions are discovered using the solutions found in the exploration phase. Most metaheuristic algorithms show good performance in exploration or extraction.

Unlike classical methods, metaheuristic algorithms use a population of solutions in their process instead of using a single solution and seek to find the optimal solution using their operators [37, 38]. They can also use some valuable operators to find optimal solutions to specific and complex problems. In general, the performance of metaheuristic algorithms can be improved by taking the following steps:

- Changing the pattern of initialization of agents in metaheuristic algorithms
- Adding new operators and parameters to metaheuristic algorithms or change operators and parameters
- Improving metaheuristic algorithms using the parameter setting technique
- Improving metaheuristic algorithms using the multi-population technique
- Hybrid algorithms
- Using information exchange mechanisms between agents

The SHO algorithm was introduced as a metaheuristic algorithm in 2017 [39]. The SHO has been evaluated on 29 famous test functions and engineering problems. It has been proven that this algorithm can solve problems with problems such as computational complexity and reduced convergence of solutions. The main contributions of this paper are as follows:

- We are analyzing the SHO algorithm based on flow-chart and pseudocode.
- We are reviewing SHO methods in terms of hybridization, improvement, SHO variants, and optimization problems.
- We are analyzing SHO efficiency in solving various problems based on convergence rate, exploration, and extraction.
- We are emphasizing future research in line with the SHO.

The rest of this paper is structured as follows: Sect. 2 describes the SHO algorithm and its operators. Section 3 categorizes SHO methods into four areas: hybridization, improvement, SHO variants, and optimization problems. Section 4, substantial explanations and essential points are discussed. Finally, Sect. 5 provides the conclusion and future works.

2 Spotted Hyena Optimization (SHO) Algorithm

Spotted hyenas exhibit an interesting group hunting behavior. Inspired by these behaviors, Dhiman et al. (2017) presented SHO to solve optimization problems [39]. SHO is a population-based metaheuristic algorithm with a group approach inspired by spotted hyena hunting behavior to solve computer science optimization problems. The process of collecting papers to investigate SHO is shown in Fig. 3. In order to collect papers, we searched various journals such as Springer, Elsevier, Wiley, IEEE, Google search, etc., based on keywords and collected research papers. Initially, most research papers were deleted after reading the title of the papers because their titles did not match the present research. In addition, some research papers were deleted due to a lack of validity and preliminary conclusions. Initially, we collected 52 research papers on the SHO algorithm. However, most of the papers were deleted after the study due to a

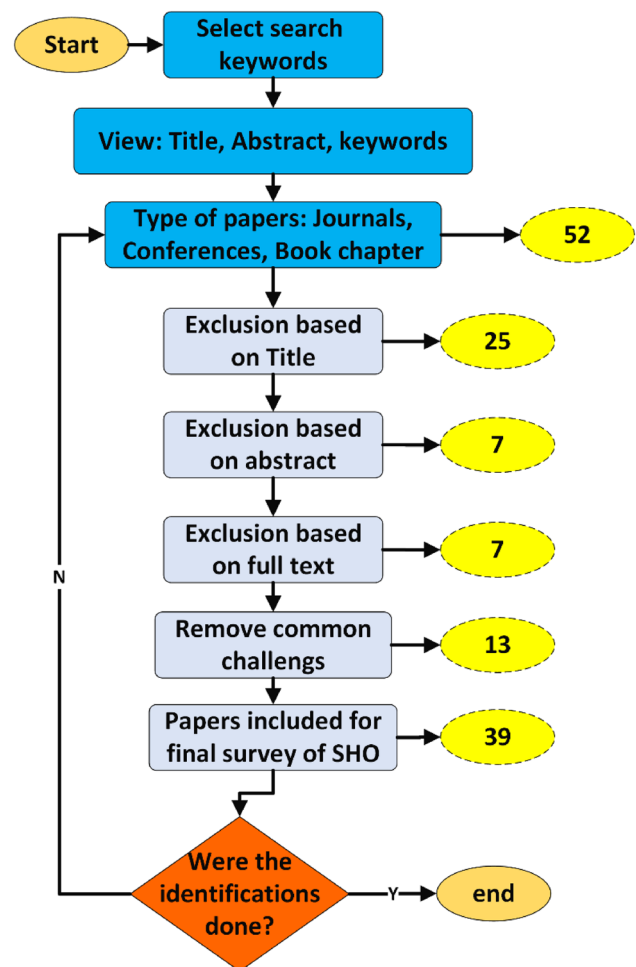


Fig. 3 Review technique of papers identification in SHO algorithm

lack of connection with the SHO algorithm. The remaining research papers were analyzed based on title, abstract, full text and keywords, which were about 39 primary papers.

Since 2017, various studies have been conducted by SHO to solve optimization problems. First, we downloaded all the papers that used SHO. Figure 4 shows the percentage of papers that used SHO, published in various journals. As it turns out, Springer had the highest percentage of published papers.

Figure 5 shows the number of papers published that researchers proposed from SHO by year. As can be seen, the number of published papers from SHO in 2018 and May 2020 were 5 and 12 papers, respectively. Figure 5 indicates a gradual increase in SHO papers in 2021 in comparison with 2017.

To hunt their prey, spotted hyenas use a four-stage group mechanism as follows:

- (1) Search the surroundings and track the prey.
- (2) pursue the prey to make it bored and easy to hunt
- (3) encircle the prey by a group of spotted hyenas to hunt at the right time
- (4) make the final attack on the prey and hunt it.

To model SHO, it is assumed that each spotted hyena's position is a solution to the optimization problem, and the position of the best-spotted hyena estimates the position of the prey. According to Fig. 6, spotted hyenas or problem-solving solutions can circle the optimal solution (or prey)

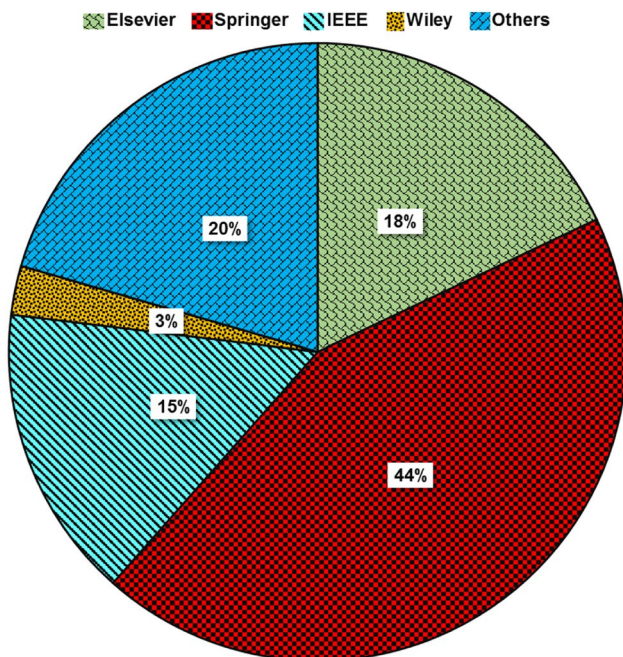


Fig. 4 Percentage of papers that used SHO, published in various journals

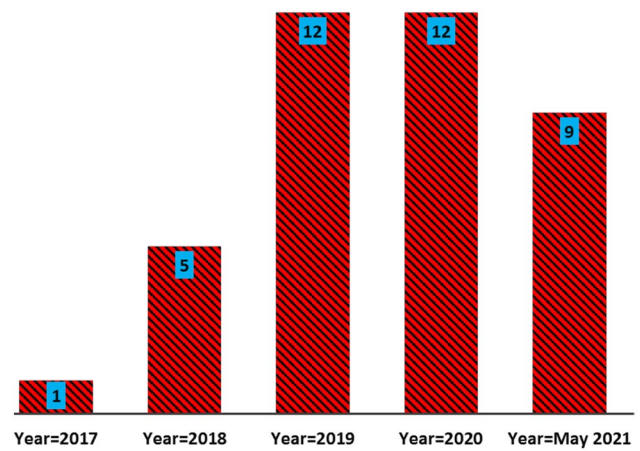


Fig. 5 Number of papers proposed from SHO by year

in different states and situations to discover possible and optimal solutions.

Figure 6 displays the effect of Eq. (1) and Eq. (2) in a 2D environment. According to Fig. 7, spotted hyenas (A, B) can improve their location relative to their prey position (A^*, B^*). By adjusting the value of the vectors \vec{B} and \vec{E} miscellaneous numbers of places are created that can be about the current position. Spotted hyenas or problem-solving hyenas can have two states in the face of the optimal solution. In the first case, they simply search for the optimal solution (Fig. 8), which is considered a kind of exploration in the problem environment, or they attack the optimal point, i.e., the position of the prey, and hunt it, according to Fig. 9.

Encircle the prey In this algorithm, it is assumed that the prey is located near the optimal spotted hyena position, and the optimal spotted hyena tries to direct the members of the population to that point. SHO assumes that the spotted

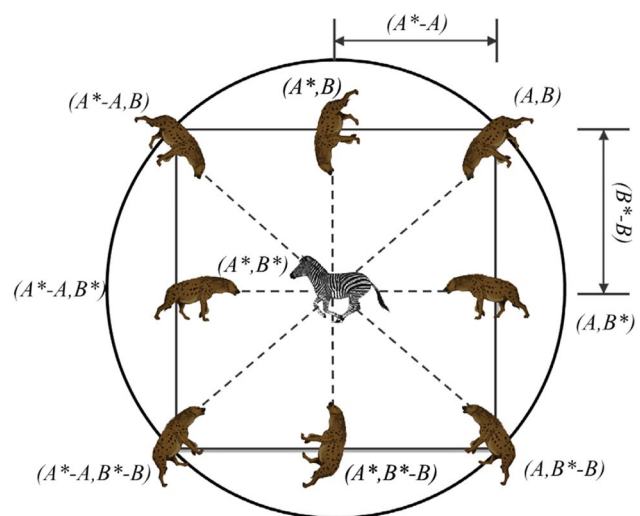


Fig. 6 The 2D position of spotted hyena vectors [39]

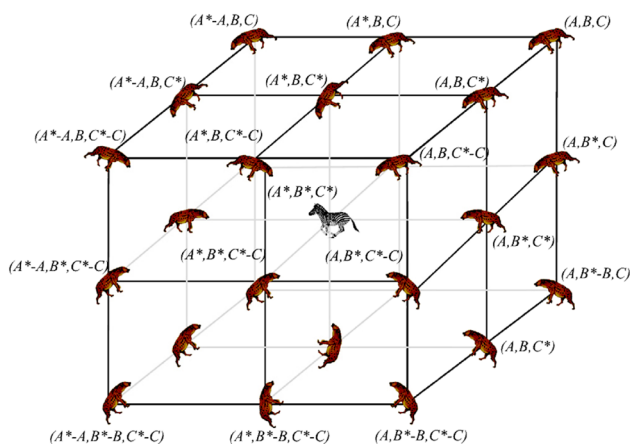


Fig. 7 Position vectors in 3D and following possible positions of spotted hyena [39]

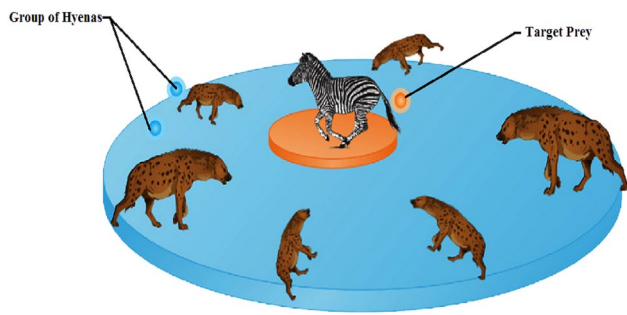


Fig. 8 Search for spotted hyenas around the best-fit solution (exploration) [39]

hyena’s optimal position in the problem environment is considered the prey’s estimated point, which directs the other members to this point.

