International Journal of Machine Learning and Cybernetics (2023) 14:889–909
https://doi.org/10.1007/s13042-022-01670-z

ORIGINAL ARTICLE

A novel metaheuristic optimisation approach for text sentiment analysis

Ali Hosseinalipour[1] · **Reza Ghanbarzadeh**[2]

Received: 2 January 2022 / Accepted: 18 September 2022 / Published online: 15 October 2022

Abstract

Automated sentiment analysis is considered an area in natural language processing research that seeks to understand a text author's mood, thoughts, and feelings. New opportunities and challenges have arisen in this field due to the popularity and accessibility of a variety of resources of ideas, such as online review websites, personal blogs, and social media. Feature selection, which can be conducted using metaheuristic algorithms, is one of the steps of sentiment analysis. It is crucial to use high-performing algorithms for feature selection. This paper applies the Horse herd Optimisation Algorithm (HOA) for feature selection in text sentiment analysis. HOA is a metaheuristic algorithm and uses six key behaviours to simulate the social performance of horses of various ages, to solve high-dimensional optimisation problems. In order to improve HOA, this paper adds another behaviour of horses to the basic algorithm; thus, the new algorithm uses seven key behaviours of horses of different ages to imitate their social performance. It is then discretised and converted to a multi-objective algorithm. The improved algorithm's performance is evaluated using 15 CEC benchmark functions, and the results are compared to the Binary Social Spider Algorithm, the Binary Grey Wolf Optimizer, and the Binary Butterfly Optimization Algorithm. The new algorithm, the Multi-objective Binary Horse herd Optimisation Algorithm (MBHOA), excels at solving high-dimensional complex problems. To evaluate the algorithm's performance in feature selection, as a practical example, it is employed in text sentiment analysis and examined on various data sets. The simulation results indicate that MBHOA has a better performance in analysing sentiment compared to similar approaches.

Keywords Text sentiment analysis · Metaheuristics · Feature selection · Horse herd optimisation algorithm

1 Introduction

Emotions play a crucial role in human actions and behaviours. To a considerable extent, how others see and judge the world impacts our reality perceptions and thoughts and the actions we take based on them. As a result, we frequently seek the counsel of others when we need to make a decision. This is true not only for individuals but also for organisations [32]. Sentiment analysis is the process of computationally identifying and categorising people's opinions conveyed in a text, primarily to assess whether the writer's attitude towards a specific topic or product is favourable, negative, or neutral, and it is becoming a necessary tool for converting emotions and attitudes into useful data [36]. Addressing sentiment analysis has some requirements as it is a very complex natural ÅÅlanguage processing task that includes different phases for analysing the sentiment in the source materials. Several sub-tasks are required to perform analysis on given texts. These sub-tasks include data collection, text preparation, sentiment detection, sentiment classification, and presentation of the findings. Due to the complexity of the problem, a robust algorithm is required to address the analysis. Sentiment analysis, in nature, is a classification problem that determines if the written text is positive, neutral, or negative automatically. A classifier is required to classify the given text. The classifier requires the text to be presented as a vector of features, and a feature selection technique should be used to select the best features. Feature selection is a multi-objective problem which aims to

✉ Reza Ghanbarzadeh
Reza.Ghanbarzadeh@scu.edu.au

Ali Hosseinalipour
Ali.Hosseinalipour@yahoo.com

[1] Department of Computer Engineering, Heris Branch, Islamic Azad University, Heris, Iran

[2] Faculty of Science and Engineering, Southern Cross University, Gold Coast, Australia

improve the classification performance while reducing the number of selected features. Therefore, the selected optimisation algorithm should be able to address multi-objective problems. Furthermore, feature selection is a discrete problem; thus, the used algorithms should be able to solve discrete optimisation problems.

In an optimisation problem, finding the best solution is not the sole consideration; researchers also examine the cost and efficiency of the solution. Applied optimisation problems have high dimensions and complexity in which multiple decision variables and complex non-linear relationships could be used. To solve optimisation problems, various methods have been developed, including gradient-based methods and numerical calculations. The application of methods like gradient is limited to solving simple derivative functions. The gradient approach is useless and inapplicable when the conditions of continuity and derivability of functions are not met. The accuracy of numerical computation methods, on the other hand, depends on selecting the suitable initial solution, despite their widespread use in solving optimisation problems [44]. In these types of methods, selecting incorrect initial solutions results in getting trapped in a local optimum. Due to the limitations of approaches such as gradients and numerical computations in solving optimisation problems, metaheuristic algorithms have been introduced as a new form of intelligent search algorithms. Metaheuristic algorithms are random methods inspired by nature, attempting to move their initial population towards the global optimum and provide appropriate solutions in a fair amount of time close to the global optimum. Several metaheuristic algorithms are based on behaviours of the swarm intelligence of animals, insects, cells and other natural phenomena. In these types of algorithms, the overall intelligence of the system is derived from the intelligence of any individual component [215, 41].

In this study, the Horse herd Optimisation Algorithm (HOA) [35], which is a metaheuristic algorithm, was improved and applied for text sentiment analysis. HOA shows a remarkable performance when solving complex problems due to the diversity of control factors on the basis of the behaviour of horses of varying ages. This algorithm's performance has been compared to several well-known nature-inspired optimisation algorithms. HOA outperforms well-known optimisation algorithms in efficiency and accuracy in solving high-dimensional global optimum problems, and it has lower cost and computational complexity.

Feature selection is now considered an essential and significant process when using data mining techniques and machine learning as a result of the expansion of data volume and dimensionality. In reality, feature selection is an important stage that is sometimes regarded as a precondition for classification algorithms [34, 42]. Training a classifier using large data sets with high dimensions could cause a problem called overfitting in the learning methods. This problem reduces the model's generalisability, contributing to a reduction in the classification methods' accuracy for new test samples. Also, high-dimensional data sets require more processing time to develop a model based on the training data as well as to test the model [48]. Therefore, feature selection aims to simplify the data set and improve its quality by selecting essential and critical features [4]. Moreover, feature selection can help better understand the domain and maintain appropriate features based on some significant criteria to describe inherent patterns in data, reducing the effects of the dimensional curse [17].

Using six important behaviours, HOA simulates the social performance of horses of various ages. In this study, the performance of HOA is improved by adding another behaviour of horses of various age groups. As a result, the new algorithm integrates seven key features of the social performance of horses at different ages. The new algorithm is binarised since it will be employed in solving discrete feature selection problems. The binarised algorithm is then modified to become a multi-objective algorithm, as the feature selection would be considered a multi-objective problem. Finally, the new Binary Multi-objective HOA algorithm (MBHOA) is used for text classification and sentiment analysis.

For evaluating the new algorithm's performance, 15 CEC functions are used, and then a comparative evaluation with some well-known algorithms is presented. As a practical example for further validation, MBHOA is applied in sentiment analysis, and the results prove its efficiency. In summary, nature-inspired optimisation is very successful and effective and is highly recommended to researchers interested in using metaheuristic algorithms to solve complex problems with high dimensions.

The followings are the main contributions of the current paper:

- HOA is improved
- The improved HOA is discretised
- The discrete HOA is made multi-objective to be suitable for multi-objective feature selection problems
- The new algorithm is applied for text sentiment analysis

The remainder of this article is organised as follows: the related works to the current study are presented in Sect. 2. The proposed algorithm is introduced in Sect. 3. The performance of the proposed algorithm and its evaluation and analysis are discussed in Sect. 4. Section 5 presents a practical example of the application of the proposed algorithm, followed by the conclusion in Sect. 6.

2 Related works

To solve optimisation problems, various nature-inspired metaheuristic methods are available. Metaheuristic methods provide a general optimisation framework for iteratively improving current solution(s) to reach an optimal solution [20]. This is done using intelligent operators of knowledge acquisition and controlled random features. Robust metaheuristic algorithms' operators can explore multiple areas in the problem's search space and exploit the accumulated knowledge achieved during the exploration in an efficient way. Exploration and exploitation are mutually exclusive concepts. The key algorithmic challenge of optimisation approaches is to balance exploration and exploitation while searching. The following are the primary benefits of metaheuristic algorithms [6]:

1) These algorithms' simplicity makes them adaptable for a variety of optimisation problems with minimal modification. The optimisation problem is treated as a mathematical BlackBox by metaheuristic algorithms in which problem-solving knowledge has not been studied in depth.
2) In the initial search, metaheuristic algorithms do not need derivative information.
3) Metaheuristic algorithms are able to avoid local optimum easily with the help of their stochastic-based components.

A wide range of metaheuristic algorithms have been inspired by natural phenomena and are categorised into four main categories: evolutionary-based algorithms (e.g., the genetic algorithm), swarm-based algorithms (e.g., swarm intelligence optimisation algorithm), physic-based algorithms (e.g., simulated annealing algorithm), and human-based algorithms (e.g., imperialist competitive) [6]. The base algorithm used in the proposed method in the current paper is the Horse herd Optimisation Algorithm (HOA) [35], a metaheuristic algorithm introduced in 2021 which has shown successful results.

Feature selection is an important process in text mining, and there are several algorithms available to implement feature selection using machine learning techniques. Filtering methods are one type of these methods. However, since these methods are not efficient enough, recently, optimisation and metaheuristic algorithms have been used for feature selection [23, 24]. In this section, two filtering methods are explained. The first one, which is an algorithm for feature ranking, is Mutual Information Features Selection (MIFS). MIFS uses class labels to determine the amount of information between features and shows a numeric value for each feature, which is the classification value for that feature [11]. The second one is Random Selection Features Selection (RSFS) which selects a feature subset at random. Based on the training data set, this algorithm iteratively evaluates the selected feature's quality until a subset of features with the best classification according to class labels is selected [40].

In 2007, the Ant Colony Algorithm (ACO) was employed to solve feature selection problems. In this method, two strings of zero and one are defined where the length of the strings equals the number of variables; A string of ones and zeros is then generated. Next, the generated string by each ant is encrypted with the value of one, indicating the selection of the feature or the value of zero, indicating non-selection [29].

De Stefano, Fontanella, Marrocco, and Schirinzi [14] used the genetic algorithm in solving feature selection. In this method, the search operation is carried out using the genetic algorithm. Each individual encodes the selection of space, and its fitness would be the measurement of the separability of the class in that space.

The firefly algorithm was used for feature selection in 2011. There are *n* fireflies, and each one is related to a particular feature in the entire feature space. Every firefly crawls towards the nearest firefly, increasing the brightness of their light. Any firefly that is not attracted by the others or can not find an equivalent is instantly absorbed by the environment and eliminated [10]. Forsati, Moayedikia, Keikha, and Shamsfard [22] employed the bee colony optimisation algorithm for feature selection. In this algorithm, bees first take a certain number of steps ahead and decide to select a feature at each step.

