Knowledge-Based Systems 213 (2021) 106711

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems

Farid MiarNaeimi [a,*], Gholamreza Azizyan [a], Mohsen Rashki [b]

[a] Civil Engineering Department, University of Sistan and Baluchestan, Zahedan, Iran
[b] Department of Architecture Engineering, University of Sistan and Baluchestan, Zahedan, Iran

ARTICLE INFO

Article history:
Received 26 May 2020
Received in revised form 10 November 2020
Accepted 17 December 2020
Available online 24 December 2020

Keywords:
Horse's life
Swarm intelligence
Meta-heuristic
High dimension
Global optimization

ABSTRACT

This paper proposes a new meta-heuristic algorithm inspired by horses' herding behavior for high-dimensional optimization problems. This method, called the Horse herd Optimization Algorithm (HOA), imitates the social performances of horses at different ages using six important features: grazing, hierarchy, sociability, imitation, defense mechanism and roam. The HOA algorithm is created based on these behaviors, which has not existed in the history of studies so far. A sensitivity analysis is also performed to obtain the best values of coefficients used in the algorithm. HOA has a very good performance in solving complex problems in high dimensions, due to the large number of control parameters based on the behavior of horses at different ages. The proposed algorithm is compared with popular nature-inspired optimization algorithms, including grasshopper optimization algorithm (GOA), sine cosine algorithm (SCA), multi-verse optimizer (MVO), moth–flame optimizer (MFO), dragonfly algorithm (DA), and grey wolf optimizer (GWO). Solving several high-dimensional benchmark functions (up to 10,000 dimensions) shows that the proposed algorithm is highly efficient for high-dimensional global optimization problems. The HOA algorithm also outperforms the mentioned popular optimization algorithms for the case of accuracy and efficiency with lowest computational cost and complexity.

1. Introduction

Evolutionary Algorithms (EAs) are heuristic mimicking the Darwin principles of evolution [1]. Darwin asked two basic questions in a book entitled "Origin of Species" [2].:

- Are today animals and plants descendants of prehistoric creatures?
- Does one species turn into another one?

This theory is too broad to be explained in this paper. However, as documented, Darwin was aware of the positive responses to these questions [2]. EAs follow the same Darwin principles, in which a population of solutions are evolved (improve) iteratively.

Swarm Intelligence (SI) is another interesting type of intelligence in nature, which refers to the collective behavior of organisms. The SI methods have been one of the most popular research topics during recent years. They mostly focus on the design of an intelligent system with three features (adaptation, dispersion, and flexibility), which provide many opportunities to solve complex problems in various fields. A large number of SI algorithms are based on SI and inspired by the social behavior of animals [3–5].

There is a strong connection between organisms and SI, which is called *interaction* [6]: The collective behaviors of creatures result in SI, and SI changes the conditions. Some examples of such interactive behaviors in nature are as follows:

- Termites build large and complex nests, and this is beyond the understanding and ability of a single termites [7].
- Various tasks in an ant colony are detected automatically with no central management [8].
- Waggle dance of bees is leads them to find more food [9].
- Birds and fish are organized in optimal spatial patterns [10].
- Bacteria follow environmental changes by using molecules [11].

The purpose of computational SI methods is to model the behaviors of creatures and their local interactions with the environment to detect more complex behaviors. These methods can be used to solve complex continuous and discrete optimization problems [12].

The use of meta-heuristic algorithms in solving various problems has increased dramatically in recent years due to the simplicity of understanding and applying such algorithms [3]. Some

* Correspondence to: Civil Engineering Department, University of Sistan and Baluchestan, Zahedan, IR 98155-987, Iran.
E-mail addresses: Farid.Naeimi@pgs.usb.ac.ir (F. MiarNaeimi), g.azizyan@eng.usb.ac.ir (G. Azizyan), mrashki@eng.usb.ac.ir (M. Rashki).

https://doi.org/10.1016/j.knosys.2020.106711
0950-7051/

of the natural evolution phenomena underpinning the meta-heuristic algorithms are the evolution of several generation of creatures, refrigeration or cooling process in metals, the lifecycle of ants in a colony, the migration of birds, human defense system, etc.

There are a variety of meta-heuristic algorithms to address different types of optimization problems, which benefit from two key features, namely exploration and exploitation, to search the optimization problem spaces and to search the good-enough responses. Exploration is the capability of an algorithm to search freely, regardless of the accuracy of results. Exploitation also refers to the performance of an algorithm regarding its achievements in the previous iteration loops. Obviously, the behavior of an algorithm becomes mostly random and unpredictable when the search capabilities increase. Enhanced exploitation ratio in an algorithm, on the contrary, leads to a more cautious performance. The exploration and exploitation capabilities in an algorithm can be controlled since almost all the meta-heuristic search methods possess some adjustable parameters [4].

The main contribution of this study is presenting a novel optimization algorithm that imitates the behaviors of horses. It is shown that the proposed algorithm is applicable to solve the simple and complex single-objective high dimensional problems. This feature is tested by solving seven high-dimensional examples (500, 1000, 2000, 5000, and 10000 dimensions) and the results are then compared with the solutions of the strongest existing optimization algorithms. The rest of this paper is outlined as follows:

An in-depth literature review is presented in Section 2. Section 3 proposes the HOA algorithm, and the results and conclusions are provided in Sections 4 and 5, respectively.

2. Literature review

Meta-heuristic optimization algorithms can be classified in three main categories: Evolutionary algorithms, physic-based algorithms, and SI algorithms.

The first category has been inspired by the idea of evolution in nature. These algorithms are based on the theories of Darwin, as mentioned. This theory is an optimization process, aiming at improving an organisms' ability to survive in a dynamic environment. Living environment is one of the factors, which plays a vital role in determining the best creatures. In other words, several species of organisms living in a particular environment evolve over several generations. Each living creature is considered as an answered question from the nature to solve the (adaption) problem of living in a particular environment.

The second category includes physic-based optimization algorithms, in which the motion of particles is inspired by the laws of physics in magnetic fields, gravitational forces among the particles of galaxies, electron charge transfer, chemical reactions, and so on.

The third category of meta-heuristic optimizers is swarm-based algorithms (SI). These algorithms often use the swarm of particles, in which a single particle fails to be detected and compared. The particles find the response using group communications. A swarm of particles can be a group of agents generally moving and communicating with each other during different activities. The problem-solving behavior of SI is usually derived from studying the social behavior of organisms and their interactions.

A large number of researches have addressed optimization algorithms. In this regard, the most important studies and algorithms are briefly listed in Table 1.

The EA algorithms (and also hybrid algorithms with EAs) have been proposed according to the laws of Darwin's theory, as mentioned in Table 1. Moreover, some of the SI algorithms have been originated from the rules of life, hunting, defense systems, laws of gravity, gradient-based methods [64,65] and so on. Despite the large number of the proposed algorithms, none of the algorithms can solve all optimization problems. This has been logically proved by the No Free Lunch (NFL) theorem. This was an initiative to us to model the social behavior of horses in a herd mathematically and propose a new SI algorithm.