Spotted hyenas can detect the position of the prey and encircle it. In this algorithm, the prey is known as the near-optimal solution. Therefore, the best member of the population is determined first. The rest of the members are trying to change and update their position towards this

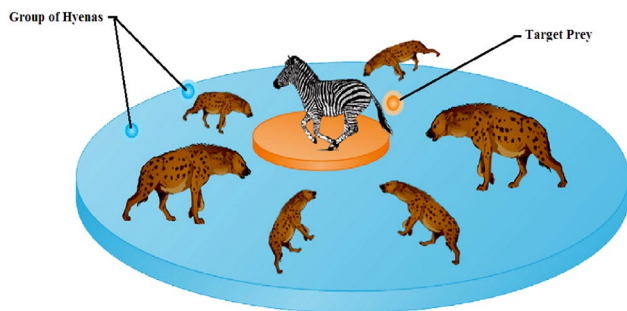


Fig. 9 Spotted hyenas attack best-fit (exploitation) [39]

population member. The encircling mechanism used for spotted hyenas is modeled using Eq. (1).

$$\vec{D}_h = \left| \vec{B} \cdot \vec{P}_p(x) - \vec{P}(x) \right| \tag{1}$$

$$\vec{P}(x + 1) = \vec{P}_p(x) - \vec{E} \cdot \vec{D}_h \tag{2}$$

In Eq. (1) D is the distance between a spotted hyena/solution to the problem and the position of the prey ($\vec{P}_p(x)$) and $\vec{P}(x + 1)$ is the new position of a spotted hyena in the new iteration or $x + 1$. Besides, x is the iteration number of the algorithm. Equation (1) can be used to calculate a spotted hyena’s distance from its prey as an encircling mechanism. In Eqs. (1) and (2), x shows the current iteration, \vec{B} and \vec{E} are vector coefficients, \vec{P}_p is the position vector of the best current solution, \vec{P} is the position vector of spotted hyenas, $||$ is the value of absolute, and “.” is an element-by-element multiplication. The vectors \vec{B} and \vec{E} are computed according to Eq. (3) and Eq. (4).

$$\vec{B} = 2 \cdot r\vec{d}_1 \tag{3}$$

$$\vec{E} = 2\vec{h} \cdot r\vec{d}_2 - \vec{h} \tag{4}$$

$$\vec{h} = 5 - \left(\text{iteration} * \left(\frac{5}{\text{Max}_{\text{Iteration}}} \right) \right); \tag{5}$$

where $\text{iteration} = 1, 2, 3, \dots, \text{Max}_{\text{Iteration}}$

In Eq. (4), a linear reduction from 5 to 0 is made by \vec{h} during the iteration period, and $r\vec{d}_1, r\vec{d}_2$ are random vectors in the range $[0,1]$. The parameter $\text{Max}_{\text{Iteration}}$ displays the high number of iterations. A large number of iterations can enhance exploitation. Figure 7 displays the updated position of spotted hyenas in a 3D environment.

Hunting To mathematically define spotted hyenas’ behavior, it is assumed that the best search factor is an optimal factor that is aware of the prey’s position. Other search agents try to form a group to move towards the best search factor and update the best solutions so far.

$$\vec{D}_h = \left| \vec{B} \cdot \vec{P}_h - \vec{P}_k \right| \tag{6}$$

$$\vec{P}_k = \vec{P}_h - \vec{E} \cdot \vec{D}_h \tag{7}$$

$$\vec{C}_h = \vec{P}_k + \vec{P}_{k+1} + \dots + \vec{P}_{k+N} \tag{8}$$

\vec{P}_h to be the best position of the first spotted hyena, and \vec{P}_k to be the position of other spotted hyenas. Also, N is the number of spotted hyenas defined by Eq. (9).

$$N = \text{count}_{nos}(\vec{P}_h + \vec{P}_{h+1} + \vec{P}_{h+2}, \dots, (\vec{P}_h + \vec{M})) \quad (9)$$

In Eq. (9), \vec{M} is a random vector in the interval [0.5, 1]. The number of solutions and all selective solutions are determined by *nos*. Also, \vec{C}_h is a group of N optimal solutions.

Attacking Prey (Exploitation) The value of the vector \vec{h} are reduced for mathematical modeling to attack the prey. The value of the vector \vec{E} are also reduced to change the value of the vector \vec{h} , probably from 5 to 0 during iteration. As shown in Fig. 9, if E's value is $|E| < 1$, the group of pure spotted hyenas is forced to attack the prey. Attack on prey is defined as Eq. (10).

$$\vec{P}(x+1) = \frac{\vec{C}_h}{N} \quad (10)$$

where $\vec{P}(x+1)$ stores and updates the best position, and other search factors are changed based on the best factor position.

Exploration Based on changes in the vector \vec{E} . This method can be used to search for prey (random search). The vector \vec{E} is defined with a random value greater than 1 or less than -1 to ensure that search factors move away

from the prey. In the random search step, a search factor's position is achieved based on the random selection of search factors instead of the best current search factor. As shown in Fig. 7, the $|E| > 1$ mechanism emphasizes random search and allows the SHO to perform a global search.

The steps of SHO are:

Step 1 Produce the initial individuals of spotted hyenas (P_i), where $i = 1, 2, \dots, n$

Step 2 Initialize the variables, such as B, E, h and N, and the maximum iterations.

Step 3 The fitness value of each search agent is computed.

Step 4 In the search environment, it is discovered that the best search agents

Step 5 Determine the group of best solutions using Eq. (8) and Eq. (9)

Step 6 Using Eq. (10) position of search agents is updated.

Step 7 Check and adjust the search agents' search environment to prevent them from exceeding a specific limit.

Step 8 Calculate the suitability of the updated search agents and the vector Ph update if there is a better solution than the former optimal solution.

Step 9 Update the Spotted Hyenas (C_h) Group to Update Search agent Fit

Step 10: The algorithm stops if satisfied. Otherwise, it goes back to Step 5.

Input: P_i ($i = 1, 2, \dots, n$) is the spotted hyenas' population .

Output: The best search agents are discovered.

01: **procedure** SHO

02: The variables h, B, E, and N are Initialed.

03: It is computed the fitness of each search agent.

04: P_h = the best search agent

05: C_h = the group or class of all unseemly optimal solutions

06: **while** ($x < \text{Max number of iterations}$) **do**

07: **for** each search agent, **do**

08: The position of the current agent is updated by Eq. (10)

09: **end for**

10: Update h, B, E, and N

11: The range of each search agent should be checked and adjusted if their value exceeds the search range

12: The fitness is computed to each search agent

13: Update P_h if there is a better solution than the prior optimal solution

14: The group C_h and P_h are updated

15: $x = x + 1$

16: **end while**

17: **return** P_h

18: **end**

Fig. 10 Pseudocode of spotted hyena optimizer [39]

Step 11: Show the best optimal solution after satisfying the stopping criteria

Figure 10 presents the pseudocode of the SHO algorithm. A diagram of the SHO algorithm is described in Fig. 11. The most important key features of the SHO algorithm are:

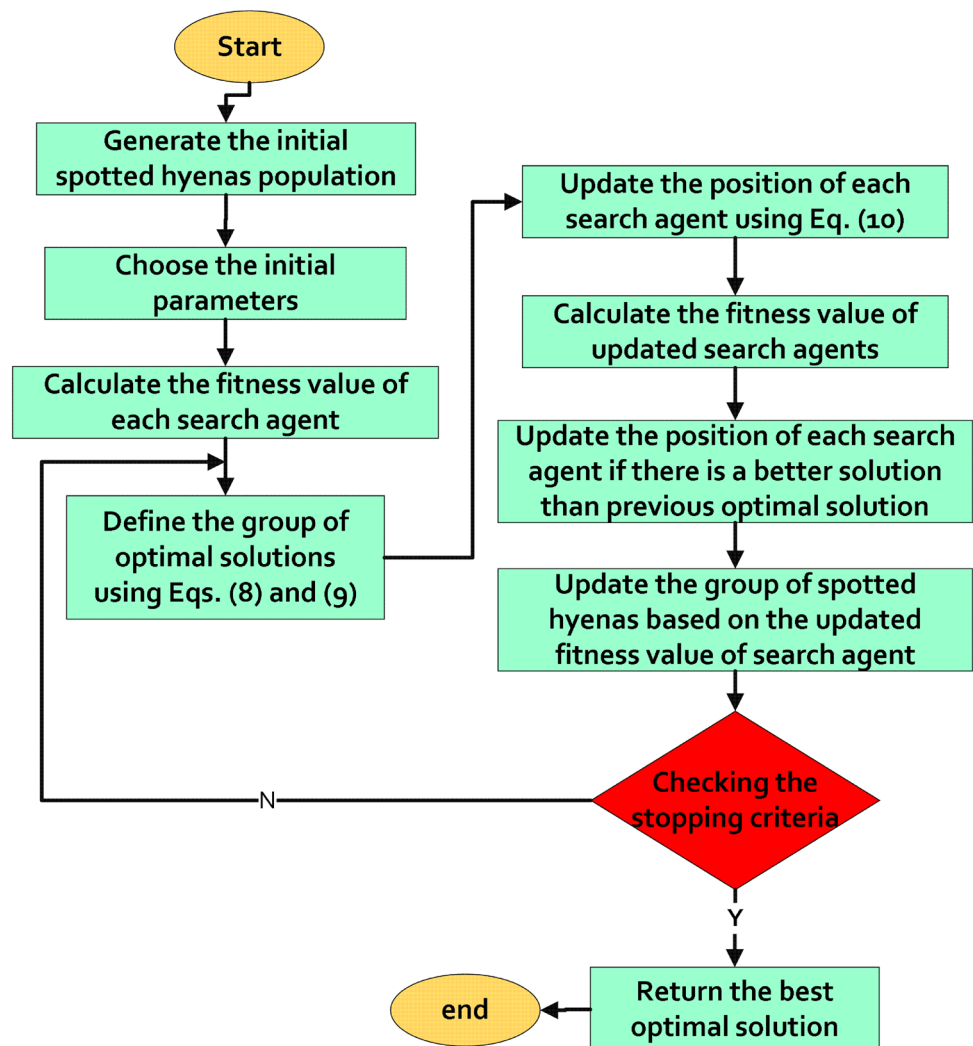
- Throughout the iteration, the SHO maintains the best solutions achieved so far.
- The SHO encircling technique describes circle-shaped whereabouts around the solutions, generalized to large dimensions as a hyper-sphere.
- Random vectors \vec{B} and \vec{E} assist selective solutions to have hyper-spheres with various random positions.
- The SHO hunting method allows selective solutions to place the probable location of the prey.
- The inclination of extraction and exploration by the corrected values of the vectors \vec{E} and \vec{h} Moreover, it allows

SHO to change over between exploration and extraction easily.