Diao and Shen [16] introduced another feature selection approach based on harmonic search. An expert or feature selector is best described as a musician. A feature, to be included in a subset, can be voted on by each musician. In the end, harmony includes the votes of all musicians. The harmonic evaluation function is then used to select subsets of features.

The binary cuckoo algorithm was introduced by Yampolskiy and El-Barkouky [46]. In this approach, the search space can be modelled as a cube with *d* dimensions, where *d* is the quantity of the features. One set of binary coordinates is developed for sharing each nest, indicating whether or not a particular feature belongs to the final feature set. The solution's quality highly depends on the number of nests.

For feature selection, Mousavirad and Ebrahimpour-Komleh [38] introduced the modified imperialist competitive algorithm. First, primitive empires are produced. Each country is defined as a single-dimensional string array of zeros and ones where one indicates selecting a feature and zero means not selecting a feature.

By applying the improved the shuffled frog leaping algorithm, Hu et al. [26] developed a method for selecting features in high-dimensional biomedical data. This approach has thousands of features and can be utilised for the molecular diagnosis of disease. The methods for feature selection can be applied to identifying patterns accurately. The basic aim of feature selection is to isolate a subset of the main set of features to reduce the computational cost of data mining.

In the literature, several metaheuristic algorithms are employed for feature selection. Table 1 demonstrates a list of some sample publications that used metaheuristic algorithms for sentiment analysis, with their advantages and disadvantages.

3 Methodology

3.1 Horse herd optimisation algorithm (HOA)

As mentioned earlier, the baseline algorithm used and improved in this study is the HOA [35]. HOA is a metaheuristic algorithm that was inspired by the social behaviours of horses in the herd. This algorithm is developed based on six essential behaviours of the social performance of horses at different ages. It has a remarkable ability to explore and exploit, and can achieve optimal solutions for highly complex and complicated problems. The six behaviours of horses that are inspired by this algorithm are as follows: grazing, hierarchy, sociability, imitation, defence mechanism, and roaming [35, 45].

Due to the variety of control factors, HOA shows an excellent performance in solving complex high-dimensional problems. This algorithm could be compared to a wide range of well-known nature-inspired algorithms. The use of multiple benchmark functions with high dimensions (maximum of 10,000 dimensions) demonstrates that HOA is highly efficient for solving complex global optimisation problems [35].

The maximum lifespan of a horse is 25–30 years, and at different ages, horses exhibit various behaviours. Horses are split into four groups in the HOA algorithm, based on their age: δ represents horses between the ages of 0 and 5 years, γ represents horses between the ages of 5 and 10 years, β represents horses between the ages of 10 and 15 years, and α represents horses that are beyond 15 years old. In order to determine the age of horses, each iteration must conduct a comprehensive matrix of responses. This matrix could be sorted based on the best responses by selecting the first ten per cent of horses at the top of the matrix as the α horses. The next twenty, thirty and forty per cent will be in the β, γ and δ groups, respectively. In each iteration, the horses' movement is implemented in accordance with Eq. (1):

Table 1 Sample studies used metaheuristic approaches in sentiment analysis

Reference	Approach	Advantage	Disadvantage
[43]	"Feature selection using binary ant algorithm" (ACO)	Solves the problems of the discrete ant colony algorithm in feature selection	The algorithm's search capability is extremely limited. Because each ant sees the same set of features and can only choose whether or not to select a feature, Ants are unable to see other features
[1]	"Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums" (IG + GA)	Improves sentiment accuracy of classification successfully and identifies the best subset of features	The data is in a document in this approach. Data, in the form of a document, only considers emotions according to the general document without seeing the details of the document. Sentiment analysis at the document level does not allow for the identification of people's interests
[5]	"Sentiment classification using rough set based hybrid feature selection" (RST)	Can assess the significance of the features of a document	The relationship between features is not taken into account in this method. First, the threshold value must be set
[7]	"Different feature selection for sentiment classification (RST)"	Decreases the quantity of irrelevant, redundant, or noisy features. RST considers the features' combinational dependencies	It is challenging to obtain the ideal number of features using this method. This method has a lengthy procedure

$$X_m^{Iter,AGE} = \vec{V}_m^{Iter,AGE} + X_m^{(Iter-1),AGE}, AGE = \alpha, \beta, \lambda, \delta \tag{1}$$

where $X_m^{Iter,AGE}$ indicates the position of the m^{th} horse, $\overrightarrow{V}_m^{Iter,AGE}$ indicates the velocity vector of the m^{th} horse, *AGE* indicates the horse age range, and *Iter* indicates the current iteration. Equation 1 calculates the movement applied to horses at each iteration. In every iteration, horses move according to Eq. (1). Using this Equation, each horse is updated within the optimisation process, taking into consideration the age of the horse to determine which feature is from its advantages.

The mathematical processes for modelling six behaviours are undertaken to detect the vector of velocity. According to the above-mentioned horse behavioural patterns, Eq. (2) can be used to express the motion vector of horses at different ages during each cycle of the algorithm [30, 35, 45]:

The motion vector of α horses:

$$\vec{V}_m^{Iter,\beta} = \vec{G}_m^{Iter,\alpha} + \vec{D}_m^{Iter,\alpha}$$

The motion vector of β horses:

$$\vec{V}_m^{Iter,\beta} = \vec{G}_m^{Iter,\beta} + \vec{H}_m^{Iter,\beta} + \vec{S}_m^{Iter,\beta} + \vec{D}_m^{Iter,\beta}$$

The motion vector of γ horses:

$$\vec{V}_m^{Iter,\gamma} = \vec{G}_m^{Iter,\gamma} + \vec{H}_m^{Iter,\gamma} + \vec{S}_m^{Iter,\gamma} + \vec{I}_m^{Iter,\gamma} + \vec{D}_m^{Iter,\gamma} + \vec{R}_m^{Iter,\gamma}$$

The motion vector of δ horses:

$$\vec{V}_m^{Iter,\delta} = \vec{G}_m^{Iter,\delta} + \vec{I}_m^{Iter,\delta} + \vec{R}_m^{Iter,\delta} \tag{2}$$

Based on the age of the horses, their velocity will be updated according to the above equations for α, β, γ, and δ horses, respectively.

Horses' six behaviours that the HOA algorithm imitates are explained below.

3.1.1 Grazing

Horses are herbivores who eat plants, grasses, and other forage, and they spend 16 to 20 hours a day grazing on a pasture, taking short breaks. Continuous eating is the term for this type of gradual grazing [30]. Horses graze at any age, and HOA usens coefficient g to simulate the grazing area around each horse, resulting in each horse grazing in specified areas (according to Fig. 1). Equations (3) and (4) are used for the mathematical implementation of horse grazing.

$$\vec{G}_m^{Iter,AGE} = g_{Iter}\left(\breve{u} + \rho\,\breve{l}\right) + [X_m^{(Iter-1)}], AGE = \alpha, \beta, \gamma, \delta \tag{3}$$

$$g_m^{Iter,AGE} = g_m^{(Iter-1),AGE}\omega_g \tag{4}$$

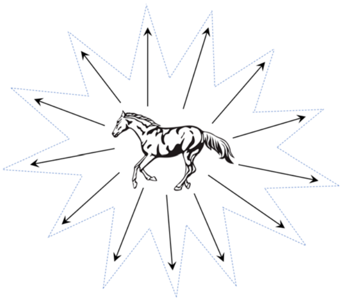


Fig. 1 Simulation of a horse grazing behaviour

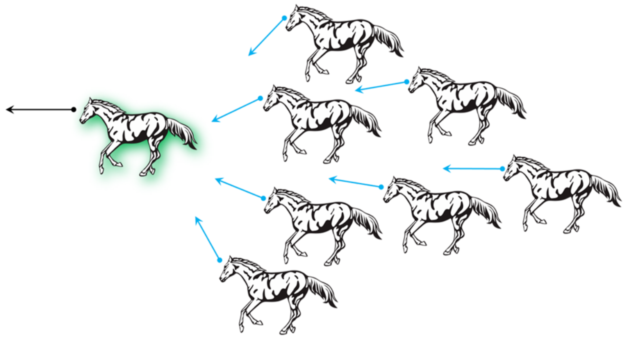


Fig. 2 Simulation of Hierarchy of horses

In the above equations, $\overrightarrow{G}_m^{Iter,AGE}$ is the *ith* horse's motion parameter indicating its graze tendency. With ω_g in each iteration, this factor decreases linearity. $\breve{u}$ and $\breve{l}$ are the upper bound and the lower bound of the grazing area, and their recommended value are 1.05 and 0.95, respectively. ρ is a random number between zero and one. The recommended coefficient g for all age ranges is 1.5.

3.1.2 Hierarchy

Horses are not self-sufficient and require leadership. Most of the time, the leader is a human; however, in wild horse herds, an adult mare or stallion could also take on the role of leader [12]. In HOA, the coefficient h is the tendency of horse herds to follow the horse with the most experience and strength. This is known as the law of hierarchy in horse herds. Figure 2 shows the simulation of the hierarchy in the horse herd. Between the ages of 5 and 15 (β and γ), horses appear to follow the hierarchy law [45]. Equations (5) and (6) are used to define a hierarchy in horse herds:

$$\vec{H}_m^{Iter,AGE} = h_m^{Iter,AGE}\left[X_*^{(Iter-1)} - X_m^{(Iter-1)}\right], AGE = \alpha, \beta\ and\ \gamma \tag{5}$$

$$h_m^{Iter,AGE} = h_m^{(Iter-1),AGE}\omega_h \tag{6}$$

Where $\overrightarrow{H}_m^{Iter,AGE}$ indicates the impact of the location of the leader horse on the velocity, and $X_*^{(Iter-1)}$ is the location of the leader horse.

3.1.3 Sociability

Horses are social creatures who can cohabit with other animals in the wild. The herd life has maintained their safety since predators often hunt them. Pluralism improves the survival chance and makes it easier to flee from predators. Some horses appear to enjoy being with other animals too, but they generally dislike being alone, and singularity can contribute to irritability [30]. This behaviour of horses is implemented by factor s as the horse's movement towards the average location of other horses, and it is depicted in Fig. 3. The horses mostly between the ages of 5 and 15 are attracted to the herd, and this fact is incorporated in Eqs. (7) and (8):

$$\vec{S}_m^{Iter,AGE} = s_m^{Iter,AGE}\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right], AGE = \beta, \gamma \tag{7}$$

$$s_m^{Iter,AGE} = s_m^{(Iter-1),AGE} \times \omega_s \tag{8}$$

where $\vec{S}_m^{Iter,AGE}$ is the social vector motion of the i^{th} horse, and $s_m^{Iter,AGE}$ is that horse's orientation towards the herd in the $Iter^{th}$ iteration. With a ω_s factor, $s_m^{Iter,AGE}$ is reduced in each cycle. N is the total number of horses, and AGE is each horse's age range. The s coefficient of β and γ horses is calculated in the parameters sensitivity analysis.