3. Horse Optimization Algorithm (HOA)

This work is based on the behavior patterns of horses in their living environment. The horses' behavior patterns generally include Grazing (G), Hierarchy (H), Sociability (S), Imitation (I), Defense mechanism (D), and Roam (R) [66–68]. So, this algorithm is inspired by the six mentioned general behaviors of horses at different ages. The movement applied to horses at each iteration is according to Eq. (3.1).

$$X_m^{Iter,\ AGE} = \vec{V}_m^{Iter,\ AGE} + X_m^{(Iter-1),\ AGE}, \quad AGE = \alpha, \beta, \gamma, \delta \tag{3.1}$$

where, $X_m^{Iter,AGE}$ indicates the position of mth horse, AGE shows the age range of the considered horse, $Iter$ is the current iteration, and $\vec{V}_m^{Iter,AGE}$ shows the velocity vector of this horse. Horses exhibit different behaviors at different ages. Maximum lifetime of a horse is about 25–30 years [69]. In this regard, δ denotes the horses at the age range of 0–5 years, γ indicates the horses at the range of 5–10 years, β shows the horses the age range of 10–15 years, and α demonstrates the horses older than 15 years. A comprehensive matrix of responses should be conducted per iteration to select the age of horses. In this regard, the matrix can be sorted based on the best responses and consequently, the first 10 percent of the horses from top of the sorted matrix are selected as α horses. The next 20 percent are in the β group. The γ and δ horses account for 30% and 40% of the remaining horses, respectively. The steps to simulate the six behaviors of the horses are mathematically implemented to detect the velocity vector.

The motion vector of horses at different ages during each cycle of the algorithm can be written as Eq. (3.2), with regard to the above behavior patterns [66,67,69].

$$\begin{aligned}
\vec{V}_m^{Iter,\alpha} &= \vec{G}_m^{Iter,\alpha} + \vec{D}_m^{Iter,\alpha} \\
\vec{V}_m^{Iter,\beta} &= \vec{G}_m^{Iter,\beta} + \vec{H}_m^{Iter,\beta} + \vec{S}_m^{Iter,\beta} + \vec{D}_m^{Iter,\beta} \\
\vec{V}_m^{Iter,\gamma} &= \vec{G}_m^{Iter,\gamma} + \vec{H}_m^{Iter,\gamma} + \vec{S}_m^{Iter,\gamma} + \vec{I}_m^{Iter,\gamma} + \vec{D}_m^{Iter,\gamma} + \vec{R}_m^{Iter,\gamma} \\
\vec{V}_m^{Iter,\delta} &= \vec{G}_m^{Iter,\delta} + \vec{I}_m^{Iter,\delta} + \vec{R}_m^{Iter,\delta}
\end{aligned} \tag{3.2}$$

The main steps of social and individual intelligence for horses are discussed below.

3.1. Grazing (G)

Horses are grazing animals, which feed on plants, grasses, forages, etc. They graze on a pasture for 16 to 20 h a day, and their rest time is short. This method of slow grazing is called continuous eating. Perhaps you have seen the mares with their foal, while they graze in pasture [69].

The HOA algorithm models the graze area around each horse with coefficient g as such each horse is grazing on certain areas according to Fig. 1. Horses graze at any age throughout their lifetime. The mathematical implementation of grazing is in

Table 1
Literature of meta-heuristic optimization algorithms.

	Algorithm full name	Arbitrary Name	Author (s)	Year
Evolutionary	Genetic Algorithm [13]	GA	Holland JH	1975
	Simulated Annealing [14]	SA	Kirkpatrick S, Gelatt CD, Vecchi MP	1983
	Tabu Search [15]	TS	Glover F	1989
	Genetic programming [16]	GP	Koza JR	1992
	Evolution Strategy [17]	ES	Rechenberg I	1994
	Memetic Algorithm [18]	MA	Radcliffe NJ, Surry PD	1994
	Cultural Algorithm [19]	CA	Reynolds RG	1994
	Differential Evolution [20]	DE	Storn R, Price K	1997
	Evolutionary Programming [21]	EP	Yao X, Liu Y, Lin G	1999
	CoEvolutionary Algorithm [22]	CoEa	Kim YK, Kim JY, Kim Y	2000
	Gradient Evolution Algorithm [23]	GEA	Sinha A, Goldberg DE	2003
	Imperialistic Competitive Algorithm [24]	ICA	Atashpaz-Gargari E, Lucas C	2007
	Biogeography-Based Optimization [25]	BBO	Simon D	2008
	States of Matter Search [26]	SMS	Cuevas E, Echavaría A, Ramírrez AM	2014
	Sine Cosine Algorithm [27]	SCA	Seyedali Mirjalili	2016
	Multi-level Cross Entropy Optimizer [28]	MCEO	Miarnaeimi F, Azizyan G, Rashki M	2018
Physic-based	Small-World Optimization Algorithm [29]	SWOA	Du H, Wu X, Zhuang J	2006
	Central Force Optimization [30]	CFO	Formato RA	2007
	Magnetic Optimization Algorithm [31]	MOA	Tayarani N, Akbarzadeh M	2008
	Gravitational Search Algorithm [32]	GSA	Rashedi E, Nezamabadi-Pour H	2009
	Charged System Search [33]	CSS	Kaveh A, Talatahari S	2010
	Chemical-Reaction Optimization [34]	CRO	Albert Y, Lam S	2010
	Black Hole [35]	BH	Hatamlou A	2012
	Curved Space Optimization [36]	CSO	Moghaddam FF, Moghaddam RF, Cheriet M	2012
	Water Evaporation Optimization [37]	WEO	Kaveh A, Mahdavi VR	2016
	Ideal Gas Molecular Movement [38]	IGMM	Varae M, Ghasemi MR	2016
	Multi-Verse Optimizer [39]	MVO	Mirjalili S, Mirjalili SM	2015
	Vibrating Particles System [40]	VPS	Kaveh A, Ghazaan MI	2017
Swarm Intelligence (SI)	Particle Swarm Optimization [10]	PSO	Eberhart, R.C, J. Kennedy	1995
	Artificial Fish Swarm Algorithm [41]	AFSA	Li X	2003
	Honey Bee Optimization [42]	HBO	Dervis Karaboga D	2005
	Termite Colony Optimization [43]	TCO	Roth M	2005
	Ant Colony Optimization [8]	ACO	Dorigo M	2006
	Shuffled Frog-Leaping [44]	SFL	Eusuff M, Lansey K, Pasha K	2006
	Monkey Search [45]	MS	Mucherino A, Seref O	2007
	Dolphin Partner Optimization [46]	DPO	Shiqin Y, Jianjun J, Guangxing Y	2009
	Firefly Algorithm [47]	FA	Yang X-S	2010
	Bat Algorithm [48]	BA	Yang XS	2010
	Bird Mating Optimizer [49]	BMO	Askarzadeh A, Rezazadeh A	2012
	Fruit Fly Optimization [50]	FFO	Pan W-T	2012
	Lion Pride Optimizer [51]	LPO	Wang B, Jin X, Cheng B	2012
	Krill Herd [52]	KH	Gandomi AH, Alavi AH	2012
	Grey Wolf Optimizer [53]	GWO	Mirjalili S, Mirjalili SM, Lewis A	2014
	Cuckoo Search [54]	CS	Gandomi AH, Yang XS, Alavi AH	2013
	Soccer League Competition Algorithm [55]	SLCA	Moosavian N, Roodsari BK	2014
	Ant Lion Optimizer [56]	ALO	Mirjalili S	2015
	Dragonfly Algorithm [57]	DA	Mirjalili S	2015
	moth–flame Optimization [58]	MFO	Mirjalili S	2015
	Whale Optimization Algorithm [59]	WOA	Mirjalili S, Lewis A	2016
	Salp Swarm Algorithm [60]	SSA	Mirjalili S, Gandomi AH, Zahra S, Saremi S	2017
	Grasshopper Optimization Algorithm [61]	GOA	Saremi S, Mirjalili S, Lewis A	2017
	Harris Hawks Optimization [62]	HHO	Heidari A, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H	2019
	Flying Squirrel Optimizer [63]	FSO	Azizyan G, Miarnaeimi F, Rashki M, Shabakhty N	2019