- half of the iterations are assigned to searching (exploration) ($|E| \geq 1$) by vector \vec{E} , and the other half are assigned to hunting (exploitation) ($|E| \leq 1$).

The analyses of research and assessments [39] demonstrated that the SHO algorithm is more accurate than the PSO, MFO, GWO, MVO, SCA, Gravitational Search Algorithm (GSA), Genetic Algorithm (GA), and Harmony Search (HS). Also, the SHO algorithm’s environment complexity is the maximum amount of environment used at any time, taking into account during the early stages. Therefore, the overall environment complexity of the SHO algorithm is equal to $O(n \times \text{dim})$.

Fig. 11 SHO algorithm flow-chart [39]



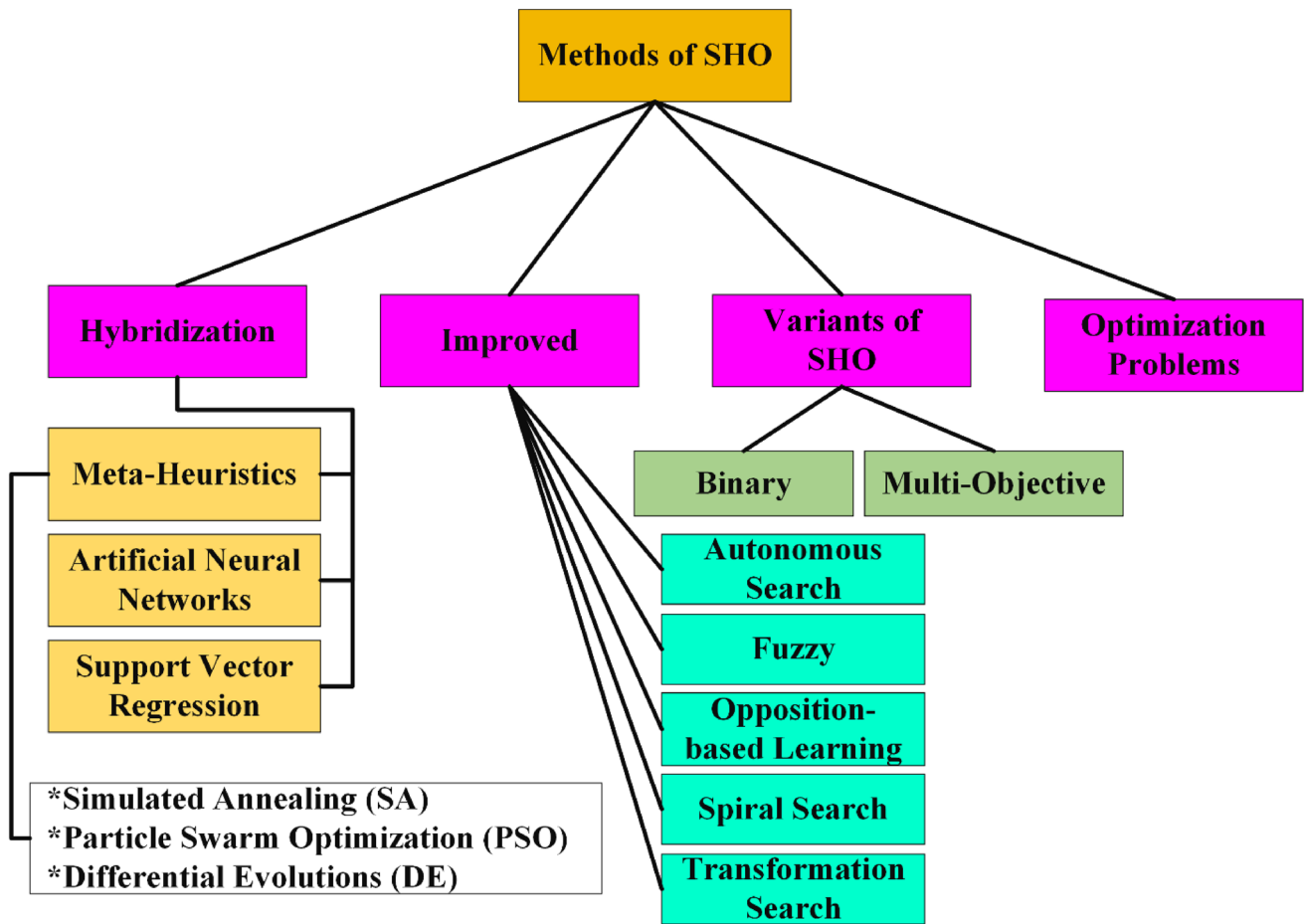


Fig. 12 Classification of SHO methods

3 Methods of SHO

Figure 12 shows how SHO methods are categorized based on hybridization, improvement, SHO variants, and optimization problems. Hybridization includes three subcategories: meta-heuristics, Artificial Neural Networks (ANNs), and Support Vector Regression (SVR). In the improved category, different subcategories are used to improve the solutions. Variants of SHO include binary and multi-objective subcategories. Finally, in optimization problems, the SHO is applied to solve various optimization issues to find the best solution.

3.1 Hybridization

Hybrid algorithms play an essential role in improving the problem search environment capability. In hybridization, any fundamental weakness is minimized using a combination of the advantages of two or more simultaneous algorithms. In general, hybrid models improve factors such as computational speed or accuracy. This section describes how to

combine the SHO algorithm with meta-heuristics, ANNs, and support vector regression.

3.1.1 Meta-Heuristics

Using SHO for Feature Selection (FS) issue designed two dissimilar hybrid models [40]. SHO can discover the optimal or nearly optimal feature subset in the feature environment for minimizing the given adaptability function. The optimal solution discovered by the SHO algorithm following each iteration is boosted through embedding the Simulated Annealing (SA) algorithm in the SHO algorithm (called SHOSA-1) in the first model. SA boosts the ultimate solution achieved by the SHO algorithm (called SHOSA-2) in the second model. These methods are assessed in terms of their performance on 20 datasets in the UCI repository. According to the analyses of the assessments, SHOSA-1 outperformed the native algorithm and SHOSA-2. SHOSA-1 is compared with six ultramodern optimization algorithms. The analyses demonstrated that the categorization accuracy

was enhanced by SHOSA-1, decreasing the selected features compared to other wrapper-based optimization models.

In [41], a new PSO and SHO-based hybrid metaheuristic optimization approach are called Hybrid PSO and SHO (HPSSHO). This algorithm boosts SHO's hunting strategy through PSO primarily. Compared with four metaheuristic algorithms (i.e., SHO, DE, PSO, and GA), the HPSSHO is compared to 13 commonly recognized exemplar complex functions, including unimodal and multimodal ones. HPSSHO's convergence analysis and other metaheuristics were conducted and compared. HPSSHO's applicability is tested by evaluating it on the issue of 25-bar actual constraint engineering design. HPSSHO outperformed other metaheuristic algorithms, according to the study.

In [42] introduced a new hybrid metaheuristic optimization algorithm according to Differential Evolution (DE) and recently developed SHO named Hybrid DE and SHO (HDESHO). HDESHO enhances the mutation strategy of DE through the SHO algorithm. Following the first gene selection, different machine learning algorithms were used to classify cancer. According to the analyses, HDESHO performed better compared to other methods.

In [43], a hybrid algorithm is designed based on the Sine-Cosine functions and attacking method of SHO. The SHO algorithm is used to identify the best possible solutions to real-world issues. Better exploration and exploitation are due to the SCA and attacking approach of the SHO algorithm. Experiments on welded beam design, tension/compression spring design, pressure vessel design, multiple disk clutch brake design, gear train design, and car side crashworthiness were conducted using IEEE CEC' 17 and six real-life engineering issues. The outcomes of the experiments were based on IEEE CEC' 17 and six real-world engineering challenges, including welded beam design, tension/compression spring design, pressure vessel design, multiple disk clutch brake design, gear train design, and automobile side crashworthiness. In contrast to previous competitor techniques, the suggested SSC algorithm proved robustness, effectiveness, efficiency, and convergence analysis.

3.1.2 ANNs

In [44], introduced SHO-Training Feedforward Neural Network (SHO-FNN) model according to SHO to train FNN. Eight standard datasets, including five categorization datasets (XOR, Balloon, Iris, Breast Cancer, and Heart) and three proper functions (Sigmoid, Sine, and Cosine), were applied to assess the performance of SHO to train FNNs. Based on the SHO algorithm's mathematical model, finding the appropriate prey, $|E| > 1$ eases the spotted hyenas from moving away from the prey. Exploring the search environment in this way helps find different FNN structures during optimization. The outcomes of the recommended SHO-FNN method

are compared to other population-based optimization algorithms in the literature, including GWO, ABC, GGSA, PSO, ACO, HS, DE, GA, and ES PBIL. According to the analyses, SHO-FNN can efficiently train FNN regarding exploration and avoidance of local minima. Furthermore, in terms of accuracy and rate of convergence, SHO was conducted for training FNN. SHO-FNN has achieved better analyses on the majority of datasets in terms of the accuracy of categorization. SHO-FNN improved the issue of local minima trapping with a significant convergence rate.

In [45] investigated the optimal modification of heating, ventilating, and air conditioning (HVAC) systems embedded in residential buildings through predicting Heating Load (HL) and cooling load (CL) that was done using four smart meta-heuristic algorithms, namely wind-driven optimization (WDO), WOA, SHO, and SSA. Algorithms are applied for Multi-Layer Perceptron (MLP) for tackling the computational inadequacies. These algorithms are applied to regulating the weight of neurons. The used dataset covers overall height, glazing area, relative compactness, orientation, glazing area distribution, wall area, roof area, surface area as autonomous factors, and the HL and CL as target factors. According to the analyses, all four meta-heuristic ensembles can understand the non-linear relationship between the mentioned factors. In the meantime, a comparison between the used models demonstrated that SSA-MLP (Error-HL = 1.9178 and Error-CL = 2.1830) is the most accurate model next to WDO-MLP (Error-HL = 1.9863 and Error-CL = 2.2424), WOA-MLP (Error-CL = 2.5390 and Error-HL = 2.1921), and SHO-MLP (Error-CL = 4.5930 and Error-HL = 3.1092).