3.1.4 Imitation

Horses imitate one another and learn each other's good and bad habits, such as seeking the best grazing area [12]. This behaviour is also considered in HOA as another social performance in horses. Young horses usually imitate other horses, which persists throughout their lives until they become mature. Simulation of horse imitation is shown in Fig. 4, and Eqs. (9) and (10) are used for the mathematical implementation of this behaviour.

$$\vec{I}_m^{Iter,AGE} = i_m^{Iter,AGE}\left[\left(\frac{1}{pN}\sum_{j=1}^{pN} \hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right], AGE = \gamma \tag{9}$$

$$i_m^{Iter,AGE} = i_m^{(Iter-1),AGE}\omega_i \tag{10}$$

In the above equations, $\vec{I}_m^{Iter,AGE}$ is the ith horse's motion vector towards the best horses' average with locations of $\hat{X}$. pN is the total number of horses with the best locations, and the recommended value for p is 10 per cent of the total number of horses. ω_i is a reduction factor in each iteration for i_{iter}.

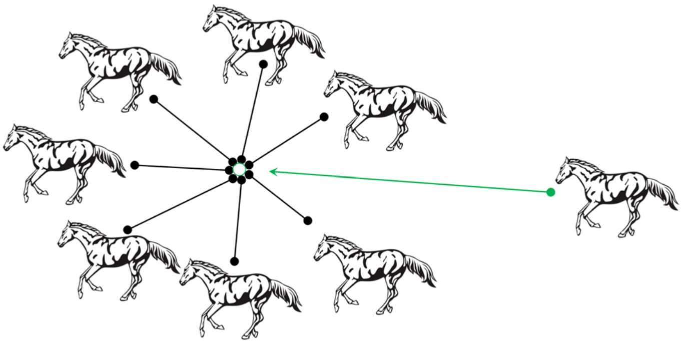


Fig. 3 Simulation of general movements of horses for their sociability

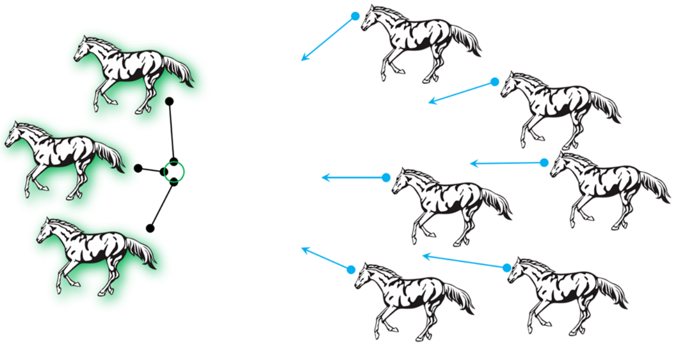


Fig. 4 Simulation of imitation of horses

3.1.5 Defence mechanism

Horses' behaviour reflects the fact that they have been preyed upon [45]. They use the fight-or-flight reaction to defend themselves. Their first reaction is to flee, and when trapped, they buck. Horses fight for food and water to keep competitors at bay, and instinctively avoid dangerous places where enemies such as wolves exist [30]. When possible, such a defence mechanism is present throughout a horse's life, whether young or old. Horses' defence mechanism is simulated in HOA as running away from other horses demonstrating inappropriate behaviours. These behaviours are far from optimal. Figure 5 depicts the simulation of the defence mechanism of horses. Factor d characterises the defence mechanism. In order to keep these animals away from inappropriate positions, their defence mechanism is presented as a negative coefficient in Eqs. (11) and (12):

$$\vec{D}_m^{Iter,AGE} = -d_m^{Iter,AGE}\left[\left(\frac{1}{qN}\sum_{j=1}^{pN}\dot{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right], AGE =, \alpha, \beta \text{ and } \gamma \tag{11}$$

$$d_m^{Iter,AGE} = d_m^{(Iter-1),AGE}\omega_d \tag{12}$$

In the above equations, $\vec{D}_m^{Iter,AGE}$ is the the escape vector of ith horse from the average of some horses with worst locations, which are shown by the $\breve{X}$ vector. The number of horses with the worst locations is qN. The recommended value for q is 20 per cent of the total number of horses. ω_d is the reduction factor per cycle for d_{iter}.

3.1.6 Roam

In the wild, in search of food and graze, horses roam from pasture to pasture. Most horses are maintained in stables, although they preserve their roaming behaviour. Horses are inquisitive animals and frequently visit various pastures to find new grazing places and to know their surroundings [45]. Figure 6 depicts the simulation of the roaming behaviour of horses. In HOA, roaming behaviour is simulated as a random movement of a horse, and it is indicated by r. The roaming behaviour is more common in younger horses and gradually decreases with maturity. This is simulated using the following Eqs. (13) and (14):

$$\vec{R}_m^{Iter,AGE} = r_m^{Iter,AGE}pX^{(Iter-1)}, AGE = \alpha, \beta \text{ } and \text{ } \gamma \tag{13}$$

$$r_m^{Iter,AGE} = r_m^{(Iter-1),AGE}\omega_r \tag{14}$$

$\vec{R}_m^{Iter,AGE}$ is the random velocity vector of ith horse for a local search and an escape from local minima. The reduction factor of $r_m^{Iter,AGE}$ per iteration is represented by ω_r.

The general velocity vector is calculated through substituting Eqs. (3) to (14) at Eq. (2). The velocity of δ, γ, β, and α horses are calculated by Eqs. (15), (16), (17), and (18), respectively.

The velocity of δ horses is calculated by:

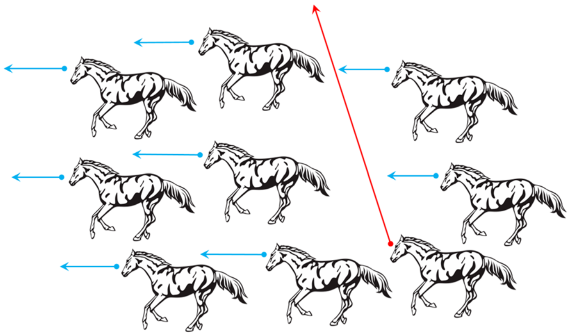
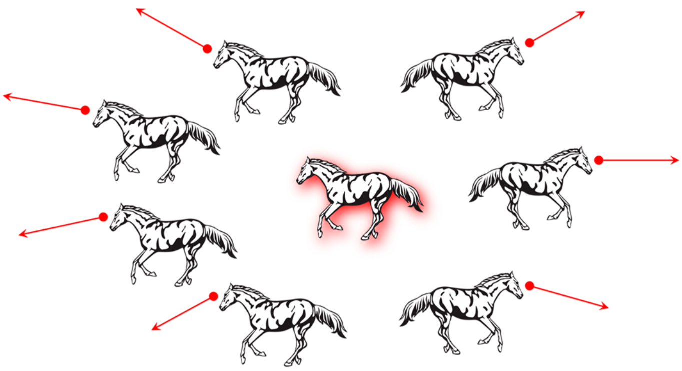


Fig. 6 Simulation of the roam

Fig. 5 Simulation of the defence mechanism of horses



$$\vec{V}_m^{Iter,\delta} = \left[g_m^{(Iter-1),\delta}\omega_g\left(\breve{u}+\rho\breve{l}\right)+[X_m^{(Iter-1)}]\right]+\left[i_m^{(Iter-1),}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(Iter-1)}\right)-X^{(Iter-1)}\right]\right]+\left[r_m^{(Iter-1),\delta}\omega_r pX^{(Iter-1)}\right] \quad (15)$$

The velocity of γ horses is calculated by:

$$\begin{aligned}\vec{V}_m^{Iter,\gamma} =& \left[g_m^{(Iter-1),\gamma}\omega_g\left(\breve{u}+\rho\breve{l}\right)+[X_m^{(Iter-1)}]\right]+\left[h_m^{(Iter-1),\gamma}\omega_h\left[X_*^{(Iter-1)}-X_m^{(Iter-1)}\right]\right]+\left[s_m^{(Iter-1),\gamma}\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{(Iter-1)}\right)-X_m^{(Iter-1)}\right]\right]\\ &+\left[i_m^{(Iter-1),\gamma}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(Iter-1)}\right)-X^{(Iter-1)}\right]\right]-\left[d_m^{(Iter-1),\gamma}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN}\breve{X}_j^{(Iter-1)}\right)-X^{(Iter-1)}\right]\right]+\left[r_m^{(Iter-1),\gamma}\omega_r pX^{(Iter-1)}\right]\end{aligned} \quad (16)$$

The velocity of β horses is calculated by:

$$\begin{aligned}\vec{V}_m^{Iter,\beta} =& \left[g_m^{(Iter-1),\beta}\omega_g\left(\breve{u}+\rho\breve{l}\right)+[X_m^{(Iter-1)}]\right]+\left[h_m^{(Iter-1),\beta}\omega_h\left[X_*^{(Iter-1)}-X_m^{(Iter-1)}\right]\right]\\ &+\left[s_m^{(Iter-1),\beta}\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{(Iter-1)}\right)-X_m^{(Iter-1)}\right]\right]-\left[d_m^{(Iter-1),\beta}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN}\breve{X}_j^{(Iter-1)}\right)-X^{(Iter-1)}\right]\right]\end{aligned} \quad (17)$$

The velocity of α horses is calculated by:

$$\vec{V}_m^{Iter,\alpha} = \left[g_m^{(Iter-1),\alpha}\omega_g\left(\breve{u}+\rho\breve{l}\right)+[X_m^{(Iter-1)}]\right]-\left[d_m^{(Iter-1),\alpha}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN}\breve{X}_j^{(Iter-1)}\right)-X^{(Iter-1)}\right]\right] \quad (18)$$

As stated earlier, many metaheuristic algorithms have been developed and introduced to address various optimisation problems. However, not all of those algorithms are highly efficient. Many of them have time complexity problem, some have high computational complexity, others can be trapped in the local optimum and so on. HOA, on the other hand, is a novel, fast, robust and reliable algorithm that has none of the aforementioned issues and surpasses the majority of the strongest available metaheuristic algorithms as several popular test functions have benchmarked it at high dimensions. The algorithm outperforms existing high-performance algorithms such as GOA, SCA, MVO, MFO, DA, and GWO in high-dimensional spaces in analysing eleven test functions [35]. Followings are some of the main strengths and differences of HOA compared to the state of the art algorithms:

- It is a fast optimisation algorithm; the inclusion of various parameters in this algorithm results in a coherent algorithm capable of finding the optimum solution in the shortest possible time.
- It is a robust optimisation algorithm and avoids stagnation phenomena. It uses a suitable solution to avoid trapping in local optimum by employing a sorting technique in a global matrix.
- It has the ability to solve simple as well as highly complex optimisation problems. Many other algorithms are unable to solve complex optimisation problems and are only efficient in solving non-complex problems. Due to the large number of control parameters based on the horses' behaviour at different ages, this algorithm has an excellent performance in solving high-dimensional complex problems.
- This algorithm is highly efficient in exploration and exploitation (benchmarked by seven well-known test functions). The establishment of harmonious relationships between the movements of horses leads to a higher level of exploration and exploitation in this algorithm.
- This algorithm has a highly favourable balance between its components which increases the efficiency of computational complexity.
- The algorithm has the ability to solve problems that have many unknown variables in high-dimensional spaces.
- The improvement of the algorithm by incorporating the seventh horse behaviour has increased its performance even more. This will be discussed later in Sects. 4 and 5.
- The conversion of the algorithm to a binary and multi-objective algorithm has allowed the updated algorithm

to address a wide range of discrete and multi-objective optimisation problems. This will be discussed in Sect. 4.