accordance with Eqs. (3.3) and (3.4).

$$\vec{G}_m^{Iter,AGE} = g_{Iter}\left(\check{u} + \mathcal{P}\check{l}\right)\left[X_m^{(Iter-1)}\right], \quad AGE = \alpha, \beta, \gamma, \delta \tag{3.3}$$

$$g_m^{Iter,AGE} = g_m^{(Iter-1),AGE} \times \omega_g \tag{3.4}$$

where, $\vec{G}_m^{Iter,AGE}$ is the motion parameter of ith horse, and it shows the concerned horse's tendency to graze. This factor reduces linearity with ω_g per iteration. $\check{l}$ and $\check{u}$ are lower and upper bounds of grazing space, respectively, and $\mathcal{P}$ is a random number between 0 and 1. It is recommended to consider $\check{l}$ and $\check{u}$ equal to 0.95 and 1.05, respectively, and the coefficient g be equal to 1.5 for all age ranges.

3.2. Hierarchy (H)

Horses are not free on their own [70]. They pass their lives following a leader, which is often undertaken by human beings. An adult stallion or a mare is also responsible for leadership in the herds of wild horses, which occurs in the law of hierarchy [70].

In this case, the coefficient h in HOA is considered to be the tendency of a herd of horses to follow the most experienced and strongest horse (Fig. 2). Studies have shown that the horses follow the law of hierarchy at the middle ages β and γ (aged between 5–15 years) [66,67]. It can be defined as Eqs. (3.5) and (3.6).

$$\vec{H}_m^{Iter,AGE} = h_m^{Iter,AGE}\left[X_*^{(Iter-1)} - X_m^{(Iter-1)}\right], \quad AGE = \alpha, \beta \text{ and } \gamma \tag{3.5}$$

$$h_m^{Iter,AGE} = h_m^{(Iter-1),AGE} \times \omega_h \tag{3.6}$$

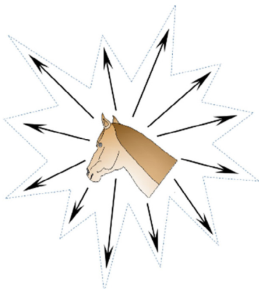


Fig. 1. Simulation of horse grazing in environment.

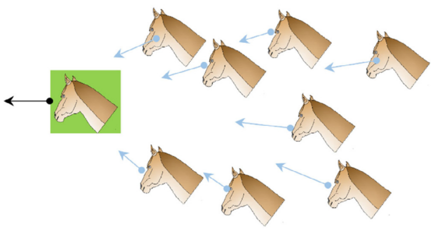


Fig. 2. Hierarchy of horses and simulation.

where, $\vec{H}_m^{Iter,AGE}$ indicates the effects of the best horse location on the velocity parameter, and $X_*^{(Iter-1)}$ shows the location of the best horse.

3.3. Sociability (S)

Horses require a social life and sometimes live with other animals. Herd life has guaranteed the horses' security since they are being hunted by predators. Pluralism increases the chances of survival and makes it easy to escape. You may often see that the horses are fighting each other due to their social features,

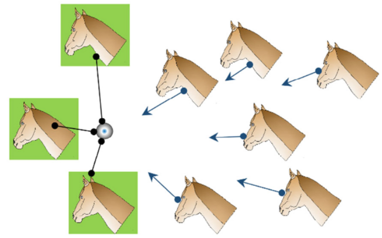


Fig. 4. Horses' imitation and simulation.

and horse's singularity is a cause of their irritability. Some of the horses seem to be happy around other animals such as cattle and sheep; however, they rarely like loneliness [69].

This behavior is considered as a movement toward the average position of other horses and is shown by factor s, as Fig. 3. It is clearly observed that the horses at the ages of 5–15 years are interested in the herd, that expressed in the following equation:

$$\vec{S}_m^{Iter,AGE} = s_m^{Iter,AGE}\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right], \quad AGE = \beta, \gamma \tag{3.7}$$

$$s_m^{Iter,AGE} = s_m^{(Iter-1),AGE} \times \omega_s \tag{3.8}$$

in Eq. (3.7), $\vec{S}_m^{\mathrm{Iter,AGE}}$ shows the social motion vector of ith horse and $s_m^{Iter,AGE}$ indicates the concerned horse's orientation toward the herd in $Iter^{\mathrm{th}}$ iteration. $s_m^{Iter,AGE}$ decreases in each cycle with a ω_s factor. N also shows the number of total horses, and AGE is the age range of each horse. The s coefficient for β and γ horses are calculated in the sensitivity analysis of the parameters.

3.4. Imitation

Horses imitate each other, and they learn each other's good and bad habits such as finding the appropriate location of a pasture [70].

The imitation behavior of horses is also considered as the factor i in the present algorithm. Young horses try to mimic others, and this feature does not disappear throughout their full

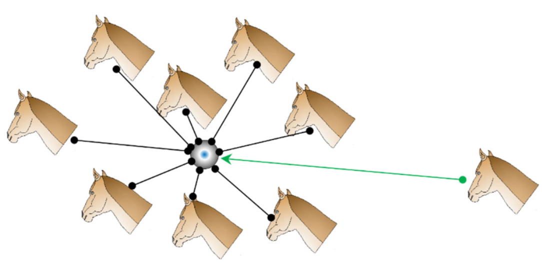


Fig. 3. General movement of horses, their sociability, and simulation.

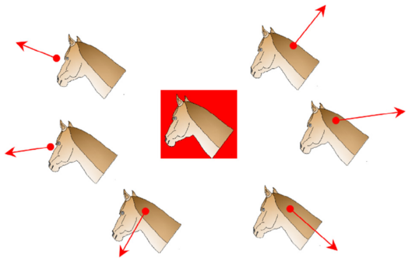


Fig. 5. Horses' defense mechanism and simulation.