For simulating Soil Shear Strength (SSS), two new hybrid models are recommended as the amalgamation of ANN, coupled with SHO and ALO meta-heuristic methods [46]. It is tried to neutralize this model's computational disadvantages by applying these algorithms to the ANN. As a function of ten key factors of the soil (including the percentage of sand, depth of the sample, percentage of loam, percentage of moisture content, percentage of clay, liquid limit, wet density, plastic limit, plastic Index, and liquidity index), the SSS was set as the response variable. The best-fitted structures were set by a trial and error process after developing the ALO-ANN and SHO-ANN ensembles. According to the analyses, both applied algorithms were efficient as the division error of the ANN was decreased by around 35% and 18% by the ALO and SHO, in the respective order. Comparing the analyses demonstrated that the ALO-ANN (Correlation = 0.9348 and Error = 0.0619) is more efficient compared to the SHO-ANN (Error = 0.0874 and Correlation = 0.8866).

In [47], authors suggested ALO-MLP and SHO-MLP models for the prediction of the F_{ult} placed in layered soils. Models are implemented on 16 databases of soil layer (soft soil placed onto more rich soil and vice versa), considering

a database covering 703 testing and 2810 training datasets to prepare the training testing datasets. Through a trial and process, the independent factors in terms of ALO and SHO algorithms underwent optimization. The input data layers include (I) upper layer foundation/thickness width (h/B) ratio, (II) bottom and topsoil layer properties, (III) vertical settlement (s), (IV) footing width (B), where the main target was taken F_{ult} . According to R2 and RMSE, values of (0.034 and 0.996) and (0.044 and 0.994) are achieved to train the dataset and values of (0.040 and 0.994) and (0.050 and 0.991) are found for the testing dataset of suggested SHO–MLP and ALO–MLP best-fit prediction network structures, in the respective order. With a shallow circular foot carrying capacity, the SHO-MLP was highly durable.

In [48] used SHO to train MLP. Choosing parameters is believed to be the critical aspect while training the MLP using a meta-heuristic algorithm. The essential parameters for MLP, i.e., weights and biases, are firstly selected through randomization. Therefore, the trainer should supply a set of initial values to obtain optimum categorization accuracy. In the suggested work, a single hidden layer is used for designing the MLP network, and it was trained through the application of SHO. SHO-MLP can be assessed in terms of efficacy through fifteen primary datasets in which fourteen are chosen from the UCI Machine Learning Repository. According to the analyses, SHO proudly circumvented the local minima trap problem revealing accurate categorization compared to other meta-heuristic methods. According to the line diagram, SHO had faster convergence than DE, GA, PSO, SSA, and GWO.

In [49], They introduced SHO for Feedforward Neural Networks (FNNs). Training FNNs is not easy since it is easy to fall into local optima. FNNs are trained in finding the best weight and bias value. The vector of FNN is shown according to Eq. (11)

$$\vec{V} = \{ \vec{W}, \vec{\theta} \} = \{ W_{1,1}, W_{1,2}, \dots, W_{n,n}, h, \theta_1, \theta_2, \dots, \theta_h \} \quad (11)$$

In Eq. (11), the number of the input nodes is determined by n , W_{ij} signifies the connection weight from the i^{th} node to the j^{th} node, θ_h is the bias (threshold) of the j^{th} middle node. The analyses of the suggested SHO-FNN are compared to the other population-based optimization algorithm such as GSA, GWO, HS, and DE. According to the analyses, SHO-FNN can efficiently train FNN regarding the exploration and avoidance of local minima. Furthermore, SHO performed extensively in training FNNs concerning analyses and convergence rate accuracy given the fitness value.

3.1.3 Support Vector Regression

In [50] used six meta-heuristic algorithms, namely, Multi-Verse Optimizer (SVR-MVO), Ant Lion Optimization

(SVR-ALO), Harris Hawks Optimization (SVR-HHO), (SVR-SHO), SVR-PSO, and Bayesian Optimization (SVR-BO) through the optimization of support vector regression (SVR) for predicting daily streamflow in Naula watershed, State of Uttarakhand, India. First, the SVR optimization was done by applying six meta-heuristic models (i.e., MVO, HHO, ALO, SHO, PSO, and BO). Then, practically speaking, the non-linear components and the number of input factors are significant for sturdy predicting. Gamma Test is applied for extracting the remarkable inputs and factor amalgamations for hybrid SVR methods before processing. Thus, the Gamma Test (GT) is applied for detecting the best combinations of input variables. The analyses of hybrid SVR methods during calibration (training) and validation (testing) periods were compared with observed streamflow using accomplishment indicators of scattering index (SI), root means square error (RMSE), Willmott index (WI), coefficient of correlation (COC), and by visual examination (time-series plot, scatter plot and Taylor diagram). According to the analyses, SVR-HHO during calibration/validation periods (SI = 0.401/0.715, RMSE = 92.038/181.306 m³/s, WI = 0.928/0.777, and COC = 0.881/0.717) outperformed the SVR-ALO, SVR-SHO, SVR-MVO, SVR-BO, and SVR-PSO models in forecasting daily streamflow in the study basin. Furthermore, the HHO algorithm performed better compared to the other meta-heuristic algorithms regarding prediction accuracy. The SVR-SHO model outdid SVR-PSO, SVR-MVO, and SVR-BO.

The SVR can learn and model non-linear relationships between input and output data in a higher dimension, thus minimizing the observed error in training or distribution proportionally to generalized revenue. The benefits of SVR are as follows:

- Definite convergence to optimal solutions because of using quadratic programming with linear constraints for learning the data.
- Computational efficiency for non-linear relationship modeling by applying kernel mapping.
- A predominant global accomplishment (lower error rates on the test set).

SVR is computationally efficient that depends on several hyperparameters affecting finding the optimal solution both directly or indirectly.

By utilizing EDI (Effective Drought Index), the goal of [51] was to investigate the possible capacity of support vector regression (SVR) combined with GWO and SHO for meteorological drought (MD) prediction (effective drought index). The two-hybrid SVR–GWO and SVR–SHO models, were built in Uttarakhand's Kumaon and Garhwal areas to achieve this goal (India). The EDI was calculated in both

research zones by calibrating and validating the advanced hybrid SVR models using monthly rainfall data. The auto-correlation function (ACF) and partial-ACF (PACF) were used to establish the ideal inputs (antecedent EDI) for EDI prediction. The outcomes of the hybrid SVR models were compared to the estimated (observed) values using statistical indicators and visual examination. According to the results, for all research stations in the Kumaon and Garhwal areas, the hybrid SVR–GWO model outperformed the SVR–SHO model.

Table 1 analyzes the various SHO models in the field of hybridization (metaheuristic algorithms, ANNs, and SVRs).

Figure 13 shows the percentage diagram that the SHO algorithm combines with meta-heuristics, ANNs, and SVR models. As shown in Fig. 13, the percentages of

meta-heuristics, ANNs, and SVR models, combined with SHO, were 30%, 60%, and 10%, respectively. Accordingly, the percentage of ANNs used in combination with SHO is higher than the other two models.

3.2 Improved

As the complexity of real-world optimization problems increases, more robust optimization methods are needed to solve such problems. In SHO, the optimization process is weakened in the extraction phase. Therefore, extensive research was conducted to establish a good trade-off between exploration and extraction to improve solutions. This section addresses research on a trade-off between exploration and extraction.

Table 1 Analysis of different SHO models in the field of hybridization

References	Methods	Problems	Advantages	Disadvantages	Comparison
[40]	SHOSA-1	Feature selection	Diversification Global search capability Good convergence acceleration	Random movement	ACO, SSA, FPA, PSO, SCA and CS
	SHOSA-2	Continuous optimization issues	Optimal solution	Unbalanced exploration	ACO, SSA, FPA, PSO, SCA and CS
[41]	HPSSHO	Categorization	Extraction/intensification search strategies	Slow convergence rate	PSO, DE, and GA
[42]	HDESHO	Global optimization	Good balance between extraction and exploration	Trap of local optima	DE, PSO
[43]	SCA-SHO	Engineering optimization problems	Optimal solution	Random movement	SCA, PSO
[44]	SHO-FNN	Engineering optimization problems	Training FNN Faster convergence rate Optimal solutions Local optima avoidance	High execution time	GWO, PSO, GA, ACO
[45]	SHO-MLP	Engineering optimization problems	Avoid local solutions	High environment complexity	WDO-MLP, WOA-MLP, and SSA-MLP
[46]	SHO-ANN	Engineering optimization problems	Training ANN	High execution time	ALO
[47]	SHO-MLP	Optimization	Global minimum Higher performance outcome	High environment complexity	ALO
[48]	SHO-MLP	Optimization	Training MLP Avoid local solutions Higher exploration and extraction capability	High cost	DE, GA, PSO, GWO, SSA, ANN
[49]	SHO-FNN	Engineering optimization problems	Avoiding local minima Fast convergence Good exploration	High iterations	GWO-FNN, HS-FNN, GGSA-FNN, DE-FNN
[50]	SVR-SHO	Prediction	A good balance between exploration and extraction	High execution time	SVR-ALO, SVR-MVO, SVR-HHO, SVR-PSO
[51]	SVR-SHO	Meteorological drought (MD) prediction	Fast convergence Good exploration	High iterations	GWO, PSO, GA

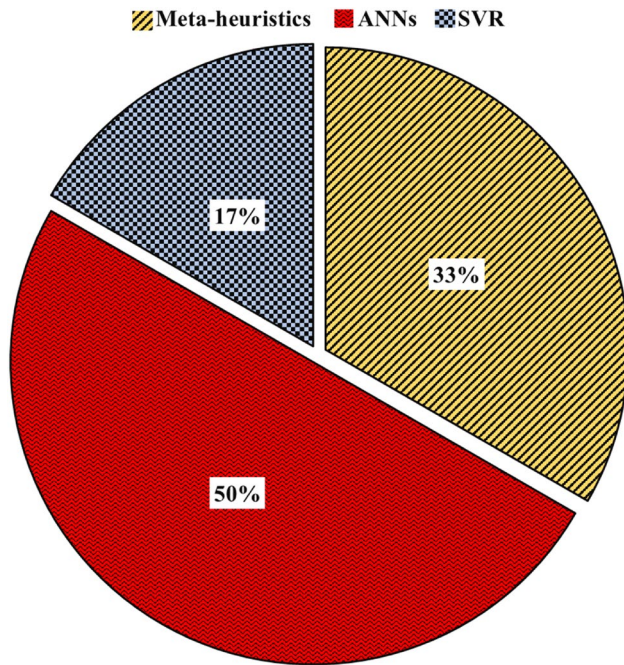


Fig. 13 Percentage diagram of the SHO combined with meta-heuristics, ANNs, and SVR models

3.2.1 Autonomous Search

The Set Covering Problem (SCP) is a significant combinatorial optimization problem. It has been used in many practical areas, chiefly in airline crew scheduling or vehicle routing and facility placement problems. To solved SCP employed the SHO. In [52] used Autonomous Search (AS) for enhancing the performance of SHO through adjusting the parameters, either by self- matching or supervised matching. The population of spotted hyenas was adjusted if any of the following three criteria are met:

- Increasing the population evenly: If the subtraction of fitness values between the best and worst spotted hyena is equal to 0, it has increased the population by five and randomly spotted hyenas.
- Increase population by variation: If the factor variable is less than 0, 3, high cohesion between the spotted hyenas will increase the population by Encircling prey and producing randomly spotted hyenas.
- Decrease of the population: if two or more spotted hyenas have the same fitness, it will remove at least one spotted hyena and decline the population by the same amount.