3.2 Multi-objectivity

Some optimisation problems may have only one objective function. The models used to optimise these kinds of problems are called "single-objective" models. For a single-objective problem, optimisation aims to seek the best solution among all available solutions. Many of the design and engineering problems have multiple objective functions in practice. These kinds of optimisation problems are called "multi-objective" optimisation problems. In most cases, the objective functions defined in a multi-objective optimisation problem are in opposition to one another, which means the objectives are incompatible [17]. Sentiment analysis is considered a multi-objective problem. The pursued objective in these problems has less error in text classification and a lower number of features. To do this, text classification methods are used so that there is an optimal state between the objectives. Many single-objective metaheuristic methods have been upgraded to multi-objective [47]. To be able to apply HOA for text sentiment analysis, the algorithm requires to be converted to a multi-objective algorithm. Additionally, the text sentiment analysis problem is discrete; therefore, HOA also needs to be discretised. In the following sections, the process of upgrading HOA to a multi-objective algorithm and a binary algorithm is described.

4 The proposed algorithm

HOA is a fast and powerful approach to solving complex optimisation problems. This algorithm has been examined with seven well-known high-dimensional test functions, and its efficiency in exploration and exploitation is proven. HOA, which was originally a continuous metaheuristic algorithm, was described in Sect. 2. In this section, an improved edition of HOA will be presented that can be applied in solving feature selection problems.

Horses mature by the end of their third year of life, and attain full maturity at five years old. In some breeds, however, maturity occurs before the age of three. Mating is another key behaviour of horses, which was not imitated in the original HOA. In the proposed algorithm, this behaviour is considered along with the other six behaviours as the 7th behaviour of the social performance of horses. After incorporating the mating behaviour of horses of varied ages, their motion vector throughout each cycle of the algorithm is determined using the Eq. (19) as follows:

$$\begin{aligned}
&\vec{V}_m^{Iter,\alpha} = \vec{G}_m^{Iter,\alpha} + \vec{D}_m^{Iter,\alpha} + \overrightarrow{Ma}_m^{Iter,\alpha}\\
&\vec{V}_m^{Iter,\beta} = \vec{G}_m^{Iter,\beta} + \vec{H}_m^{Iter,\beta} + \vec{S}_m^{Iter,\beta} + \vec{D}_m^{Iter,\beta} + \overrightarrow{Ma}_m^{Iter,\beta}\\
&\vec{V}_m^{Iter,\gamma} = \vec{G}_m^{Iter,\gamma} + \vec{H}_m^{Iter,\gamma} + \vec{S}_m^{Iter,\gamma} + \vec{I}_m^{Iter,\gamma} + \vec{D}_m^{Iter,\gamma} + \vec{R}_m^{Iter,\gamma} + \overrightarrow{Ma}_m^{Iter,\gamma}\\
&\vec{V}_m^{Iter,\delta} = \vec{G}_m^{Iter,\delta} + \vec{I}_m^{Iter,\delta} + \vec{R}_m^{Iter,\delta}
\end{aligned}\tag{19}$$

4.1 Mating

The majority of the significant movements of horses and their social behaviours are considered in the original HOA algorithm, and great results have been obtained. However, it was believed that adding more behaviours of horses in the herd could also be beneficial in improving the performance of the algorithm in solving high-dimensional complex problems. After thorough research on the horses' herd behaviour, we found that the act of mating is directly involved in the formation of a set of horses in the herd. The act of mating is also important in creating a new population in order to continue the algorithm. Another important aspect of mating behaviour in horses is that the evaluation of horses at different ages begins at birth, and birth is not possible without mating behaviour as horses over five years of age (α, β, γ) in all breeds spend a part of their lives in this direction to find a mate. Seeking mates is one of the regular and significant behaviours of horses in the herd, and even affects other horse behaviours such as sociability, defence mechanism and roaming. Based on this, the mating behaviour was considered significant in horse herd life, and it was added as an additional horse behaviour to the algorithm. Although horses of age five or younger (δ) may show some act of mating, in this approach, this behaviour is only considered with horses older than five years old (α, β, γ). According to the results, the HOA with the seventh behaviour performs better than HOA with six behaviours. The simulation results of the original HOA algorithm and the improved HOA algorithm are both demonstrated in Table 3. It shows that the improved HOA has a better performance in all functions except the MaF1 function compared to the original HOA. The mating behaviour is described as follows.

Horses frequently seek mates after the age of five and continue to do so until the end of their lives; this process occurs every year between June and August. Horse mating is modelled using the *Ma* factor. This behaviour is implemented mathematically using Eqs. (20) and (21):

$$\overrightarrow{Ma}_m^{Iter,AGE} = ma_m^{Iter,AGE}\left[\left(\frac{1}{fN}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right], AGE = \alpha, \beta, \gamma \tag{20}$$

$$ma_m^{Iter,AGE} = ma_m^{(Iter-1),AGE}\omega_{ma} \tag{21}$$

where $\overrightarrow{Ma}_m^{Iter,AGE}$ is the vector of finding the ith female horse by the best (strongest) males, which is shown by $\widehat{X}$, and X_j^{Iter-1} is the position of the $\mathrm{m}^{\mathrm{th}-1}$ horse in accordance with Eq. (1). *fN* is the number of the best (strongest) males. The recommended value for *f* is 10 per cent of the total number of horses. ω_{ma} indicates the reduction factor of the d_{Iter} per cycle ($ma_m^{Iter,AGE}$ decreases in each cycle by the factor). *N* also indicates the total number of horses, and *AGE* is the age range of each horse. The *ma* coefficient decreases for horses α, β and γ, i.e. with an age range of 5 to 25 per iteration.

In Sect. 4.2, based on the Sigmoid function, a binary version of HOA will be introduced. We attempted to propose a binary version of HOA by applying minimal changes to the original version of HOA. In Sect. 4.3, a multi-objective function will be presented, which is used to increase the classification accuracy and reduce the number of features.

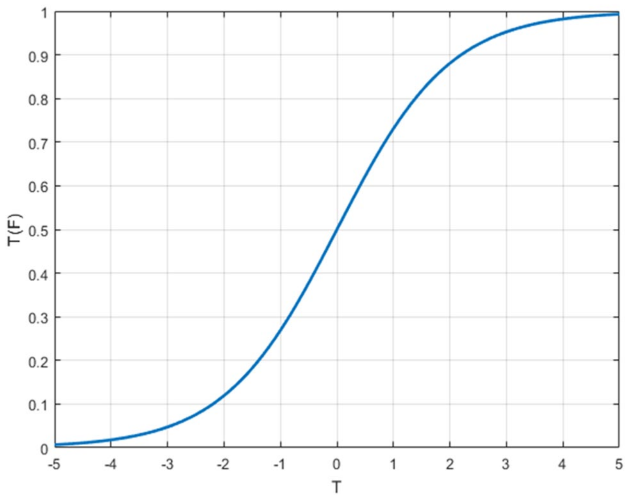


Fig. 7 The Sigmoid transfer function (*S*-shape)

4.2 The binary HOA

A binary edition of HOA according to the Sigmoid function will be introduced in this section. As previously stated, the processes of the original HOA move in continuous space; hence all potential solutions in this algorithm's population include continuous numbers. Given that the goal of discrete problems is selecting or not selecting a particular feature, the new binary solution should contain the numbers zero and one, with one indicating selecting a feature for the new data set and zero indicating not selecting that. To implement this, the HOA processes will be moved in discrete space using the Sigmoid function (S-shaped) [8, 27]. As a result, the Sigmoid function is used in the proposed model to shift the continuous position of solutions in the binary HOA, according to Eq. (22):

$$sg\left(\mathrm{HOA}_i^d(t)\right) = \frac{1}{1 + \mathrm{e}^{-\mathrm{HOA}_i^d(t)}} \tag{22}$$

In Eq. (22), HOA_i^d represents the continuous value of i^{th} solution in the HOA population in d^{th} dimension in t^{th} iteration. The Sigmoid function is depicted in Fig. 7. This function's output analysis and how it's integrated into HOA will be detailed further below.