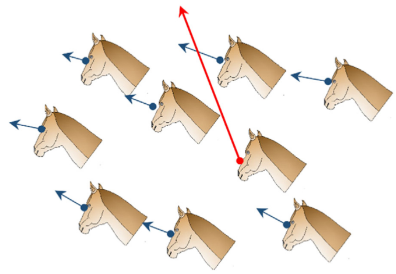


Fig. 6. Horses' roam and simulation.

maturity (Fig. 4), as it described in Eqs. (3.9) and (3.10).

$$\vec{I}_m^{Iter,AGE} = i_m^{Iter,AGE}\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right], \quad AGE = \gamma \tag{3.9}$$

$$i_m^{Iter,AGE} = i_m^{(Iter-1),AGE} \times \omega_i \tag{3.10}$$

In Eqs. (3.9) and (3.10), $\vec{I}_m^{Iter,AGE}$ is the motion vector of ith horse toward the average of best horses with $\hat{X}$ locations. pN shows the number of horses with the best locations. It is proposed that p is set as 10% of the horses. Moreover, ω_i is a reduction factor per cycle for i_{Iter} as shown previously.

3.5. Defense mechanism (D)

Horses' reaction is a reflection of the fact that they have been victims of predators [66]. They defend themselves by showing the fight-or-flight response. Their first reaction is to escape. Furthermore, they buck in case of trapping. Horses fight for food and water to take away the rivals and avoid hazardous environments, where there are enemies such as wolves by instinct [66,69].

Horses' defense system in the HOA algorithm functions by running away from horses showing inappropriate responses that are far from optimal, in accordance with Fig. 5. Their defense system is characterized by the factor d. Horses have to flee or fight against their enemies, as mentioned above. Such a defense mechanism exists throughout the entire lifetime of a young or mature horse, whenever possible. The defense mechanism of horses is presented by a negative coefficient in Eqs. (3.11) and (3.12), to keep the horse away from inappropriate positions.

$$\vec{D}_m^{Iter,AGE} = -d_m^{Iter,AGE}\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right], \quad AGE = \alpha, \beta \text{ and } \gamma \tag{3.11}$$

$$d_m^{Iter,AGE} = d_m^{(Iter-1),AGE} \times \omega_d \tag{3.12}$$

where $\vec{D}_m^{Iter,AGE}$ shows the escape vector of ith horse from the average of some horses with worst locations, which are shown by $\check{X}$ vector. qN is also shows the number of horses with the worst locations. It is suggested that q is equal to 20 percent of the total horses. ω_d shows the reduction factor per cycle for d_{Iter}, as previously noted.

3.6. Roam (R)

Horses roam and graze in the nature from pasture to pasture in the search of food [66]. Most horses are kept in stables, though they retain the mentioned attribute. A horse may suddenly go to another location for a graze. Horses are extremely curious, and they often visit everywhere to discover new pastures and to know their neighborhoods. Side walls are designed in a way that the horses can see each other and their curiosity is met in an appropriate stable [66].

This behavior is simulated as a random movement and shown by a factor r. Roaming in horses is almost observed at young ages and disappears gradually as they reach maturity. This process is also shown in Fig. 6, Eqs. (3.13) and (3.14).

$$\vec{R}_m^{Iter,AGE} = r_m^{Iter,AGE}\,\wp X^{(Iter-1)}, \quad AGE = \gamma, \delta \tag{3.13}$$

$$r_m^{Iter,AGE} = r_m^{(Iter-1),AGE} \times \omega_r \tag{3.14}$$

where, $\vec{R}_m^{Iter,AGE}$ represents the random velocity vector of ith horse for a local search and an escape from local minima, and ω_r shows the reduction factor of $r_m^{Iter,AGE}$ per cycle.

General velocity vector is obtained by substituting Eqs. (3.3) to (3.14) in Eq. (3.2).

Velocity of δ horses at the age of 0–5 years:

$$\begin{aligned}\vec{V}_m^{Iter,\delta} = &\left[g_m^{(Iter-1),\delta}\omega_g\left(\check{u}+\wp\check{l}\right)\left[X_m^{(Iter-1)}\right]\right] \\ &+\left[i_m^{Iter-1,\delta}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{Iter-1}\right)-X^{Iter-1}\right]\right] \\ &+\left[r_m^{Iter-1,\delta}\omega_r\wp X^{Iter-1}\right]\end{aligned} \tag{3.15}$$

Velocity of γ horses at the age of 5–10 years:

$$\begin{aligned}\vec{V}_m^{Iter,\gamma} = &\left[g_m^{Iter-1,\gamma}\omega_g\left(\check{u}+\wp\check{l}\right)\left[X_m^{Iter-1}\right]\right] \\ &+\left[h_m^{Iter-1,\gamma}\omega_h\left[X_*^{Iter-1}-X_m^{Iter-1}\right]\right] \\ &+\left[s_m^{Iter-1,\gamma}\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{Iter-1}\right)-X_m^{Iter-1}\right]\right] \\ &+\left[i_m^{Iter-1,\gamma}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{Iter-1}\right)-X^{Iter-1}\right]\right]\end{aligned}$$

A. *Basic structure: Introducing bound variables and other parameters;*

B. *Initialization: Providing the uniform random distribution of horses in a feasible space;*

C. *Check fitness: Checking the cost of each horse based on its location and objective function;*

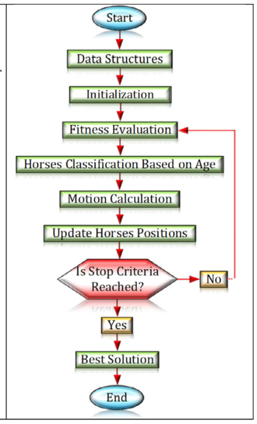
D. *Age determination: Calculating the age of α, β, γ and δ horses;*

E. *Apply velocity: Applying velocity to each horse by age;*

F. *Determining new positions: Updating the horses' positions in the search space;*

G. *Iteration: Moving to step C until the stop criterion is reached;*

(a)



(b)

Fig. 7. (a) Pseudo code of HOA and (b) flowchart of HOA.

$$- \left[d_m^{Iter-1,\gamma}\omega_d \left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{Iter-1} \right) - X^{Iter-1} \right] \right] + \left[r_m^{Iter-1,AGE}\omega_r \mathcal{P} X^{Iter-1} \right] \tag{3.16}$$

Velocity of β horses at the age of 10–15 years:

$$\vec{V}_m^{Iter,\beta} = \left[g_m^{Iter-1,\beta}\omega_g \left(\check{u} + \mathcal{P}\check{l} \right) \left[X_m^{Iter-1} \right] \right] + \left[h_m^{Iter-1,\beta}\omega_h \left[X_*^{Iter-1} - X_m^{Iter-1} \right] \right] + \left[s_m^{Iter-1,\beta}\omega_s \left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{Iter-1} \right) - X_m^{Iter-1} \right] \right] - \left[d_m^{Iter-1,\beta}\omega_d \left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{Iter-1} \right) - X^{Iter-1} \right] \right] \tag{3.17}$$

Velocity of α horses older than 15 years:

$$\vec{V}_m^{Iter,\alpha} = \left[g_m^{Iter-1,\alpha}\omega_g \left(\check{u} + \mathcal{P}\check{l} \right) \left[X_m^{Iter-1} \right] \right] - \left[d_m^{Iter-1,\alpha}\omega_d \left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{Iter-1} \right) - X^{Iter-1} \right] \right] \tag{3.18}$$

The pseudo code of the HOA algorithm is presented in Fig. 7(a) and (b).