According to the analyses, SHO is significantly robust with symmetric convergence.

3.2.2 Fuzzy

In [53] introduced Automatic generation control (AGC) of a three-area power system aided with a *FuzzyPID* controller. In each area, conventional sources like thermal, gas, hydro, and energy system are used for injecting wind energy sources. In a wind farm, the inertia and droop control method is implemented, and the produced power is transacted through the HVDC link. For other familiar sources, *FuzzyPID* controller is used as the peripheral controller. Therefore, this controller manages the system's transient response for reaching the system's steady-state at the appointed time with a less steady-state error. The analyses achieved by the PID control structure are compared to explore the potential of this controller.

Moreover, the abrupt load change of high vastness is injected in the first area for analyzing the controller sensitivity. Therefore, to set the proper controller parameter for boosting the system accomplishment, an efficient optimization algorithm is advised. This work used a recently improved SHO computational algorithm to determine the controller.

3.2.3 Opposition-based Learning

Opposition-based learning (OBL) is mainly used to generate an initial population to increase diversity. In optimization problems, it is necessary to adopt an appropriate method to maintain population diversity due to the continually changing nature of solutions. Thus, the diversity of population members is guaranteed, and OBL improves the SHO algorithm's efficiency. Among the essential benefits of OBL are:

- Move-in the initial iterations from exploration to extraction
- Avoid getting stuck in the local optimum
- Provide a search environment to evaluate all solutions

In [54] introduced an improved SHO (ISHO) with a non-linear convergence factor for Proportional Integral Derivative (PID) parameter optimization in an Automatic Voltage Regulator (AVR). In ISHO, an OBL method is applied for setting the spotted hyena agents' position in the search environment that reinforces the diversity of agents in the global searching process. SHO is boosted in extraction and exploration abilities by applying a new non-linear update equation for the convergence rate. According to the analyses, the ISHO algorithm outperformed other models regarding the solution precision and convergence rate.

In [55] introduced OBL-MO-SHO to boost the existing SHO in terms of performance. The oppositional learning concept was mixed with the mutation operator to deal with more intricate realistic issues and boost the SHO exploitative

and explorative strength. The OBL-MO-SHO (oppositional SHO with mutation operator) performed well regarding reaching the global optimum and superior convergence rate verifying its enhanced extraction and exploration ability within the searching area. Setting the competency of proposed OBLMO-SHO, the same is appraised using main functions set belongs to IEEE CEC 2017. Several performance metrics were applied, and the outcomes were compared with ultramodern algorithms to confirm the method above in terms of efficacy. Friedman and Holms’s test was run as one non-parametric test to examine its uniqueness in statistical terms. Moreover, the afore-mentioned OBL-MO-SHO, as an application for sorting out complicated real-world difficulties, has been cast off for training the wavelet neural network by considering datasets selected from the UCI repository. According to the assessment, the developed OBL-MO-SHO might be one potential algorithm to identify different optimization difficulties effectively.

3.2.4 Spiral Search

In this method, all spotted hyenas in the group can benefit from other spotted hyenas’ best experiences for problem environment dimensions instead of following the best-spotted hyena. Also, the extraction and extraction rates of mottled hyenas are determined by the learning rate. Therefore, the SHO algorithm utilizes this strategy for the beneficial use of exploration and extraction.

In [56] introduced the revised version of SHO to boost performance by applying spiral moment and astrophysics concepts in the proposed model. SHO intensification capability of SHO is boosted by applying the spiral moment. The diversification and intensification are both enhanced through the incorporation concept in SHO. The proposed model’s performance is compared with five commendable meta-heuristic algorithms over CEC 2015 exemplar intricate functions, and analyses demonstrated that the proposed algorithm performed better than the others in terms of the fitness function value. They also studied the impacts of scalability and sensitivity analysis. The proposed model is also applied to limited engineering design problems, and analyses demonstrated that the proposed model outperformed the other algorithms.

3.2.5 Transformation Search

It is possible to transfer the current search area to a different environment by applying the Environment Transformation Search (STS) [57] technique. Hence, it would be possible to guess the existent selective solutions in both the search regions simultaneously. Thus, better global optimal results are anticipated. In mathematical terms, STS can be described

as follows: let z be the solution in the current search region, X , and $z \in [x, y]$. The new solution z^* in the recently shifted environment can be determined as $z^* = \Delta - z$, where Δ is a countable value and $z^* \in [\Delta - y, \Delta - x]$. Accordingly, the search environment center is distinguished between the current search area z and the newly shifted search area z^* . Therefore, the focal for the current search environment and focal of the newly shifted search environment can be specified as $(x + y)/2$ and $(2\Delta - x - y)/2$ as well. Nevertheless, the diameter of current search environment and newly shifted search environment will not alter by transferring the search environment, and it remains the same value as $y - x$. Let $\Delta = r_1(p + q)$, where r_1 is a real number and p and q shows limits of the search part. Then, STS can be expressed as in Eq. (12)

$$z^* = r_1(p + q) - z \tag{12}$$

Which can also be expressed as

$$z_{ab}^* = r_1 [p_b(t) + q_b(t)] - z_{ab} \tag{13}$$

where $p_b(t) = \min(z_{ab}(t))$ and $q_b(t) = \max(z_{ab}(t))$, where t signifies the current iteration. Considering the value of r_1 , the STS method can be grouped into four methods:

- (1) $r_1 = 0$, STS-SS (There is an asymmetric solution in STS)
 - The method can be defined as
 - $z^* = -z$, where $z \in [p, q]$ and $z^* \in [-q, -p]$. Both z and z^* deceit on the symmetry of source.
- (2) $r_1 = 0.5$, STS-SI (There is the asymmetric distance in STS)
 - The method can be denoted as $z^* = \frac{p+q}{2} - z$, where $z \in [p, q]$ and $z^* = \left[-\frac{q-p}{2}, \frac{q-p}{2}\right]$, interim of z^* is at symmetry of origin.
- (3) $r_1 = 1$, STS-OBL
 - The model can be signified as:
 - $z^* = p + q - z$, where $z \in [p, q]$ and $z^* \in [p, q]$.
- (4) $r_1 = \text{rand} [0, 1]$, STS-R (STS is random)
 - The method can be signified as $z^* = r(p + q) - z$, r is defined as a random number within $[0, 1]$
 - $z \in [p, q]$ and $z^* \in [r(p + q) - q, r(p + q) - p]$. The center of the transferred search area lies in between $-\frac{q-p}{2}$ and $\frac{q-p}{2}$.

The improved version of classical SHO has been suggested to boost the search region’s explorative strength and

Table 2 Analysis of various improved SHO models

References	Methods	Problems	Advantages	Disadvantages	Comparison
[52]	SHO-AS	Solving the set covering problem	Optimal solution Speed of convergence	Unbalanced exploration	binary cat swarm optimization (BCSO)
[53]	SHO-Fuzzy	Optimal design and fuzzy logic-based controllers in order to LFC study in power system	Exploration and extraction capability	Without considering all search environment	–
[54]	SHO-OBL	PID parameters in an AVR system	Global solid search capability Convergence speed Balance between the global search and local search Optimization efficiency	Inappropriate variation in the initial population	SCA, FPA, PSO, GSA, GWO, SHO
[55]	OBL-MO-SHO	global optimization and training ANNs	Exploration and extraction capability Convergence speed	Inappropriate variation in the initial population	DE, GA, GWO, PSO, SHO
[56]	MSHO	Constraint Engineering Problems	Balance between extraction and exploration Optimal solution	High execution time	SHO, GWO, PSO, GSA and MVO
[58]	STS-SHO	categorization	Balance between extraction and exploration Exploring the search area and converges closer toward global optimum rapidly	Inappropriate variation in the initial population	PSO, GWO, MFO, MVO, SCA, GA, GSA

extraction; the proposed method is designated as STS-SHO [58]. A new revolutionary method called the STS method has been combined with the original SHO in STS-SHO. IEEE CEC 2017 exemplar problems have been used for the assessment of the suggested method. The standard measures such as given accomplishment metrics in CEC 2017, convergence analysis, complexity analysis, and statistical implications have been applied to verify the aforementioned method in terms of efficacy. Moreover, as a real-world application, the aforementioned algorithm was applied for training the pi-sigma neural network through 13 exemplar datasets determined from the UCI repository. The paper can be terminated that STS-SHO's recommended method is an operative and trustworthy method capable of resolving actual optimization complications. Table 2 analyzes the various improved SHO models.

3.3 Variants of SHO

3.3.1 Binary

The Binary Optimization Problem (BOP) is demonstrated as a binary-based issue environment showing a significant class of combinatorial optimization problems. In continuous optimization, search agents take continuous values in the search environment, while in binary optimization, search agents in

the search environment take {0,1} values. "0" represents false, and "1" signifies true.

In [59], they suggest a new binary model of SHO (BSHO) imitating spotted hyenas' hunting behavior in the binary search environment. The BSHO used the tangent function for mapping the continuous search environment into a binary search environment. The BSHO is applied as a wrapper approach for Feature Selection (FS). FS mainly aims to find a small subset of features from a remarkable main set of features optimizing the categorization precision. Since FS chiefly aims to keep down the number of selected features while save or increasing the maximum categorization accuracy, it can be described as an optimization task. Eleven datasets from the UCI machine learning repository are applied for comparing the BSHO with six well-known binary metaheuristic techniques, namely Binary GWO (BGWO), Binary GSA, Binary PSO, Binary Bat Algorithm (BBA), Binary WOA (BWOA), and binary Dragonfly Algorithm (BDA). The maximum number of iteration and population size of the mentioned algorithms are set to 100 and 30, in the respective order. The assessment achieved convergence curves from BGWO, BSHO, BGSA, BBA, BPSO, BWOA, and BDA. BSHO has explored the most promising area of the search environment.

In BSHO, the float-encoding layout of SHO is altered with a binary encoding layout since each variable's

solutions are limited to 0 and 1. The position updating method of spotted hyenas has been suggested to enhance the local search capability to find an improved solution in the binary search environment. To this aim, the hyperbolic tangent function is applied to update spotted hyenas based on the optimal solution. In BSHO, the positions are changed between "0" and "1" only. Therefore, the dimension of the search environment lies in the domain of 0 or 1. BSHO and another binary version of the meta-heuristic are mainly different in terms of group formation mechanisms. The group formation is mathematically represented according to Eq. (14) [59].

$$\vec{C}_h = \vec{P}_k + \vec{P}_{k+1} + \dots + \vec{P}_{k+N} \tag{14}$$

In Eq. (15) \vec{C}_h is the group of optimal solutions achieved from the proposed method.

$$x_d^{s+1} = \begin{cases} 1, & T(\vec{C}_h) \geq rand \\ 0, & \text{Otherwise} \end{cases} \tag{15}$$

$$T(\vec{C}_h) = \tanh(\vec{C}_h) = \frac{\left(\exp^{-\tau(\vec{C}_h)} - 1 \right)}{\left(\exp^{-\tau(\vec{C}_h)} + 1 \right)} \tag{16}$$

In Eq. (15), a *rand* is a random number taken from a uniform production [0,1]. x_d^{s+1} is the binary position of agents, d is the dimension, s is the iteration number, and τ is 1. Figure 14 shows nine features f1 to f9. The binary values in the vector \vec{C}_h show that f1, f3, f4, f5, f8, and f9 are chosen, and the features f2, f6, and f7 remain unchosen.