Since the output of the Sigmoid transfer function is still continuous between zero and one, a threshold must be specified to transform it into a binary number. The random threshold shown in Eq. (23) is applied in the Sigmoid function to convert to the binary value solution for feature selection:

$$HOA_i^d(t+1) = \begin{cases} 0\, if\ \ \mathrm{rand} < sg\left(HOA_i^d(t)\right) \\ 1\, if\ \ \mathrm{rand} \geq sg\left(HOA_i^d(t)\right) \end{cases} \tag{23}$$

In the above Equation, HOA_i^d represents the i^{th} solution's position in the HOA population in d^{th} dimension in t^{th} iteration. The *rand* is a number of a uniform distribution type that ranges from 0 to 1. Therefore, Eqs. (22) and (23) are used to force solutions in the HOA population to move in a discrete (binary) search space. These equations are integrated into HOA accurately, and the algorithm's pseudo-code is provided as follows:

Algorithm 1: Binary HOA

```
BHOA: S-Shaped Algorithm
{------Step 1: Initialise the HOA parameters------}
Initialise the parameters (Reduction factors, Percent of best horses, Percentage of worst horses, Grazing of $\alpha,\beta,\gamma,\delta$, horses Defense Mechanism of $\alpha,\beta,\gamma,\delta$, horses and Age).
{------Step 2: Generate population Initially------}
Generate initial population
Defense Candidates
BadPos (Variable) =mean(Cost Counterl( (1-pN) *nHorse:nHorse, 3));
Imitation Candidates
GoodPos (Variable) =mean (CostCounter1(1:9NnHorse, 2+Variable));
Sociability Candidates
MeanPos (Variable) mean (CostCounterl(1:nHorse, 2+Variable));
Calculate the fitness of each search agent and Sorted global matrix based on Cost
CostCounter=sortrows (CostCounter2,2);
{------Step 3: Herd evolution------}
While ($\mathbf{t \le Max\_itr}$) do
Calculating Alpha Horses Velocity
Horse. Velocity.Alpha-g_Alha* (0.95+0.1 rand) * (Horse. Best. Position-Horse. Position) - Alpha*rand. + (BadPos-Horse. Position);
Calculating Beta Horses Velocity
Horse. Velocity. Beta-g_Beta (0.95+0.1*rand) - (Horse. Best. PositionHorse. Position)... +h_Beta rand. (GlobalBest. Position-Horse. Position) +s Beta*rand.* (MeanPos-Horse. Position)... -d Beta rand." (BadPos-Horse. Position);
Calculating Gamma Horses Velocity
 Horse. Velocity. Gamma=g_Gamr.a* (0.95+0.1*rand) * (Horse. Best. PositionHorse. Position)... th Gamma *rand.* (GlobalBest. Position-Horse. Position)s Gamma rand. + (Mean Pas-Horse. Position).. +i_Gamma *rand. (GoodPos-Horse. Position)... -d Gamma*rand.* (BadPos-Horse. Position)... +r_Gamma. (Horse. Velocity);
   Calculating Delta Horses Velocity
Horse. Velocity. Delta=g_Delta* (0.95+0.1*rand) * (Horse. Best. PositionHorse. Position)... +r_Delta. * (Horse. Velocity);
   End for
{------Step 4: Update population ------}
  For i=1: N
      Convert Continuous($X_{inew}$) to Binary($BX_{inew}$) using transfer function according to Equations (22) and (23)
   End for
         Evaluation of All-New Solutions
         For all solutions
            If ($Q > rand$)
               Changes of Solution according to Equation (10).{part: $Q > rand$ }
            else
               Changes of Solution according to Equation (10).{part: else}
            End if
         End for
 for i=1: N
      Convert Continuous($X_{inew}$) to Binary($BX_{inew}$) using transfer function according to Equations (22) and (23)
  End for
         Evaluation of All-New Solutions
End while
Print Best Solution
```

Defense Candidates: *BadPos(Variable)=mean(CostCounterl((l-pN)*nHorse:nHorse,3));*
Imitation Candidates: *GoodPos(Variable)=mean(CostCounter1(1:qN*nHorse,2+Variable));*
Sociability Candidates: *MeanPos(Variable)=mean(CostCounter1(1:nHorse,2+Variable));*
In which:
Sorted global matrix based on Cost: *CostCounter=sortrows(CostCounter2,2);*
and
CostCounter2(Variable,:)=[Variable Horse(Variable).Best.Cost Horse(Variable).Best.Position];

Algorithm 1 presents the first proposed approach to binarise HOA using the Sigmoid function. The initialisation of parameters is the first phase in this algorithm (lines 01:02). In the second phase (lines 03:08), the population is generated with 0 and 1 at random, in contrast to the continuous HOA. The third phase of the algorithm (lines 09:15) includes the main steps of the continuous HOA and is the main loop of improvement of HOA. The fourth phase (lines 16:19) is a new stage to binarise the generated solutions in the earlier phases using the Sigmoid transfer function using Eqs. (22) and (23). All continuous solutions are transformed into binary solutions at this phase. Lines 20 to 27 represent the evaluation and updating of new solutions, leading to generating of new solutions. Lines 28 to 30 include the new phase of binarising the new solutions which are generated in the previous phases by the Sigmoid transfer function using Eqs. (22) and (23). All continuous solutions are transformed into binary solutions at this phase. Finally, in lines 31 to 33, the new binary solutions are evaluated. In case the algorithm reaches the final condition, the best solutions will be displayed. As a result, in the first proposed approach, a new binary algorithm is developed by placing lines 16 to 9 and 28 to 30 in two different parts of HOA and applying the Sigmoid transfer function. Because HOA evaluates and fundamentally changes the solutions in two steps, the Sigmoid transfer function is used in two sections of the algorithm.

4.3 The multi-objective binary HOA

The feature selection objective function for the new algorithm and other metaheuristic methods covered in the current study is described in this section. Feature selection problem is regarded as multi-objective optimisation, as its two primary objectives, decreasing the number of the selected features and increasing the precision of the classification, are often at odds with each other [39]. Thus, a classification algorithm is required to specify the problem's objective function [25]. In most cases, the K-Nearest Neighbor (KNN) classifier can be employed for this purpose. Therefore, in the proposed algorithm, KNN is used to specify the objective function for feature selection. KNN classifier evaluates the selected features by the new algorithm. The proposed multi-objective function is applied for evaluating each solution, and it relies on the classifier. In the new multi-objective function, to minimise the number of selected features in each solution and to maximise the accuracy of the classification, the fitness function of Eq. (24) is employed for evaluating the solutions in any metaheuristic algorithm [37]:

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|N|} \tag{24}$$

In Eq. (24), $\alpha\gamma_R(D)$ is the error rate of the classifier, $|R|$ is the multi-linearity of the selected subset, and $|N|$ is the overall number of features within the data set $\cdot$ α and β are the significance of the classification's quality and the length of the subset, respectively, where $\alpha \in [0, 1]$ and $\beta = (1–\alpha)$ [19]. The initial value of α is considered 0.99 in the current study; as a result, β will be calculated equal to 0.01.

Since the original algorithm used in this study was named Horse herd Optimisation Algorithm, the improved version of HOA is named Multi-objective Binary Horse herd Optimisation Algorithm (MBHOA).

5 Simulation and results

5.1 Result of CEC 2018 evaluation functions

Understanding the evolutionary algorithms' strengths and weaknesses relies heavily on benchmark functions [13]. Metaheuristic algorithms' performance varies, and most of them are evaluated using standard test functions like Schwefel, Sphere, Rosenbrock, Rastrigin, and others, which have some problems, for instance:

- Some of these functions are confusing
- These functions have some limitations
- The selected test function and algorithm may complement each other
- A test function may not work well on a certain problem

Therefore, to evaluate new algorithms, more systematic methods with fewer problems are required. CEC 2018 is a set of test functions introduced by Cheng et al. [13] with various characteristics that greatly represent real-life situations and scenarios. The aim is to improve optimisation research by developing a set of benchmark functions that accurately replicate various real-world settings. Table 2 shows the characteristics of those 15 test functions used in the current study. In these functions, $X = \left[x_1, x_2, \ldots, x_D\right]$ is the decision vector, D is the number of variables of decision-making, and M indicates the number of objectives.

In order to make a fair comparison between all simulated algorithms, the domain was 50, and up to 20 comparisons of the algorithms' computational cost were considered. The time consumed by each algorithm was used for the comparison. Therefore, the parameters that affect the simulation implementation and the results obtained are 20 iterations and 50 variables. To make the simulation results reliable and to reduce the effect of random simulation results, each function is repeated 50 times.

Table 2 Characteristics of 15 test functions (adapted from [13])

Problem	Properties
MaF1	Linear
MaF2	Concave
MaF3	Convex, Multimodal
MaF4	Concave, Multimodal
MaF5	Concave, Biased
MaF6	Concave, Degenerate
MaF7	Mixed, Disconnected, Multimodal
MaF8	Linear, Degenerate
MaF9	Linear, Degenerate
MaF10	Mixed, Biased
MaF11	Convex, Disconnected, Nonseparable
MaF12	Concave, Nonseparable, Biased Deceptive
MaF13	Concave, Unimodal, Nonseparable, Degenerate
MaF14	Linear, Partially separable, Large scale
MaF15	Convex, Partially separable, Large scale

The proposed algorithm was implemented in MATLAB. Similarly, all 15 test functions were implemented in MATLAB and integrated into PlatEMO, a newly designed platform. PlatEMO is a MATLAB-based open-source software platform for evolutionary multi-objective optimisation [13]. In the simulation, it is initialised as:

– *Number of objectives (M): 2*
– *Maximum population: 50*
– *Number of iterations: 20*

To reflect its performance, the proposed method was compared with three other algorithms, Binary Social Spider Algorithm (BSSA) [21], the Binary version of Grey Wolf Optimization (BGWO) [20], the Binary Butterfly Optimization Algorithm (BBOA)[8], as well as the original HOA. The simulation results of the MBHOA algorithm and the compared algorithms are presented in Table 3.

In Table 3, the mean is the average error observed by each algorithm with 20 iterations per function, and the standard deviation is the amount of variability, or dispersion, from the individual data values to the mean. Mean, and standard deviation are calculated using CEC 2018 test functions to show the ability of the proposed algorithm compared to other algorithms in optimal results. If the standard deviation of a set of data is close to zero, it indicates that the data are close to the mean and have little scatter, while a large standard deviation indicates significant data scatter. The standard deviation is equal to the square root of variance. The good thing about variance is that it also measures data. Standard deviation is usually more useful than analysis of variance in statistical data analysis and indicates the extent of data scatter. The lower the standard deviation, the lower the data scatter. Here, the mean is the average error observed by each algorithm, which, again, the closer the error is to zero, the better the performance of the algorithm. The closer the value of these two variables is to zero, the better the method's performance. It is noteworthy that in Table 3, the best results are shown in bold.

According to the results presented in Table 3, MBHOA shows the best performance based on the average error in most functions except MaF6 and MaF13. However, in MaF6 and MaF13 functions, it competes closely with the best-performing algorithms. Meanwhile, in the MaF1 function, the average error of the original HOA algorithm (mean) is better than the proposed algorithm, while in the same function, the standard deviation result of the proposed method is less than all other algorithms. In the MaF12 function, the standard deviation function of the BSSA algorithm is better than the proposed algorithm, while in the same function, the average error of the proposed method is less than all algorithms. It can be seen that the standard deviation of the BSSA algorithm is better than other algorithms in MaF6, MaF12 and MaF13 functions which show good stability. According to the results, the BBOA algorithm shows the lowest performance among the compared algorithms in all of the functions. The BSSA and BGWO algorithms demonstrate acceptable performance and have a good convergence rate compared to BBOA.

Based on the simulation results, the proposed algorithm, MBHOA, shows the highest performance compared to all other algorithms except in two functions, MaF6 and MaF13. The BSSA, BGWO and BBOA algorithms rank second to fourth, respectively, in terms of performance. Therefore, it can be inferred that the MBHOA algorithm is superior to other algorithms based on the accuracy of the generated solutions, stability, convergence speed and success rate. This is due to the algorithm's excellent balance between exploration and exploitation.