To see how HOA can be beneficial, some remarks are as follows:

- The α horses obtain the best responses and also provide a guideline for the others. They act as a coach as they start searching to find the best response and make an exploitation strategy. This happens when the features of grazing and defense mechanism have to be applied.
- The β horses search the most likely optimal positions carefully, with a special attention to α.
- All the natural behaviors of horses are used to create γ horses. They have both robust and random motions and are useful for the both exploration and exploitation phases.
- Young horses are more wanton and playful; hence, they are more appropriate for the exploration phase.

4. HOA computational complexity

Computational complexity examines the problem-solving time of a particular method. In other words, the computational complexity of an algorithm is used as a parameter in estimating the computational cost of the method. The computational cost of meta-heuristic algorithms is also estimated based on the number of search agents, the number of problem dimensions, and the maximum number of iterations [71]. Employing the six factors in the movement of the horses, causes the HOA algorithm achieves a good balance between exploration and exploitation phases, and it reduces the computational complexity in problem solving.

In this regard, HOA uses a suitable solution for increasing the speed of problem solving as well as avoiding local optimum trapping by using sorting mechanism in a global matrix. The global matrix is obtained by juxtaposing positions (X) and the cost of each position ($C(X)$), as it is explained in Eqs. (4.1) and (4.2).

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,d} \end{bmatrix}, C(X) = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \tag{4.1}$$

$$Global\ Matrix = \begin{bmatrix} X & C(X) \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} & c_1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,d} & c_m \end{bmatrix} \tag{4.2}$$

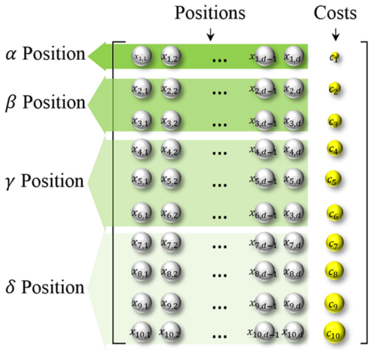


Fig. 8. Sorting process in the HOA algorithm.

In which, X and $C(X)$ denote the positions and the cost of each position, respectively. d and m are also the number of problem dimensions and horses' number, respectively. in the next step, the global matrix is sorted based on the last column (costs). The age of the horses is applied at this stage and according to Fig. 8. It is worst to mention that this figure shows the global matrix for ten horses.

The most important achievement of this strategy is exploiting with high accuracy and low speed in situations where the probability of being an optimal point is high (around the α horses), and exploring with low accuracy and high speed on situations where the probability of being an optimal point is low (around the δ and γ horses).

HOA considers the fast matrix sorting system, as it mentioned. So, the computational cost (denoted by CC) in the sorting step at best and worst state is equal to $CC(m \times \log(m))$ and $CC(m^2)$ respectively. Therefore, the overall computational cost of the HOA is also obtained by the Eq. (4.3) [58].

$$\begin{aligned} CC\,(HOA) &= CC\left(\frac{Iter_{max}}{2} \times [CC\,(Sorting) + CC\,(position\ update)]\right) \\ &= CC\left(\frac{Iter_{max}}{2} \times \left[m^2 + m \times d\right]\right) \\ &= CC\left(\frac{Iter_{max}m^2}{2} + \frac{Iter_{max}d}{2}\right) \end{aligned} \tag{4.3}$$

Some of the important benefits of HOA algorithm in reducing computational cost are listed below:

In the initial step of the algorithm and during a global search by the horses, a good reconization of the problem space is provided;

- The way the α horses search the space is also implemented so that the exploitation phase is done very efficient;
- Putting the β horses following the α horses (around the ten percent global best positions) makes it possible to find the local optimum aroundα horses. Also, the other areas constantly being searched for by the γ and δ horses.
- Six movement pattern of the horses results in the best search for problem space and a good balance between the exploration and exploitation phases.

That way, the proposed algorithm has the potential to achieve optimal response in the shortest possible time and with the lowest computational cost. In the next section, some benchmark mathematical and structural optimization problems are investigated in order to determine the efficiency of the proposed algorithm compared to the existing methods.

5. Parameter analysis

Several parameters affect the performance of HOA, as it described in the Horse Optimization Algorithm section. Now the sensitivity analysis of the response to changes in parameters is performed, in order to gain a correct understanding of the effectiveness of these parameters. Changing the g (Grazing) factor has no significant effect on the resulting responses and only increases or decreases the search range of each horse. The following factors are examined in sensitivity analysis:

- Hierarchy factor for β and γ horses (h_β and h_γ);
- Sociability factor for β and γ horses (s_β and s_γ);
- Imitation factor for γ horses (i_γ);
- Defense factor for α, β and γ horses (d_α, d_β and d_γ);
- Roam factor for δ and γ horses (r_δ and r_γ);

The coefficients were changed from 0.1 to 1 with a step of 0.1 and the algorithm was run on the spherical function in 500 dimensions, in each case. Fig. 9 shows the optimum answer obtained by each analysis and the best values for each coefficient are marked on the figure.

The best values for the coefficients are including:

- h_β and h_γ are equal to 0.9 and 0.5, respectively;
- s_β and s_γ are equal to 0.2 and 0.1, respectively;
- i_γ is equal to 0.3;
- d_α, d_β and d_γ are equal to 0.5, 0.2 and 0.1, respectively;
- r_δ and r_γ are equal to 0.1 and 0.05, respectively.

Although, we decide to develop a hyper heuristic version of HOA, that combines this method with other optimization algorithms to calculate the optimum amount of these factors for each problem.

6. Results and discussion

The results of HOA are presented in this section. As this algorithm is compared with the Grasshopper Optimization Algorithm (GOA) [61], Sine Cosine Algorithm (SCA) [27], Multi-Verse Optimizer (MVO) [39], moth–flame Optimization (MFO) [58], Dragonfly Algorithm (DA) [57], and Gray Wolf Optimizer (GWO) [53] algorithms.

It should be mentioned that the number of horses and iterations in all the algorithms is set to be 50 and 1000, respectively, to conduct a fair comparison. Statistical tests are conducted to show the significance of the results as well.