For handling distributed job shop scheduling (DJSP) problems has been proposed a discrete version of the spotted hyena optimizer (DSHO) [60]. The DSHO method is integrated with a workload-based facility order mechanism and a greedy heuristic technique. DSHO has solved 80 DJSP. The numerical results of the 480 (2 to 7 facilities) significant instances produced from well-known JSP benchmarks are compared with four alternative discrete meta-heuristic techniques to evaluate the performance of the DSHO. DSHO is a pioneer solver for DJSP, according to the experimental data.

FS	F1	F2	F3	F4	F5	F6	F7	F8	F9
\vec{C}_h	1	0	1	1	1	0	0	1	1

Fig. 14 BSHO position encoding

3.3.2 MOP

MOP denotes an issue with more than one objective function [61]. Solving this problem requires finding the best trade-offs between all objectives. The mathematical model of a minimization problem is as Eq. (17).

$$\text{Minimize } F(x) = \{f_1(x), f_2(x), f_3(x), f_4(x), \dots, f(x)\}, m \geq 2 \tag{17}$$

$$\text{Subject to } nq_i(x) \geq 0, i = 1, 2, \dots, z \tag{18}$$

$$q_i(x) = 0, i = 1, 2, \dots, k \tag{19}$$

$$L_i \leq x_i \leq U_i, i = 1, 2, \dots, p \tag{20}$$

where m is the number of objectives, nq_i defines the i^{th} inequality constraint and z define the number of inequality constraints. q_i shows the i^{th} equality constraint and k describe the number of equality constraints. p is the number of variables. L_i and U_i are the lower and upper bounds of the search environment for the i^{th} variable (x_i).

In [62], they have combined the properties of Multi-objective SHO (MOSHO), SSA, and EPO to introduce a novel hybrid optimization algorithm called MOSHO, SSA, and EPO (MOSSE). MOSSE uses MOSHO's searching capabilities for discovering the search environment, SSA's leading and selection process in an effective manner to achieve the fittest global solution with a faster convergence method and EPO's effective mover method for better regulation of the next solution. MOSHO is more substantial in terms of optimization capabilities, and it can eliminate the problem of missing selection pressure. The method is implemented on ten IEEE CEC-9 original test functions compared with seven prominent MOP algorithms in terms of their accomplishment. According to the results, MOSSE outcomes are highly reasonable regarding searchability, accuracy, and convergence speed. Technical testing is also done on IEEE CEC-9's intricate functions. MOSSE is further applied to welded beam, pressure vessel, multi-disk clutch brake, and 25-bar truss design issues to examine its efficiency. According to the analyses, the proposed algorithm is capable of resolving actual, complex MOP problems.

In [63], a new hybrid multiple-objective algorithm (MOA) named HMOSHSSA was introduced, combined with the MOSHO and SSA. The SSA leader and selection process is used by MOSHO to quickly scan the questing environment and identify the genius best option for faster conversion. The proposed algorithm is evaluated on 24 complex exemplary functions and compared to 7 well-known MOO algorithms. According to the data, HMOSHSSA produces very striking results and performs better in convergence speed, searchability, and accuracy than other models. Moreover,

HMOSSHSA is also applied to seven well-known engineering issues to confirm its effectiveness more. Based on the results, the efficacy of the proposed algorithm for solving actual MOP issues is confirmed.

In [64] proposed the RWEKA package offering an interface of Weka tool functionality to R to order the features using the select attribute function in WEKA. Applying those ordered features to choose a minimal subset of features using an SVM classifier with maximum prediction precision in the dataset is possible. An achieved minimal subset of features is given as input to the MOSHO algorithm driven by the SVM classifier's ensemble by updating the search agents with objective function for boosting the categorization precision. The suggested method has been tested with seven openly accessible microarray datasets such as CNS, colon, leukemia, lymphoma, lung, MLL, and SRBCT, showing that the suggested methodology is highly accurate with other prevailing techniques regarding feature selection and prediction accuracy. The prediction accuracy enhancement was achieved, ranging from 4 to 5% increases than the proposed approach. Microarray data analysis is an important research area in medical study. The Microarray is a dataset covering various gene expressions in which most of the features are redundant genes decreasing the classifier precision. It is not easy to find a minimal subset of extensive gene expression features, eliminating redundant features, but the significant feature is not neglected. The researchers introduce several optimization methods for identifying a minimal subset of features, but it does not offer a viable solution.

In [65], both convex and non-convex economic dispatch and microgrid power dispatch problems by introducing a novel hybrid MOA called MOSHO and Emperor Penguin Optimizer (MOSHEPO). The suggested MOSHEPO combines the two recently established bio-inspired optimization algorithms: MOSHO and EPO. MOSHEPO anticipates many non-linear specifications of power generators such as multiple fuels, dissemination losses, prohibited operating zones, and valve-point loading, as well as their operational limits for practical operation. Numerous exemplar test systems have been applied to examine the suggested algorithm, and its performance was compared with other renowned methods for assessing MOSHEPO in terms of efficiency. According to the results, the suggested MOSHEPO performed better than other models with low power-time efforts while solving microgrid power dispatch issues and economics.

In [66] suggested C-HMOSSHSA for gene selection using MOSHO and SSA. Keeping convergence and diversity is not easy for actual optimization problems with more than one objective. SSA sustains diversity; however, it must face the overhead of sustaining the necessary information. In contrast, the calculation of MOSHO is done with less computational effort and is applied for maintaining the essential

information. Consequently, the suggested algorithm is a hybrid algorithm applying the properties of both SSA and MOSHO for easing its exploration and extraction competence. Emulations are done using seven various microarray datasets for assessing the accomplishment of the suggested method. According to the results, the proposed approach is compared with existing ultramodern methods; according to the results, the proposed model performed better than the existing methods.

In [67], the authors introduced a MOO version of a recently amplified SHO called MOSHO for engineering issues. In the proposed method, a fixed-sized archive is used to keep the non-dominated Pareto ideal solutions. The archive's effective solutions are chosen using a roulette wheel mechanism for simulating the social and hunting performances of spotted hyenas. The proposed method is tested on 24 intricate exemplar functions and compared with six recently developed metaheuristic models. The suggested algorithm is then applied to six problems of limited engineering design to test its applicability to actual problems. According to the analyses, the suggested algorithm outperformed the others, leading to optimal Pareto optimal solutions with high convergence.

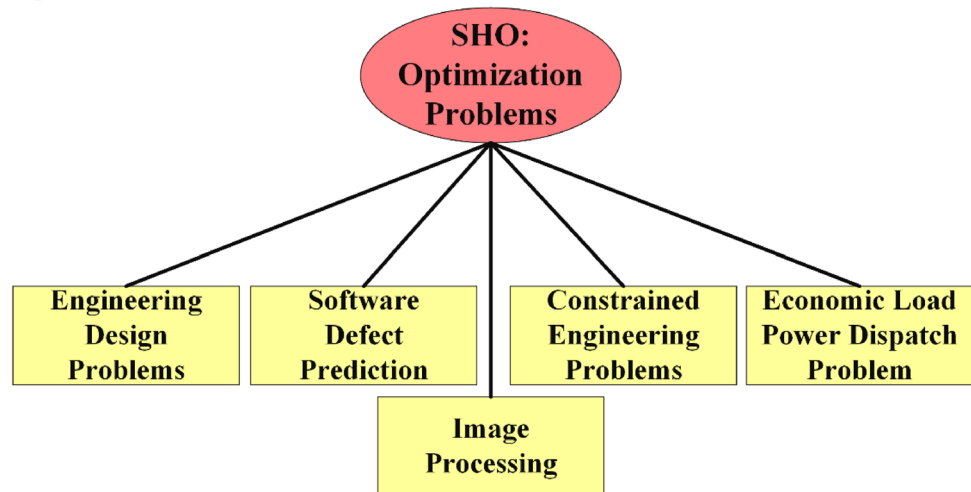
- (1) Initializing MOSHO population necessities $O(n_o \times n_p)$ time where n_o signifies the number of objectives and n_p signifies the number of individual sizes.
- (2) The adaptability computation of each search agent needs $O(Max_{iterations} \times n_o \times n_p)$ time where $Max_{iterations}$ is the maximum number of iterations to simulate the MOSHO.
- (3) The algorithm needs $O(M)$ time for determining the group of spotted hyenas, where M specifies the counting value of spotted hyenas.
- (4) It necessitates $O(n_o \times (n_{ns} + n_p))$ time for updating the archive of non-dominated solutions. The evidence is a storage section of all non-dominated Pareto optimal solutions achieved so far.
- (5) Repeat Steps 2 to 4 until the dissolution standards are acceptable.

3.4 Optimization Problems

The SHO is used to solve various optimization issues [68–72]. This algorithm has been proven to be able to solve most problems. Figure 15 shows a diagram of the applications of the SHO algorithm in solving optimization problems.

In [73] compared the WOA, the ALO, and the SHO algorithm (SHOA) to show how the optimization methods have been used for reaching weight reduction in an automobile brake pedal while keeping stress requirements. Optimization

Fig. 15 Applications of SHO algorithm in solving optimization problems



through SHO results in a decrease of 18.1% of brake pedal weight.

In [74] introduced a probable solution for cross-projects Software Defect Prediction (SDP) using the SHO algorithm as a classifier. The SHO is trained as a classifier on one dataset of projects, and resultant categorization rules are applied on various projects to set the categorization accuracy. SUP and CONF are used as multi-objective fitness functions for identifying appropriate categorization rules. According to the analyses, the SHO classifier outperformed other methods with an average accuracy of 86.4%. Furthermore, precision, sensitivity, specificity, recall, and F-measure are computed for the SHO classifier compared with the other techniques in WEKA 3.6. Assessments confirmed that the SHO classifier could efficiently anticipate software defects through cross-projects.

In [75], we conceptually compared SHO, GWO, PSO, ACO, GSA, BA, MFO, and WOA. In terms of behavior, these algorithms are modeled in mathematical terms to represent the optimization process. Twenty-three intricate exemplar functions are applied for confirming the performance of these algorithms. The convergence curve is used to analyze these algorithms in terms of exploration and extraction. Based on the analyses, SHO and GWO produced the optimal solutions in comparison with the other algorithms. Additionally, these algorithms are tested on five problems of limited engineering design, and the results verified the applicability of these algorithms in actual engineering design problems.

In [76] introduced a hybrid SHO based on lateral inhibition, which has been used to solve convolution image matching issues. The lateral inhibition mechanism is used for the image pre-process to boost the intensity gradient in the image contrast and boost the image's characters, enhancing image matching precision. In SHO, the fitness calculation method is incorporated to solve actual

optimization problems leading to a drastic reduction of the computation search location. The proposed LI-SHO blends the benefits of SHO and lateral inhibition mechanism for image matching. According to the analyses, the suggested LI-SHO based on lateral inhibition outperformed another algorithm for image matching more effectively and feasibly.