For a better comparison of the methods, the diagrams of the compared algorithms for the above 15 evaluation functions are shown in Fig. 8. In these diagrams, the value of the fitness function obtained is used to evaluate the performance of the algorithm. As can be seen, the proposed MBHOA method performs better than the other methods in most cases because the mean values of the evaluation function for the MBHOA are lower. As a result, the overall performance of MBHOA is better than other compared optimisation techniques.

As can be seen in the graphs, the performance of MBHOA is better in all evaluation functions except MaF6 and MaF13. In MaF6 and MaF13 functions, the performance of the BSSA method is better than MBHOA.

Due to the diversity of control parameters based on the behaviour of horses at different ages, the HOA algorithm has

Table 3 Mean and fit standard deviation values obtained for the 15 CEC 2018 function

	BSSA		BGWO		BBOA		HOA		MBHOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
MaF1	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	**6.35E-03**	4.73E + 10	3.98E-01	**3.98E-01**
MaF2	8.00E-02	4.03E-03	4.80E-02	6.25E-03	3.89E-49	3.64E-58	7.43E + 03	1.57E + 02	**1.91E-59**	**2.90E-71**
MaF3	1.89E + 02	5.65E + 01	2.80E + 04	1.49E + 04	7.39E + 04	2.05E + 04	2.26E + 01	1.51E − 02	**2.27E-06**	**6.32E-23**
MaF4	1.73E + 00	6.26E-01	7.87E + 01	4.38E + 01	8.38E + 01	5.22E + 00	3.29E + 11	6.14E + 10	**9.12E-26**	**9.68E-32**
MaF5	4.25E + 02	1.37E + 01	1.85E + 02	1.79E + 01	2.86E + 01	2.69E + 01	5.74E + 06	2.24E + 05	**4.80E + 00**	**2.46E-03**
MaF6	**5.91E-04**	**1.00E-05**	1.15E-02	1.02E-04	1.21E + 00	8.31E-02	1.16E + 06	2.21E + 03	3.31E-01	1.46E-04
MaF7	4.58E-01	7.48E-02	7.38E-01	1.17E-01	2.21E-02	6.38E-04	– 1.21E + 04	– 1.21E + 04	**3.73E-03**	**3.12E-05**
MaF8	– 2.96E + 03	– 7.81E + 03	– 1.08E + 04	– 1.21E + 04	– 7.69E + 03	– 1.26E + 04	9.66E + 15	1.49E + 13	– **1.26E + 04**	– **1.26E + 04**
MaF9	7.62E + 01	3.19E + 01	1.41E + 01	2.19E + 00	5.68E-14	0.00E + 00	3.18E-02	3.18E-02	**0.00E + 00**	**0.00E + 00**
MaF10	1.65E + 00	2.50E-03	1.37E + 00	3.18E-02	7.99E-15	8.88E-16	9.73E-04	1.70E + 05	**4.44E-15**	**8.88E-16**
MaF11	3.44E-02	6.35E-07	2.81E-01	9.73E-04	2.61E-01	0.00E + 00	3.65E + 07	4.45E + 08	**0.00E + 00**	**0.00E + 00**
MaF12	1.04E-01	**7.59E-08**	5.65E-03	1.52E-05	4.33E-02	4.71E-03	8.67E + 09	1.07E + 09	**2.95E-03**	5.00E-06
MaF13	5.48E-02	**5.04E-06**	**1.20E-02**	9.03E-05	9.66E-01	1.35E + 10	7.66E + 10	9.98E-01	1.68E-02	3.68E-05
MaF14	7.87E + 00	9.98E-01	9.98E-01	9.98E-01	1.08E + 01	9.98E-01	1.00E + 12	1.79E + 11	**9.98E-01**	**9.98E-01**
MaF15	1.15E-03	4.07E-04	1.46E-03	4.77E-04	4.45E-03	3.08E-04	1.00E + 08	6.75E + 06	**5.95E-04**	**3.08E-04**

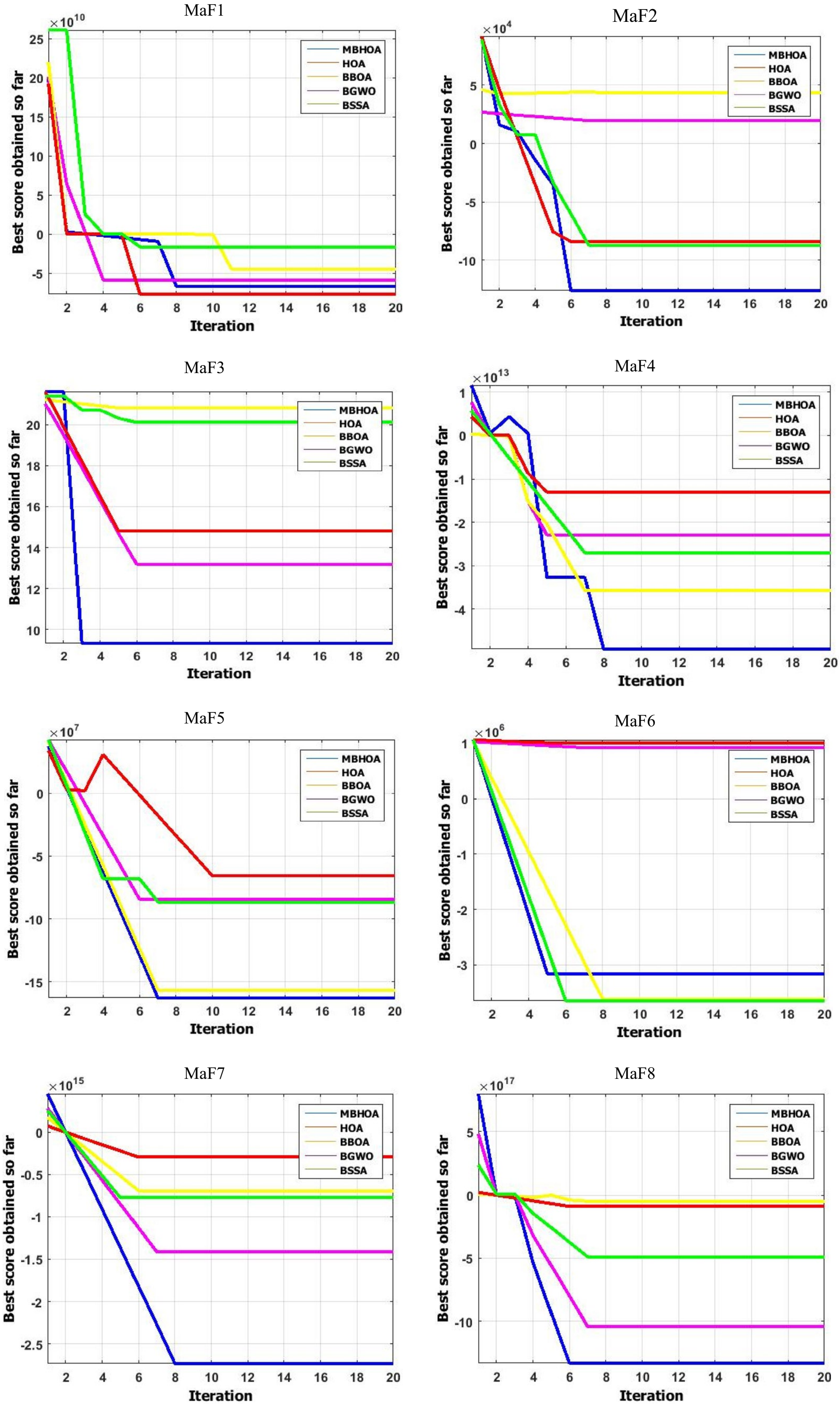


Fig. 8 Convergence diagrams for CEC 2018 benchmark functions with 20 iterations

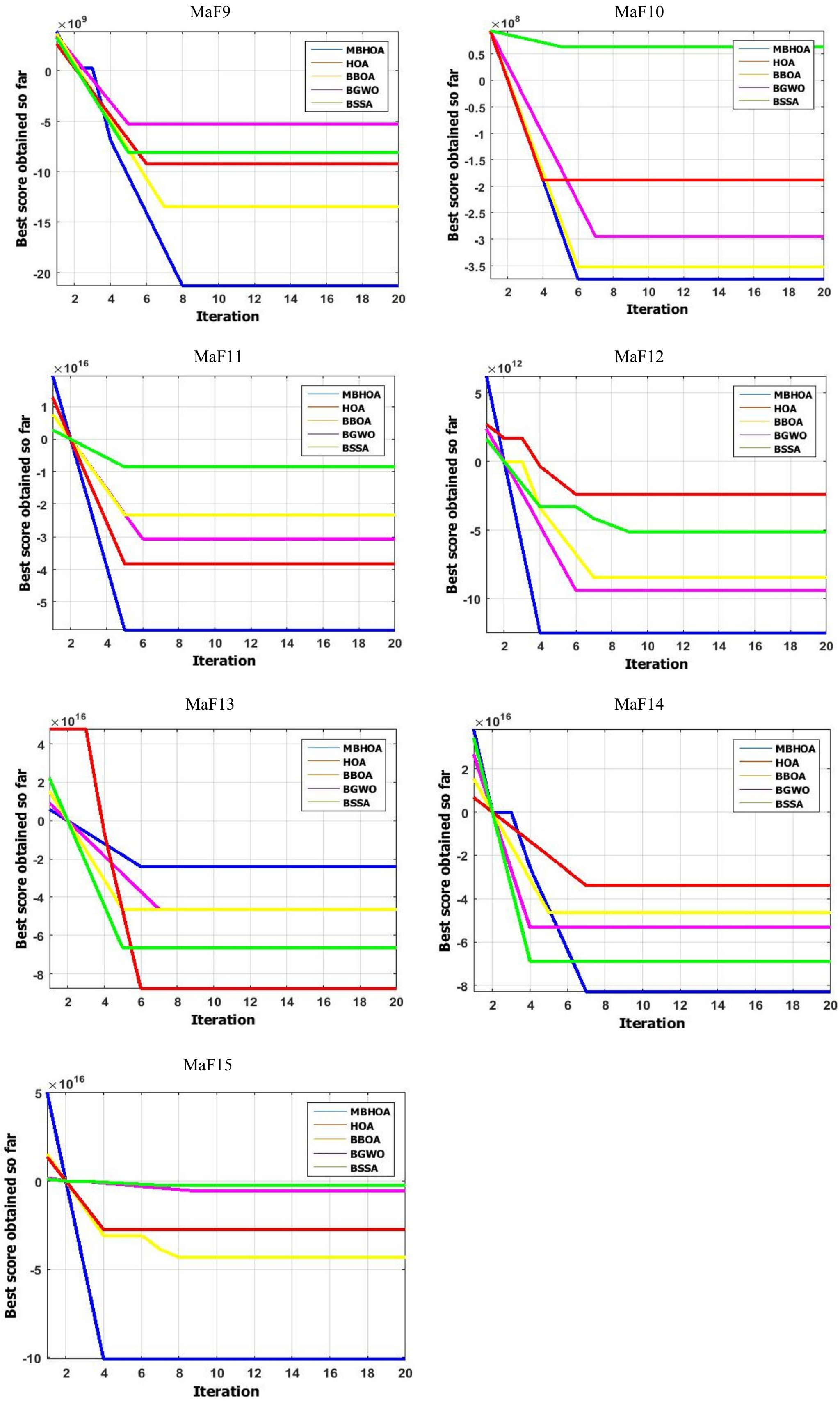


Fig. 8 (continued)