6.1. Uni-modal benchmark functions

Uni-modal test functions are ideal means to feature the algorithm's power throughout the exploration phase. in this section, there are four uni-modal benchmark functions, including Sphere

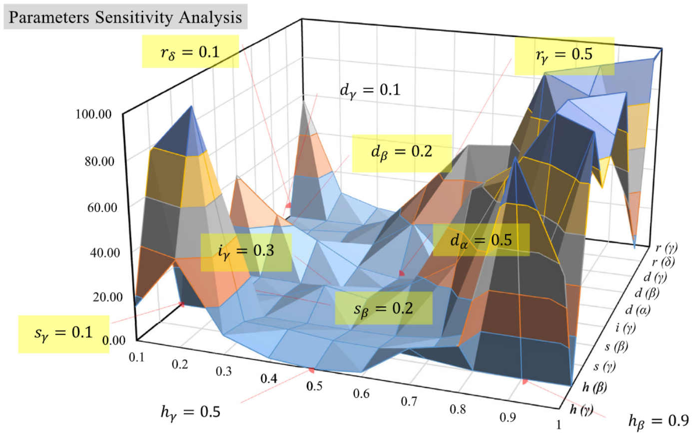
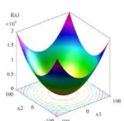


Fig. 9. HOA parameter sensitivity analysis.

(F1), Rotated Hyper-Ellipsoid (F2), Schwefel.1 (F3), and Schwefel.2 (F4). Equations, function shape, table of results, diagrams of convergence, and the best costs are provided for each function.

F1 (Sphere function) [72]



$$f(x) = \sum_{i=1}^{Dim} x_i^2 \qquad Range = [-100, 100] \qquad f_{min} = 0 \qquad \boldsymbol{x}^* = [0,0,\ldots,0]^{Dim} \tag{6.1}$$

The best cost and other relevant best positions of the Sphere function in Eq. (6.1) are shown by f_{min} and $\boldsymbol{x}^*$, which are equal to zero. *Dim* indicates the number of problem dimensions (number of variables).

Table 2 shows the results, including the best cost, standard deviation of responses (SD), *p*-value of Wilcoxon rank-sum test, and CPU running time for all of the aforementioned algorithms to solve Sphere function at different dimensions. The values are obtained from 30 independent runs of algorithms for each number of dimensions. The best results in each test function are selected and compared with those of the other algorithms separately. N/A (Not Applicable) is written down for the best algorithm and each function because the best algorithm cannot be compared with itself.

The GOA, SCA, MVO, and DA algorithms failed to find a good-enough response. GWO, on the other hand, provides approximately good-enough responses in a short time, which are close to optimal response. The results of HOA are highly close to absolute optimum point found during a reasonable time. The best cost and SD obtained by HOA is far better than those obtained by all the other algorithms. The p-values of the non-parametric statistical test, i.e. Wilcoxon's rank sum test, are N/A for HOA, indicating the statistical superiority of this algorithm. Accordingly, the HOA algorithm has the potential to solve problems that cannot be solved efficiently by other algorithms, according to the no free lunch (NFL) theorem.

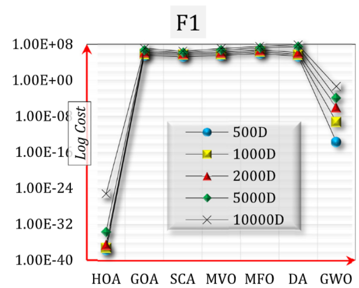


Fig. 10. Obtained best costs of F1.

Figs. 10 and 11 present the convergence curve at 2000 dimensions and the best costs at five different dimensions, respectively.

Evidently, Fig. 10 reflects the superiority of the HOA algorithm and its difference with other methods to find the best solutions in highly complex spaces.

Fig. 11 also shows the convergence curve of the abovementioned algorithms employed to solve Sphere function at 2000 dimensions with 1000 iterations. The number of search operators in all the algorithms is set as 50 to create a systematic comparison. Finding an optimal position in the primary iterations is the main advantage of HOA.

The mathematical implementation of F2 is shown in Eq. (6.2). This function is highly nonlinear and the $f(x)$ increases dramatically when the dimensions of the problem is enhanced. This can

Table 2
Optimization results of F1 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**1.84E−38**	349889	165259	313489	916271	225517	1.82E−14
	SD	2.85E−40	69212	72687	74992	50989	28738	7.66E−16
	P-Wlcxn	*N/A*	0.009773	0.005289	0.008345	0.00464	0.004098	0.002878
	Run Time (s)	49	479	8	17	9	290	8
1000	*Best Cost*	**5.35E−38**	705239	572255	672733	2432815	507688	5.36E−10
	SD	1.44E−40	95316	18146	17983	141492	63543	3.4E−11
	P-Wlcxn	*N/A*	0.00562	0.006541	0.005455	0.00405	0.000401	0.005264
	Run Time (s)	88	723	13	38	15	570	16
2000	*Best Cost*	**2.79E−37**	1780937	836374	1594276	5671196	1372209	9.70E−07
	SD	1.75E−38	230886	50946	180347	566349	77664	3.31E−08
	P-Wlcxn	*N/A*	0.008167	0.008768	0.004112	0.002392	0.008477	0.009365
	Run Time (s)	164	1862	24	82	29	755	49
5000	*Best Cost*	**1.94E−34**	4501328	1322409	4561398	1.5E+07	2.9E+07	1.15E−04
	SD	1.90E−35	327535	65737	371614	941069	1211897	6.31E−05
	P-Wlcxn	*N/A*	0.006501	0.009581	0.08234	0.004696	0.00924	0.001483
	Run Time (s)	400	2980	57	171	72	1621	80
10000	*Best Cost*	**5.47E−26**	11752861	2804620	1.1E+07	3.1E+07	7.1E+07	0.040605
	SD	7.26E−24	780214	92774	391081	665587	897899	5.22E−04
	P-Wlcxn	*N/A*	0.002851	0.009472	0.006134	0.003556	0.18954	0.000943
	Run Time (s)	795	6348	116	364	173	3912	154

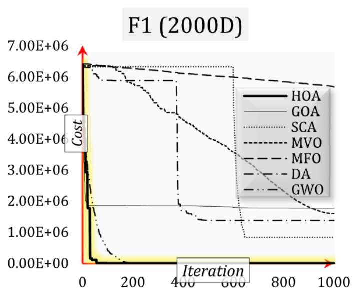


Fig. 11. Convergence curve of F1 in 2000D.

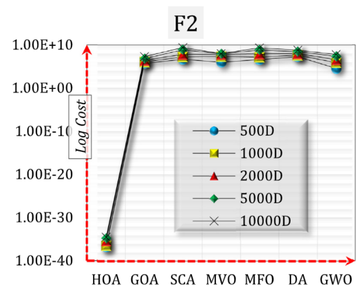
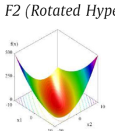


Fig. 12. Obtained best costs of F2.

be a challenging issue in high-dimensional spaces, despite its apparent simplicity.

F2 (Rotated Hyper-Ellipsoid function) [72]



$$f(x) = \sum_{i=1}^{Dim}\left(\sum_{j=1}^{i} x_j\right)^2$$
$$Range = [-100,100]$$
$$f_{min} = 0 \tag{6.2}$$
$$\boldsymbol{x}^* = [0,0,\ldots,0]^{Dim}$$

Table 3 provides the responses of seven algorithms to solve problem F2 in five different high dimensional spaces, as previously noted.