In [77], they introduced SHO for solving complex and non-linear limited engineering issues. The basis of this algorithm is the hunting strategy of spotted hyenas in nature. The three necessary steps of suggested SHO are searching, encircling, and attacking for prey. The suggested SHO is used to two actual complex and non-linear limited engineering issues for ensuring its applicability in a high-dimensional environment. According to engineering problems, SHO performed better compared to other competitive approaches.

In [78], they applied SHO on two actual problems of limited engineering design, such as optical buffer design and airfoil design problems and the results are compared with other competitor algorithms. According to the analyses of engineering design problems, the SHO is a useful optimizer for solving these problems and generating near-optimal designs.

In [79], suggested SHO solves both convex and non-convex economic dispatch problems. The suggested algorithm has been tested on various test systems (i.e., 6, 10, 20, and 40 generator systems) compared with other well-known approaches to test it in terms of effectiveness and efficiency. According to the analyses, SHO can solve economic load power dispatch problems and converges toward the optimum with less computational effort.

In [80], SHO proposed two actual problems of limited engineering design such as 25-bar truss design, innumerable disk clutch brake configuration problems and compared it with other competitor algorithms. According to the analyses, SHO is an efficient optimizer for analyzing extraction and exploration. Besides, the applicability of SHO in a high-dimensional

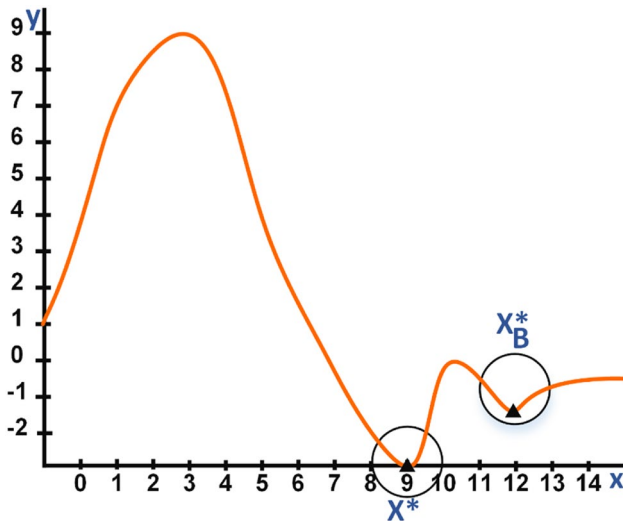


Fig. 16 Representation of global minimum and local minimum

environment with less computational effort has been confirmed according to engineering design problems.

4 Discussion

Metaheuristic algorithms cover two types of problems, maximization and minimization. Therefore, any maximization problem can become a minimization problem, and vice versa [81]. A minimization problem is defined based on Eqs. (21) and (22). The global and local minima are shown in Fig. 16. Here, S is the search environment, x^* is the global minimum, x_B^* is the local minimum, and $f(x^*)$ is the global minimum of the desired cost function based on the global minimum parameter.

$$f(x^*) \leq f(x), \quad \forall x \in S, \quad x^* \in S \tag{21}$$

$$f(x_B^*) \leq f(x), \quad \forall x \in B, \quad B \subset S \tag{22}$$

In metaheuristic algorithms, it is critical to understand the concepts of intensification and diversification. The former refers to using search experiences and the latter to solve the entire search environment's extensive exploration. In the SHO algorithm, a search is performed by the best-spotted hyena in the group. Moreover, the best spot is discovered by using its experiences. In other words, intensification is the exploration and more precise search for parts of the solution environment in which there is more hope of finding the optimal solution. As soon as the search process begins, it is necessary to calculate the value of different search environment points to find areas of hope (diversification). The algorithm then searches for more promising areas to find local optimum (intensification). Finally, the best local optimum is selected from different areas hoping that this solution is the same. Table 3 lists the advantages and disadvantages of the SHO algorithm.

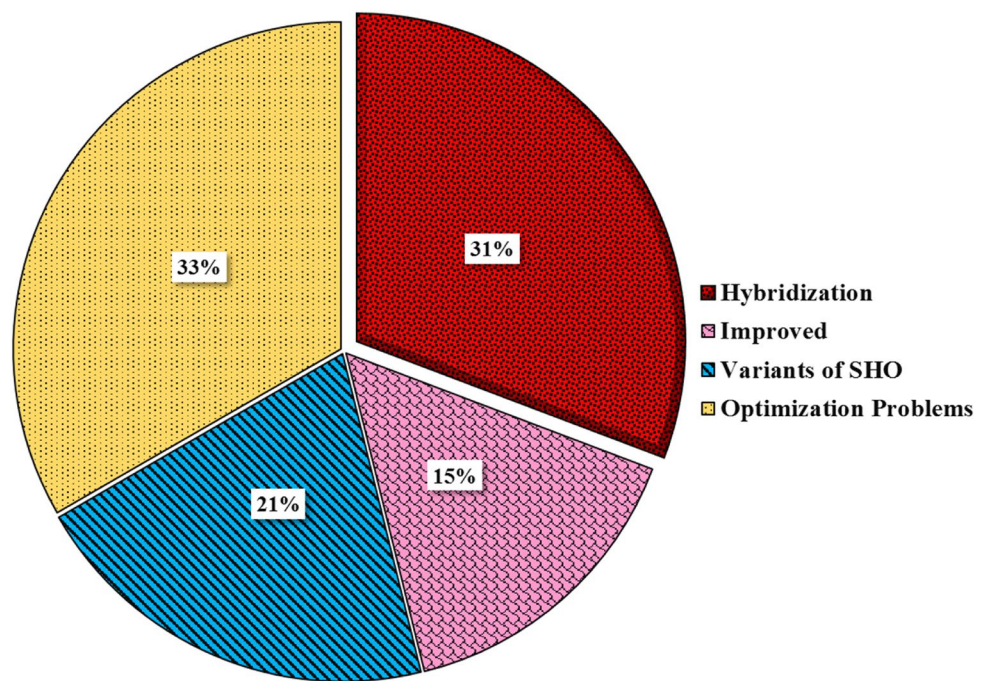
Various improvement methods should be used for SHO operators to prevent premature convergence. If the hyena population flows to one side, then early convergence occurs, and the objective function's optimization is not performed. Using variants is to stave off the stuck in local search and improve the exploration and exploitation of search capabilities.

Figure 17 shows a graph of the SHO algorithm's percentage combined with hybridization, improved variants, and optimization problems. As it turns out, the optimization problems have a higher percentage. A suitable solution may be to use combinatorial methods to overcome challenging problems. In hybrid models, a regulatory

Table 3 Advantages and disadvantages of SHO algorithm in general

Advantages/Disadvantages	Properties
Advantages	Simplicity and compatibility to combine with other algorithms Fast convergence rate Simple implementation the balance between exploration and extraction A search for all solutions Approaching the goal based on neighborhood Avoiding falling into local best solutions Suitable for multi-objective problems Vital in controlling people's decisions in the search environment Compatibility, robustness, and scalability are found critical capabilities Exploration of unknown search environments
Disadvantages	Suffer from premature convergence Long iterations in some cases

Fig. 17 Percentage diagram of using SHO algorithm with different models



parameter is used in most cases to establish a trade-off between exploration and extraction in the problem environment.

Although it is challenging to find the best optimal solution due to the extensive and dynamic range of problems and complexity of calculations, it is impossible. Applying the optimal solution to NP problems in an environment of more than n dimensions is often very computationally expensive or even impossible in a limited amount of time. Therefore, intelligent approaches, such as metaheuristic algorithms, must be developed to discover optimal solutions. Metaheuristic algorithms can learn and provide appropriate solutions to very complex problems. These algorithms have attracted researchers' attention for the following reasons: the increased complexity of problems, the increased range of solutions in multidimensional environments, the dynamic nature of the issue, and the limitations for solving optimization problems.

5 Conclusion and Future Works

This paper uses the SHO to solve various optimization problems. For this purpose, the application of the SHO in hybridization, improved variants, and optimization problems was investigated. This survey comprehensively and exhaustively summarizes references published from 2017 until May 2021. In this survey, advantages and disadvantages, robustness, and weaknesses were analyzed. Many of these papers suggested variants of the SHO algorithm, which enhance the performance of the original

SHO algorithm and enable it to solve many kinds of optimization problems. Variants of SHO were made possible by modifying or changing the control parameters. These changes were applied to increase the overall performance and speed of results due to the performance's proper and objective function. Optimal changes in the SHO stages were achieved by establishing a trade-off between the exploration and exploitation stages and combining local and global searches. An essential aspect of performance enhancement was the improvement of SHO operators by incorporating other meta-heuristic algorithms. According to studies, the SHO has proven efficient for solving MOPs and complex multidimensional search environments. In addition, introduced the applications of SHO in various fields, for instance, machine learning (i.e., feature selection and ANNs SVR), engineering (i.e., software defect prediction, image processing, economic dispatch problems), limited engineering design, and other applications. We hope that this comprehensive review will help readers interested in using SHO in solving optimization problems.

Although assessments and results have shown the excellent performance of SHO and its variants for solving many optimization problems and engineering applications, it can still be claimed that some problems are worth considering in future studies, such as machine learning (classification, clustering, and feature selection), engineering (resource allocation in cloud computing, cloud computing task off-loading, energy consumption in cloud computing, routing in Networks, clustering in sensor node), constrained engineering problems, image processing, and others.

Declarations

Conflict of interest All authors disclose any real or apparent conflicts of interest to the readers that may directly bear on the paper's subject matter. There are no conflicts of interest in this paper, and no funding has been obtained from any organization, so it is wholly independent and free research.