a very good performance in addressing simple and complex problems. It outperforms the strongest well-known optimisation algorithms. This algorithm has a better performance in solving complex problems with many unknown variables in high-dimensional spaces. The original HOA optimisation algorithm is able to solve only single-objective problems. After converting the original algorithm to a multi-objective algorithm in the current study, the new HOA has the ability to solve any single-objective and multi-objective complex optimisation problems in high dimensions. On the other hand, the original HOA algorithm is a continuous algorithm and is able to address only continuous optimisation problems. In this study, it is binarised; therefore, it has the capability to solve any discrete optimisation problems. Overall, the original HOA is superior in solving single-objective continuous complex problems, and the improved HOA is superior in solving multi-objective and discrete complex problems. Thus, the proposed MBHOA algorithm outperforms the well-known state-of-the-art methods in addressing a wide range of real-life and engineering optimisation problems.

The new algorithm's performance highly depends on the settings of the algorithm's parameters; in order to achieve the best performance, the best value for each of the parameters has been selected in the simulations in the current study. In the case of selecting less-than-optimal values for the algorithm's parameters, the result may not be optimal.

5.2 Practical example of the proposed method

A practical example is presented in the current section for the proposed MBHOA algorithm. The algorithm is employed for solving a feature selection problem. Text sentiment analysis is difficult for computers to solve, due to the computer's inability to comprehend the users.

A second objective of this study was to design a text sentiment analysis technique and to introduce a novel approach to address this problem. To date, various techniques have been used for text sentiment analysis; however, in this paper, the improved HOA (MBHOA) is utilised for solving the feature selection problem in sentiment analysis. This could be considered a new practical example of the application of MBHOA.

In general, the proposed approach consists of five key steps for classification, as indicated in Fig. 9. The data sets used in this study are standard, and the feature extraction and normalisation phases are performed and provided alongside the data sets. The data sets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Feature reduction's primary aim would be to extract a small collection of features among a large number of features in a problem. The extracted features include redundant, irrelevant, misleading, and noisy features. Eliminating the data impacting the precision of the classifiers and the prediction could be beneficial [18]. MBHOA principles are used for the optimal selection of features. Each feature is considered a position that is presented by MBHOA. A feature subset that has the highest precision of classification would be the optimal position. At this stage, the multi-objective improved HOA algorithm is used. The training phase is conducted using a machine learning method by providing the objective function as well as the texts. A two-layer neural network is utilised to calculate the cost function, with the hidden layer being Sigmoid with ten neurones and the second layer being linear with one neurone [24]. For the training network in the proposed approach, the Levenberg–Marquardt algorithm is employed. The network takes the objective function as well as a set of texts as the input, builds and trains a neural network, and then returns the outcomes. It can be used to build a cost function with two objectives: the percentage of errors and the quantity of features. The output of the MBHOA algorithm was compared to the output of five similar algorithms, Mutual Information Features (MIFS) K-Nearest Neighbour (KNN), ‘RoBERTa-based Aspect-Category Sentiment Analysis’ (RACSA), ‘Cross-Domain Sentiment

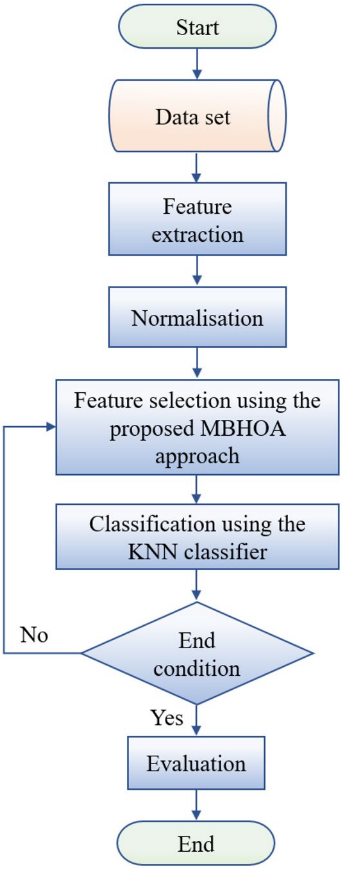

Start
Data set
Feature extraction
Normalisation
Feature selection using the proposed MBHOA approach
Classification using the KNN classifier
End condition
No
Yes
Evaluation
End

Fig. 9 Flowchart of the practical example

Table 4 Performance evaluation criteria and their equations

	Evaluation criterium	Equation
1	Accuracy	$\frac{tp+tn}{N}$
2	Sensitivity(recall)	$\frac{tp}{tp+fn}$
3	Specificity	$\frac{tn}{tn+fp}$
4	Precision	$\frac{tp}{tp+fp}$
5	*f*-measure	$\mathrm{f} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} \times \mathrm{recall}}$

Table 5 Summary of the used data sets [28]

Data set	Overall	Negative	Positive	Neutral
Twitter2016-train	6000	3094	863	2043
Twitter2016-dev	2000	844	765	391
Twitter2016-test	20,632	7059	10,342	3231
Twitter2017-test	12,284	2375	3972	5937
SMS2013	2093	492	394	1207
Tw2014-sarcasm	86	33	40	13
LiveJournal2014	1142	427	304	411

Aware Word Embeddings' (CDSAWE), and 'Membrane-Driven Hybrid Krill Herd Algorithm' (MHKHA).

After selecting the important features, the texts are classified based on sentiment analysis. In a classification of sentiments, we can consider seven classes of emotions (happiness, fear, anger, sadness, hatred, shame, and guilt). In this phase, the KNN classifier is employed.

For evaluating the proposed method, five criteria are considered that could be used as the simulation variables. The most important evaluation criteria that are defined based on the other four variables (*fn, tn, fp, tp)* are accuracy, sensitivity, specificity, precision, and f-measure [9]. The Equation for each of these criteria is presented in Table 4.

5.3 Evaluation results of the practical example

The proposed algorithm is implemented in MATLAB version R2014a on a computer with a 64-bit core i5 processor and 4 GB of memory to evaluate the practical example using the proposed approach. Adapted from the study of Júnior, Marinho, and dos Santos [28], the provided training and development data set by SemEval-2017 (specifically Twitter2016-train and Twitter2016-dev) was used to evaluate the method. For the testing, in addition to the test data set (Twitter2016-test, SMS2013, Tw2014-sarcasm, LiveJournal2014), another data set (Twitter2017-test) was also used [28]. Table 5 demonstrates a summary of the used data sets for training and evaluation of the proposed method.

Only Twitter2016-train and Twitter2016-dev were utilised in training in the Twitter2016-test evaluation. Twitter2016-train, Twitter2016-dev, and Twitter2016-test were used in the remaining evaluations. Although other data sets were available for training, only the above three were selected.

The performance of the proposed MBHOA method's classifier in feature selection was compared to five existing and recent approaches, namely Mutual Information Features (MIFS) [11], K-Nearest Neighbour (KNN), RoBERTa-based Aspect-Category Sentiment Analysis (RACSA) [31], Cross-Domain Sentiment Aware Word Embeddings (CDSAWE) [33], and Membrane-Driven Hybrid Krill Herd Algorithm (MHKHA) [3]. Table 6 demonstrates the performance comparison, as well as the superiority of the MBHOA's classifier in feature selection. As the results indicate, the proposed method has been disrupted in data sets with different origins, such as SMS2013; this may have occurred due to the use of a specific vocabulary. In the case of Tw2014, the main problem is the order of the words in the sentence, making it difficult to identify irony or modifiers. In the LiveJournal2014 data set, this proposed method remains stable even if it is a collection from another domain, probably because it resembles the Twitter database. The overall results indicate that MBHOA outperforms the other algorithms in feature selection.

Table 7 compares the outcomes of the MBHOA with other similar approaches, GA, BBOA, BSSA, BALO, BGWO, BDA, and MHKHA, in terms of the average number of selected features for sentiment analysis on the Twitter2017 data set (the best results are indicated in bold). This approach's performance is proven in the majority of the iterations. The results show that MBHOA performs exceptionally well in selecting features.

The adult horses (α) start local searching around the global optimum. β horses are also inclined to move towards them, looking for other close positions around α horses. γ horses, on the other hand, are less enthusiastic about moving towards α horses and are more interested in exploring new areas and discovering the best potential locations in the global optimum. Young horses (δ) are accustomed to chaos, making them perfect candidates for the random search step due to their distinct behavioural traits. In the shortest amount of time, HOA can discover the optimal solution [35].