Although the required time to solve this problem by HOA is not much more, compared to the others, the HOA responses are much better than those obtained by the other algorithms in terms of best cost and SD. The p-values, however, confirmed the significance and robustness of fitness values, discovered by HOA on 30 independent tests.

The best responses of the six other algorithms for F2 improved by HOA significantly since the vertical axis of Fig. 12 is presented

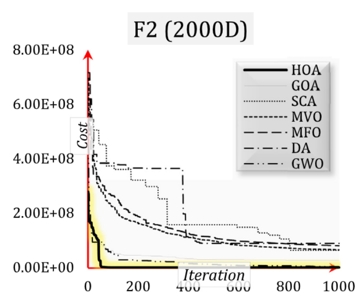


Fig. 13. Convergence curve of F2 in 2000D.

Table 3
Optimization results of F2 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
	Best Cost	**1.50E−37**	479301	4488844	891335	3719764	1.8E+07	36153.61
500	*SD*	5.65E−39	30207	385713	50794	456604	655737	1754
	P-Wlcxn	*N/A*	0.07912	0.000544	0.008519	0.004663	0.008453	0.008229
	Run Time (s)	123	272	81	91	82	261	84
	Best Cost	**2.95E−37**	1176233	15722451	2.1E+07	1.3E+07	4.1E+07	724582
1000	*SD*	1.13E−38	69212	517414	7918964	1427702	2909442	9094
	P-Wlcxn	*N/A*	0.004589	0.006722	0.000574	0.005072	0.009946	0.003558
	Run Time (s)	272	584	190	173	119	672	314
	Best Cost	**3.87E−36**	2569258	6.2E+07	6.4E+07	8.0E+07	8.8E+07	3584493
2000	*SD*	4.45E−37	196434	5300869	2511438	3.5E+07	7520569	26178
	P-Wlcxn	*N/A*	0.000378	0.009433	0.000336	0.006853	0.004958	0.009203
	Run Time (s)	678	1709	537	362	203	1283	1122
	Best Cost	**1.48E−35**	6501678	7.18E+08	8.5E+07	6.4E+08	2.7E+08	2.1E+07
5000	*SD*	1.46E−36	344335	4.1E+07	103539	6.1E+07	4.7E+07	720118
	P-Wlcxn	*N/A*	0.002353	0.002236	0.00269	0.006248	0.13436	0.001843
	Run Time (s)	3127	4099	1049	805	524	2671	2417
	Best Cost	**2.77E−35**	1.8E+07	2.25E+09	9.1E+07	2.0E+09	5.8E+08	5.9E+07
10000	*SD*	3.47E−36	4003498	8.6E+07	8242734	1.7E+07	3.1E+07	6434784
	P-Wlcxn	*N/A*	0.054894	0.000629	0.004464	0.006261	0.001373	0.002169
	Run Time (s)	7453	9820	2183	1751	2329	5613	6719

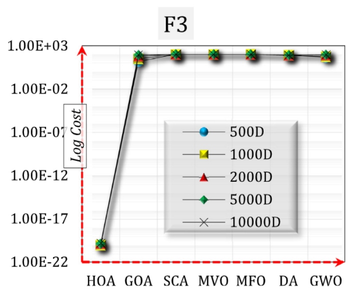


Fig. 14. Obtained best costs of F3.

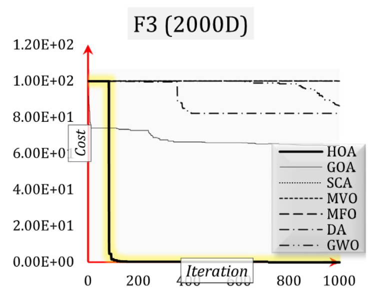
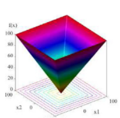


Fig. 15. Convergence curve of F3 in 2000D.

in a logarithmic scale. The results are highly different, and the highly efficient performance of HOA is shown by its convergence curve, compared to the other methods in Fig. 13.

F3 presents a simple Schwefel function, where $f(x)$ increases linearly after changing the variables.

F3 (Schwefel.1 function) [72]



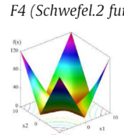
$$f(x) = \max_i\{|x_i|, 1 \le i \le Dim\}$$
$$Range = [-100,100]$$
$$f_{min} = 0 \tag{6.3}$$
$$\boldsymbol{x}^* = [0,0,\ldots,0]^{Dim}$$

The responses of HOA to the problem F3 is about 1E-20; however, the minimum value from the GWO is 46 at 500 dimensions, as reported in Table 4. In many cases, the responses are about 100, indicating the weakness of these algorithms to solve this simple problem in a high-dimensional space.

Fig. 14 shows the immense distance of the responses obtained by the presented algorithm and those explored by the other algorithms. It should be noted that the optimal values in this graph are also expressed in a logarithmic scale. Fig. 15 also presents the history of the best responses in 1000 iterations for each algorithm solving the problem F3. Evidently, HOA reached an acceptable level of optimal response in the first 100 iterations.

The fourth uni-modal benchmark function (F4) is also a Schwfel function with summation and multiplication operators.

F4 (Schwefel.2 function) [72]



$$f(x) = \sum_{i=1}^{Dim}|x_i| + \prod_{i=1}^{Dim}|x_i|$$
$$Range = [-2.5,2.5]$$
$$f_{min} = 0 \tag{6.4}$$
$$\boldsymbol{x}^* = [0,0,\ldots,0]^{Dim}$$

GWO performs well, in comparison to the other selected algorithms. Its response to F4 at 500 dimensions is 1.57E−09, indicating an excellent result. Interestingly, the optimum solutions presented by HOA are approximately 1E+10 times better than the

Table 4
Optimization results of F3 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
	Best Cost	**1.02E−20**	15.4288	98.6686	98.1294	97.7112	80.2307	46.6540
500	*SD*	2.23E−22	1.60E−02	6.03E−02	5.63E−02	9.24E−02	8.99E−02	3.49E−02
	P-Wlcxn	*N/A*	0.007162	0.00671	0.008183	0.000906	0.001954	0.001781
	Run Time (s)	46	118	8	19	9	153	9
	Best Cost	**8.87E−21**	31.7921	99.1451	98.3313	99.6257	81.3990	53.7401
1000	*SD*	4.78E−23	5.40E−02	6.97E−02	8.75E−02	9.24E−02	8.07E−02	4.22E−02
	P-Wlcxn	*N/A*	0.003351	0.008855	0.000216	0.001244	0.001517	0.004454
	Run Time (s)	94	263	16	44	15	391	17
	Best Cost	**1.33E−20**	64.4268	99.8642	99.8959	99.8074	81.9507	86.1761
2000	*SD*	1.92E−23	5.70E−02	9.10E−02	9.15E−02	9.54E−02	8.77E−02	7.35E−02
	P-Wlcxn	*N/A*	0.005925	0.003162	0.00981	0.000795	0.004976	0.007199
	Run Time (s)	201	584	25	80	28	970	31
	Best Cost	**1.22E−20**	83.6499	99.9327	99.9148	99.8959	82.0782	89.2239
5000	*SD*	4.18E−22	8.42E−02	8.21E−02	8.84E−02	9.54E−02	5.67E−02	4.34E−02
	P-Wlcxn	*N/A*	0.00607	0.009674	0.12168	0.001275	0.004989	0.008739
	Run Time (s)	492	1179	57	171	73	2471	76
	Best Cost	**1.49E−20**	89.1371	99.9781	99.9301	99.9611	83.4455	91.1764
10000	*SD*	1.01E−21	7.80E−02	4.52E−02	4.04E−02	1.01E−02	4.21E−02	8.96E−02
	P-Wlcxn	*N/A*	0.006187	0.004122	0.009327	0.000367	0.007425	0.003827
	Run Time (s)	893	2346	134	339	217	4803	153