References

- Shayanfar H, Gharehchopogh FS (2018) Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. *Appl Soft Comput* 71(1):728–746
- Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: whale optimization algorithm and its applications. *Swarm Evol Comput* 48:1–24
- Benyamin A, Farhad SG, Saeid B (2021) Discrete farmland fertility optimization algorithm with metropolis acceptance criterion for traveling salesman problems. *Int J Intell Syst* 36(3):1270–1303
- Rahnema N, Gharehchopogh FS (2020) An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering. *Multimed Tools Appl* 79(43):32169–32194
- Abdollahzadeh B, Gharehchopogh FS (2021) A multi-objective optimization algorithm for feature selection problems. *Eng Comput*. <https://doi.org/10.1007/s00366-021-01369-9>
- Karakoyun M, Ozkis A, Kodaz H (2020) A new algorithm based on gray wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems. *Appl Soft Comput* 96(1):106560
- Omran MGH, Al-Sharhan S (2019) Improved continuous ant colony optimization algorithms for real-world engineering optimization problems. *Eng Appl Artif Intell* 85(1):818–829
- Shadravan S, Naji HR, Bardsiri VK (2019) The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Eng Appl Artif Intell* 80(1):20–34
- Mohammadzadeh H, Gharehchopogh FS (2020) A multi-agent system based for solving high-dimensional optimization problems: a case study on email spam detection. *Int J Commun Syst* 34(3):e4670
- Moghdani R et al (2020) An improved volleyball premier league algorithm based on sine cosine algorithm for global optimization problem. *Eng Comput* 15(1):1–26
- Dorigo M, Maniezzo V, Colnani A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B (Cybern)* 26(1):29–41
- Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139(1):98–112
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95(1):51–67
- Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 17(12):4831–4845
- Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Comput Struct* 169(1):1–12
- Yu JJQ, Li VOK (2015) A social spider algorithm for global optimization. *Appl Soft Comput* 30(1):614–627
- Zheng Y-J (2015) Water wave optimization: a new nature-inspired metaheuristic. *Comput Oper Res* 55:1–11
- Yang X-S (2010) Firefly algorithm, lévy flights and global optimization. In: Bramer M, Ellis R, Petridis M (eds) *Research and development in intelligent systems XXVI*. Springer, London
- Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27(4):1053–1073
- Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl-Based Syst* 159(1):20–50
- Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83(1):80–98
- Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23(3):715–734
- Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89(1):228–249
- Moazzeni AR, Khamsehchi E (2020) Rain optimization algorithm (ROA): a new metaheuristic method for drilling optimization solutions. *J Petrol Sci Eng* 195(1):107512
- Moldovan D (2020) Horse optimization algorithm: a novel bio-inspired algorithm for solving global optimization problems. In: Silhavy R (ed) *Artificial intelligence and bioinspired computational methods*. Springer International Publishing, Cham
- Gonçalves MS, Lopez RH, Miguel LFF (2015) Search group algorithm: a new metaheuristic method for the optimization of truss structures. *Comput Struct* 153(1):165–184
- Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96(1):120–133
- Kaveh A, Eslamlou AD (2020) Water strider algorithm: a new metaheuristic and applications. *Structures* 25(1):520–541
- Ampellio E, Vassio L (2016) A hybrid swarm-based algorithm for single-objective optimization problems involving high-cost analyses. *Swarm Intell* 10(2):99–121
- Naidu YR, Ojha AK, Devi VS (2020) Multi-objective jaya algorithm for solving constrained multi-objective optimization problems. In: Kim JH, Geem ZW, Jung D, Yoo DG, Yadav A (eds) *Advances in harmony search, soft computing and applications*. Springer International Publishing, Cham
- Al-Madi N, Faris H, Mirjalili S (2019) Binary multi-verse optimization algorithm for global optimization and discrete problems. *Int J Mach Learn Cybern* 10(12):3445–3465
- Zhang Z et al (2019) Birds foraging search: a novel population-based algorithm for global optimization. *Memetic Comput* 11(3):221–250
- Alshinwan M et al (2021) Dragonfly algorithm: a comprehensive survey of its results, variants, and applications. *Multimed Tools Appl* 80(10):14979–15016
- Abualigah L, Diabat A (2021) Advances in sine cosine algorithm: a comprehensive survey. *Artif Intell Rev* 54(4):2567–2608
- Abualigah L et al (2021) Lightning search algorithm: a comprehensive survey. *Appl Intell* 51(4):2353–2376
- Abualigah L, Diabat A (2020) A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications. *Neural Comput Appl* 32(19):15533–15556
- Abualigah L (2021) Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Comput Appl* 33(7):2949–2972
- Shehab M et al (2020) Moth-flame optimization algorithm: variants and applications. *Neural Comput Appl* 32(14):9859–9884
- Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114(1):48–70
- Jia H et al (2019) Spotted hyena optimization algorithm with simulated annealing for feature selection. *IEEE Access* 7(1):71943–71962

41. Dhiman G, Kaur A (2019) A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization. In: Bansal JC, Das KN, Nagar A, Deep K, Ojha AK (eds) *Soft computing for problem solving*. Springer, Singapore
42. Sharma, A. and R. Rani (2018) C-HDESHO: Cancer Classification Framework using Single Objective Meta—heuristic and Machine learning Approaches. In: 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC). Solan Himachal Pradesh, India: IEEE
43. Dhiman G (2021) SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. *Knowledge-Based Syst* 222(1):106926
44. Luo Q et al (2021) Using spotted hyena optimizer for training feedforward neural networks. *Cogn Syst Res* 65(1):1–16
45. Guo Z et al (2020) Optimal modification of heating, ventilation, and air conditioning system performances in residential buildings using the integration of metaheuristic optimization and neural computing. *Energy Build* 214(1):109866
46. Moayedi H et al (2019) Spotted hyena optimizer and ant lion optimization in predicting the shear strength of soil. *Appl Sci* 9(22):4738
47. Liu W et al (2019) Proposing two new metaheuristic algorithms of ALO-MLP and SHO-MLP in predicting bearing capacity of circular footing located on horizontal multilayer soil. *Eng Comput* 37(1):1537–1547
48. Panda N, Majhi SK (2019) How effective is spotted hyena optimizer for training multilayer perceptrons. *Int J Recent Technol Eng (IJRTE)* 8(2):4915–4927
49. Li J et al (2018) Using spotted hyena optimizer for training feedforward neural networks2. In: Huang D-S, Gromiha MM, Han K, Hussain A (eds) *Intelligent computing methodologies*. Springer International Publishing, Cham
50. Malik A et al (2020) Support vector regression optimized by metaheuristic algorithms for daily streamflow prediction. *Stoch Env Res Risk Assess* 34(11):1755–1773
51. Malik A et al (2021) Support vector regression integrated with novel meta-heuristic algorithms for meteorological drought prediction. *Meteorol Atmos Phys* 133:1–20
52. Soto R et al (2019) Solving the set covering problem using spotted hyena optimizer and autonomous search. In: Wotawa F, Friedrich G, Pill I, Koitz-Hristov R, Ali M (eds) *Advances and trends in artificial intelligence. From theory to practice*. Springer International Publishing, Cham
53. Begum, B., et al. (2020) Optimal design and implementation of fuzzy logic based controllers for LFC study in power system incorporated with wind farms. In: 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSSE). Keonjhar, India: IEEE
54. Zhou G et al (2020) An improved spotted hyena optimizer for PID parameters in an AVR system. *Math Biosci Eng* 17(4):3767–3783
55. Panda N et al (2020) Oppositional spotted hyena optimizer with mutation operator for global optimization and application in training wavelet neural network. *J Intell Fuzzy Syst* 38(5):6677–6690
56. Kumar V, Kaleka KK, Kaur A (2021) Spiral-inspired spotted hyena optimizer and its application to constraint engineering problems. *Wirel Pers Commun* 116(1):865–881
57. Wang, H., et al. (2009) Space transformation search: a new evolutionary technique. In: *Proceedings of the first ACM/SIGEVO summit on genetic and evolutionary computation*. Association for Computing Machinery: Shanghai, China. pp. 537–544
58. Panda N, Majhi SK (2020) Improved spotted hyena optimizer with space transformational search for training pi-sigma higher order neural network. *Comput Intell* 36(1):320–350
59. Kumar V, Kaur A (2020) Binary spotted hyena optimizer and its application to feature selection. *J Ambient Intell Humaniz Comput* 11(7):2625–2645
60. Şahman MA (2021) A discrete spotted hyena optimizer for solving distributed job shop scheduling problems. *Appl Soft Comput* 106(1):107349
61. Sukpancharoen, S. (2021) Application of spotted hyena optimizer in cogeneration power plant on single and multiple-objective. In: 2021 IEEE 11th Annual computing and communication workshop and conference (CCWC). 2021. IEEE
62. Dhiman G, Garg M (2020) MoSSE: a novel hybrid multi-objective meta-heuristic algorithm for engineering design problems. *Soft Comput* 24(24):18379–18398
63. Kaur S, Awasthi LK, Sangal AL (2020) HMOSHSSA: a hybrid meta-heuristic approach for solving constrained optimization problems. *Eng Comput* 30(1):1–22
64. Divya S et al (2020) Prediction of gene selection features using improved multi-objective spotted hyena optimization algorithm. In: Jain LC, Tsihrintzis GA, Balas VE, Sharma DK (eds) *Data communication and networks*. Springer, Singapore
65. Dhiman G (2020) MOSHEPO: a hybrid multi-objective approach to solve economic load dispatch and micro grid problems. *Appl Intell* 50(1):119–137
66. Sharma A, Rani R (2019) C-HMOSHSSA: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. *Comput Methods Progr Biomed* 178(1):219–235
67. Dhiman G, Kumar V (2018) Multi-objective spotted hyena optimizer: a multi-objective optimization algorithm for engineering problems. *Knowl-Based Syst* 150(1):175–197
68. Martinez-Rios F, Murillo-Suarez A (2021) Multi-threaded spotted hyena optimizer with thread-crossing techniques. *Proc Comput Sci* 179(1):432–439
69. Khan MR, Das B (2021) Multiuser detection for MIMO-OFDM system in underwater communication using a hybrid bionic binary spotted hyena optimizer. *J Bionic Eng* 18(2):462–472
70. Abellán N et al (2021) Deep learning classification of tooth scores made by different carnivores: achieving high accuracy when comparing African carnivore taxa and testing the hominin shift in the balance of power. *Archaeol Anthropol Sci* 13(2):31
71. Verma S et al (2021) Toward green communication in 6G-enabled massive internet of things. *IEEE Internet Things J* 8(7):5408–5415
72. Wang N et al (2021) A novel dynamic clustering method by integrating marine predators algorithm and particle swarm optimization algorithm. *IEEE Access* 9(1):3557–3569
73. Yıldız BS (2020) The spotted hyena optimization algorithm for weight-reduction of automobile brake components. *Mater Test* 62(4):383–388
74. Elsabagh MA, Farhan MS, Gafar MG (2020) Cross-projects software defect prediction using spotted hyena optimizer algorithm. *SN Appl Sci* 2(4):538
75. Kaleka KK, Kaur A, Kumar V (2020) A conceptual comparison of metaheuristic algorithms and applications to engineering design problems. *Int J Intell Inf Database Syst* 13(2–4):278–306
76. Luo Q, Li J, Zhou Y (2019) Spotted hyena optimizer with lateral inhibition for image matching. *Multimed Tools Appl* 78(24):34277–34296
77. Dhiman G, Kumar V (2019) Spotted hyena optimizer for solving complex and non-linear constrained engineering problems. In: Yadav N, Yadav A, Bansal JC, Deep K, Kim JH (eds) *Harmony search and nature inspired optimization algorithms*. Springer, Singapore
78. Dhiman G, Kaur A (2018) Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer. *Designs* 2(3):28
79. Dhiman G, Guo S, Kaur S (2018) ED-SHO: A framework for solving non-linear economic load power dispatch problem using spotted hyena optimizer. *Mod Phys Lett A* 33(40):1850239

80. Dhiman, G, Kaur A (2017). Spotted hyena optimizer for solving engineering design problems. In: 2017 International conference on machine learning and data science (MLDS). Noida, India: IEEE
81. Abualigah L et al (2020) Salp swarm algorithm: a comprehensive survey. *Neural Comput Appl* 32(15):11195–11215

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