5.4 Limitations

Feature selection reduces the cost of computation. By removing useless features, the model becomes clearer and more comprehensive. It also speeds up the training process, reduces storage space and improves performance, such as accuracy and precision. As a result, feature selection techniques are essential to reduce data dimensions in high-dimensional data. One of the best ways to implement feature selection is to employ metaheuristic optimisation

Table 6 Comparison of the performance of the new method’s classifier with MIFS, KNN, RACSA, CDSAWE and MHKHA

Data set		Twitter2016-test	Twitter2017-test	SMS2013	Tw2014-sarcasm	LiveJournal2014
Evaluation criteria	Classification method					
F-Measure						
MIFS		0.542	0.610	0.439	0.654	0.712
KNN		0.672	0.682	0.597	0.721	0.748
RACSA		0.803	0.812	0.752	0.792	0.815
CDSAWE		0.822	0.818	0.759	0.787	0.843
MHKHA		0.842	0.836	0.761	0.812	0.862
MBHOA		0.863	0.852	0.763	0.823	0.893
MIFS		0.527	0.612	0.494	0.536	0.569
KNN		0.682	0.643	0.536	0.657	0.765
RACSA		0.795	0.821	0.732	0.784	0.796
Recall						
CDSAWE		0.812	0.841	0.754	0.855	0.799
MHKHA		0.828	0.849	0.761	0.860	0.818
MBHOA		0.825	0.853	0.755	0.863	0.812
MIFS		0.644	0.734	0.653	0.725	0.743
KNN		0.762	0.853	0.785	0.822	0.822
RACSA		0.851	0.832	0.742	0.791	0.839
Precision						
CDSAWE		0.864	0.854	0.801	0.834	0.854
MHKHA		0.887	0.893	0.811	0.875	0.894
MBHOA		0.891	0.925	0.823	0.912	0.931
MIFS		0.542	0.617	0.609	0.442	0.588
KNN		0.632	0.753	0.724	0.642	0.635
RACSA		0.852	0.875	0.818	0.875	0.901
Accuracy						
CDSAWE		0.864	0.841	0.831	0.892	0.886
MHKHA		0.843	0.855	0.843	0.924	0.942
MBHOA		0.931	0.946	0.875	0.942	0.964

Table 7 Comparison of the average number of selected features by MBHOA and other approaches

Iteration	GA	BBOA	BSSA	BALO	BGWO	BDA	MHKHA	MBHOA
20	1941	2487	2381	2212	1833	1868	**1812**	**1712**
40	1885	2120	1935	2145	1907	1887	**1721**	**1632**
60	1850	1920	1896	2014	**1754**	2138	1864	1973
80	1853	1865	1869	2352	1865	1821	1756	**1633**
100	1815	1802	1803	1959	1721	2074	1710	**1701**

algorithms; however, these algorithms have a high computation cost compared to other types of computational intelligence algorithms. Moreover, they rely heavily on classification methods. Therefore, as a metaheuristic algorithm, the proposed method is not an exception to this limitation. Although it shows the lowest computational complexity compared to the well-known existing metaheuristic methods, its computational complexity could still be higher than other stochastic or deterministic methods. This can be considered a drawback to the proposed method. The performance of the proposed algorithm also depends on the classification method that is used for sentiment analysis, which could be considered a weakness of the algorithm. On the positive side, the proposed algorithm has higher accuracy and precision in comparison with existing methods. Another weakness of the available methods for analysing sentiments is that they have difficulty recognising sarcastic and misleading statements. Despite this shortcoming, the proposed method has a lower error rate than the compared methods.

6 Conclusion

In this paper, HOA was improved by adding the seventh social behaviour of horses, and a new update of the algorithm was presented, which is binary and multi-objective. MBHOA demonstrated its ability to solve complicated optimisation problems. This algorithm was tested in high dimensions and proved that it is very efficient in terms of exploration and operation. Examining the algorithm's operators, it seemed that the low convergence rate is relevant to the division of horses into different age groups. Due to the age division of the population, MBHOA has an exceptional potential to create balance, and this balance improves the efficiency of the algorithm's computational complexity. This algorithm was used for feature selection in text sentiment analysis to test its performance, and the findings suggest that it performs well compared to other similar methods.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

References

1. Abbasi A, Chen H, Salem A (2008) Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. ACM Trans Inform Sys (TOIS) 26(3):1–34
2. Abbattista, F., Abbattista, N., & Caponetti, L. (1995). An evolutionary and cooperative agents model for optimization. Paper presented at the Proceedings of 1995 IEEE International Conference on Evolutionary Computation.
3. Abualigah L, Alsalibi B, Shehab M, Alshinwan M, Khasawneh AM, Alabool H (2021) A parallel hybrid krill herd algorithm for feature selection. Int J Mach Learn Cybern 12(3):783–806
4. Abualigah LM, Khader AT, Al-Betar MA, Alomari OA (2017) Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Expert Syst Appl 84:24–36
5. Agarwal, B., & Mittal, N. (2013). Sentiment classification using rough set based hybrid feature selection. Paper presented at the Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.
6. Al-Betar MA, Alyasseri ZAA, Awadallah MA, Doush IA (2021) Coronavirus herd immunity optimizer (CHIO). Neural Comput Appl 33(10):5011–5042
7. Arafat H, Elawady RM, Barakat S, Elrashidy NM (2014) Different feature selection for sentiment classification. Int J Inform Sci Intell Syst 1(3):137–150
8. Arora S, Anand P (2019) Binary butterfly optimization approaches for feature selection. Expert Syst Appl 116:147–160
9. Azar AT, Elshazly HI, Hassanien AE, Elkorany AM (2014) A random forest classifier for lymph diseases. Comput Methods Programs Biomed 113(2):465–473
10. Banati H, Bajaj M (2011) Fire fly based feature selection approach. Int J Comput Sci Issues (IJCSI) 8(4):473
11. Bennasar M, Hicks Y, Setchi R (2015) Feature selection using joint mutual information maximisation. Expert Syst Appl 42(22):8520–8532
12. Bogner, F. (2011). A comprehensive summary of the scientific literature on Horse Assisted Education in Germany. Van Hall Larenstein,
13. Cheng, R., Li, M., Tian, Y., Xiang, X., Zhang, X., Yang, S., Yao, X. (2018). Benchmark functions for the cec'2018 competition on many-objective optimization. Retrieved from
14. De Stefano, C., Fontanella, F., Marrocco, C., & Schirinzi, G. (2007). A feature selection algorithm for class discrimination improvement. Paper presented at the 2007 IEEE International Geoscience and Remote Sensing Symposium.
15. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186(2–4):311–338
16. Diao R, Shen Q (2012) Feature selection with harmony search. IEEE Trans Systs Man and Cybern Part B 42(6):1509–1523
17. Dong H, Li T, Ding R, Sun J (2018) A novel hybrid genetic algorithm with granular information for feature selection and optimization. Appl Soft Comput 65:33–46
18. ElSoud MA, Anter AM (2016) Computational intelligence optimization algorithm based on meta-heuristic social-spider: case study on CT liver tumor diagnosis. Comput Intell 7(4):466–475
19. Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. Neurocomputing 213:54–65
20. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381
21. Emine B, Ülker E (2020) An efficient binary social spider algorithm for feature selection problem. Expert Syst Appl 146:113185
22. Forsati R, Moayedikia A, Keikha A, Shamsfard M (2012) A novel approach for feature selection based on the bee colony optimization. Int J Comput Appl 43(8):30–34
23. Hosseinalipour A, Gharehchopogh FS, Masdari M, Khademi A (2021) A novel binary farmland fertility algorithm for feature selection in analysis of the text psychology. Appl Intell. https://doi.org/10.1007/s10489-020-02038-y
24. Hosseinalipour, A., Gharehchopogh, F. S., Masdari, M., & Khademi, A. (2021). Toward text psychology analysis using social spider optimization algorithm. Concurrency and Computation: Practice and Experience, e6325
25. Houssein, E. H., Mahdy, M. A., Shebl, D., & Mohamed, W. M. (2021). A Survey of Metaheuristic Algorithms for Solving Optimization Problems. In Metaheuristics in Machine Learning: Theory and Applications, Springer. pp. 515–543
26. Hu B, Dai Y, Su Y, Moore P, Zhang X, Mao C, Xu L (2016) Feature selection for optimized high-dimensional biomedical data using an improved shuffled frog leaping algorithm. IEEE/ACM Trans Comput Biol Bioinf 15(6):1765–1773
27. Hussien, A. G., Hassanien, A. E., Houssein, E. H., Bhattacharyya, S., & Amin, M. (2019). S-shaped binary whale optimization algorithm for feature selection. In Recent trends in signal and image processing (pp. 79–87): Springer.
28. Júnior, E. A. C., Marinho, V. Q., & dos Santos, L. B. (2017). NILC-USP at semeval-2017 task 4: A multi-view ensemble for twitter sentiment analysis. Paper presented at the Proceedings

of the 11th International Workshop on Semantic Evaluation (SemEval-2017).

29. Kanan, H. R., Faez, K., & Taheri, S. M. (2007). Feature selection using ant colony optimization (ACO): a new method and comparative study in the application of face recognition system. Paper presented at the Industrial conference on data mining.
30. Krueger K, Heinze J (2008) Horse sense: social status of horses (Equus caballus) affects their likelihood of copying other horses' behavior. Anim Cogn 11(3):431–439
31. Liao W, Zeng B, Yin X, Wei P (2021) An improved aspect-category sentiment analysis model for text sentiment analysis based on RoBERTa. Appl Intell 51(6):3522–3533
32. Liu B (2012) Sentiment analysis and opinion mining. Syn Lect Human Lang Technol 5(1):1–167
33. Liu J, Zheng S, Xu G, Lin M (2021) Cross-domain sentiment aware word embeddings for review sentiment analysis. Int J Mach Learn Cybern 12(2):343–354
34. Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453
35. MiarNaeimi F, Azizyan G, Rashki M (2021) Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. Knowl-Based Syst 213:106711
36. Mishev K, Gjorgjevikj A, Vodenska I, Chitkushev LT, Trajanov D (2020) Evaluation of sentiment analysis in finance: from lexicons to transformers. IEEE Access 8:131662–131682
37. Mohammadzadeh H, Gharehchopogh FS (2021) 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
38. Mousavirad, S., & Ebrahimpour-Komleh, H. (2013). Feature selection using modified imperialist competitive algorithm. Paper presented at the ICCKE 2013.
39. Nguyen, H. B., Xue, B., Ishibuchi, H., Andreae, P., & Zhang, M. (2017). Multiple reference points MOEA/D for feature selection. Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference Companion.
40. Räsänen, O., & Pohjalainen, J. (2013). Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from *speech.* Paper presented at the Interspeech.
41. Rothlauf, F. (2011). Design of modern heuristics: principles and application: Springer Science & Business Media.
42. Sayed SA-F, Nabil E, Badr A (2016) A binary clonal flower pollination algorithm for feature selection. Pattern Recogn Lett 77:21–27
43. Touhidi, H., Nezamabadi-pour, H., & Saryazdi, S. (2007). Feature selection using binary ant algorithm. Paper presented at the Frist joint congress on fuzzy and intelligent systems.
44. Trojovská, E., Dehghani, M., & Trojovský, P. (2022). Zebra Optimization Algorithm: A New Bio-inspired Optimization Algorithm for Solving Optimization Algorithm. *IEEE Access*.
45. Waring, G. H. (1983). Horse behavior. The behavioral traits and adaptations of domestic and wild horses, *including ponies*: Noyes Publications, Mill Road.
46. Yampolskiy RV, El-Barkouky A (2011) Wisdom of artificial crowds algorithm for solving NP-hard problems. Int J Bio-Inspir Comput 3(6):358–369
47. Zhang Y, Gong D-W, Gao X-Z, Tian T, Sun X-Y (2020) Binary differential evolution with self-learning for multi-objective feature selection. Inf Sci 507:67–85
48. Zorarpacı E, Özel SA (2016) A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst Appl 62:91–103