Table 5
Optimization results of F4 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
	Best Cost	**7.00E−20**	381.5531	22.3979	563.4516	405.8499	3279.491	1.57E−09
500	*SD*	8.75E−21	33.8411	2.4137	30.6118	57.0459	832.6754	5.20E−11
	P-Wlcxn	*N/A*	0.002496	0.006988	0.071736	0.001595	0.002301	0.008747
	Run Time (s)	51	172	8	18	9	196	8
	Best Cost	**1.49E−18**	822.3770	54.4122	952.7371	991.242	8431.233	2.73E−07
1000	*SD*	5.38E−20	36.9239	6.2500	57.3069	92.1812	303.5648	8.60E−08
	P-Wlcxn	*N/A*	0.003787	0.009429	0.009838	0.003838	0.008645	0.00146
	Run Time (s)	94	361	11	37	15	423	15
	Best Cost	**8.33E−16**	2351.962	137.7924	1982.484	2197.125	18143.76	6.24E−05
2000	*SD*	6.22E−18	145.2102	6.8321	94.1573	108.1125	1366.146	2.14E−06
	P-Wlcxn	*N/A*	0.000481	0.005189	0.007824	0.006866	0.006869	0.007767
	Run Time (s)	186	796	24	89	28	810	32
	Best Cost	**1.69E−13**	5798.4634	152.4714	4572.316	5914.408	41970.49	1.23E−03
5000	*SD*	6.19E−15	767.145	24.8406	983.9951	749.4983	9766.231	5.75E−05
	P-Wlcxn	*N/A*	0.004636	0.001007	0.001733	0.004541	0.006331	0.006963
	Run Time (s)	404	1508	71	183	71	1761	76
	Best Cost	**4.25E−18**	14733.8	178.8952	11232.44	12050.2	85046.17	0.047284
10000	*SD*	3.93E−19	512.9546	6.4995	681.9785	3065.61	1880.302	9.53E−03
	P-Wlcxn	*N/A*	0.008053	0.006661	0.000129	0.005119	0.34329	0.008311
	Run Time (s)	809	3671	126	417	148	4032	162

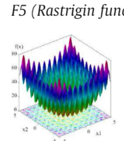
responses of GWO (7E-20 for 500 dimensions and 4.25E−18 for 10000 dimensions)! (see Table 5)

The best responses and convergence curve at 2000 dimensions with 1000 iterations are also presented in Fig. 16 and Fig. 17 for the concerned algorithms, respectively. The excellent performance of the proposed algorithm in terms of velocity and accuracy is well-presented. It should be noted that GWO algorithm has an acceptable performance; however, HOA contains the most competitive solutions.

6.2. Multi-modal benchmark functions

Multimodal benchmark functions are an effective criterion to challenge the exploitation ability of an algorithm. Three different functions, including Rastrigin (F5), Ackley (F6), and Drop-Wave (F7) function, are used in this section. As noted, these functions are optimized by HOA, GOA, SCA, MVO, MFO, DA, and GWO algorithms at 500, 1000, 2000, 5000, and 10000 dimensions. The best responses obtained from these algorithms with 50 search agents and 1000 iterations, SD of the responses for 30 independent runs, the p-values of Wilcoxon rank-sum test, and the CPU runtime are discussed below.

F5 (Rastrigin function) [72]



$$f(x) = \sum_{i=1}^{Dim}(x_i^2 - 10\cos(2\pi x_i) + 10)$$
$$Range = [-100,100]$$
$$f_{min} = 0$$
$$x^* = [0,0,\ldots,0]^{Dim} \quad (6.5)$$

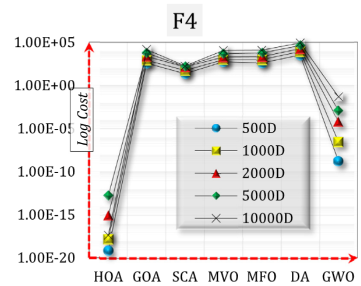


Fig. 16. Obtained best costs of F4.

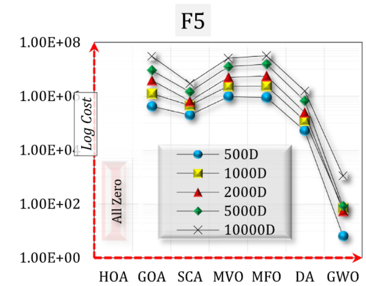


Fig. 18. Obtained best costs of F5.

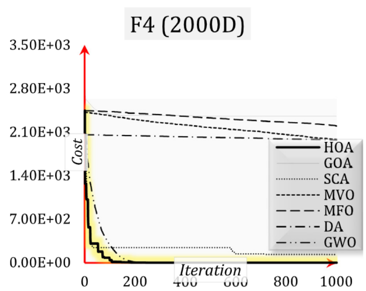


Fig. 17. Convergence curve of F4 in 2000D.

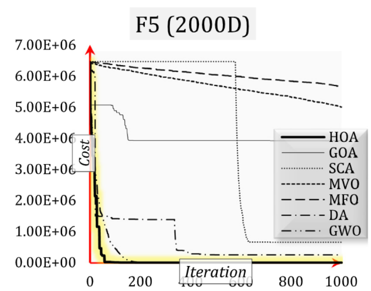


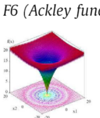
Fig. 19. Convergence curve of F5 in 2000D.

The Rastrigin function is a non-convex function and it typical example of non-linear multimodal function. To find out the minimum value of this function is a fairly difficult problem due to its large search space and its large number of local minima (see Table 6).

The performance of HOA is absolutely perfect in this case, and the best cost and SD of the results are reported to be equal to zero. On the other hand, the other algorithms, except GWO, provide no acceptable response.

As shown in Fig. 18, the global best cost obtained by HOA at all dimensions is calculated to be equal to zero in each analysis. It is worth mentioning that the result obtained by HOA is far better than those from GWO, even though the behaviors of HOA and GWO seems to be the same in Fig. 19.

F6 (Ackley function) [72]



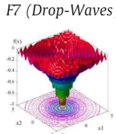
$$f(x) = -20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{Dim} x_i^2}{2}}\right) - \exp\left(\frac{\sum_{i=1}^{Dim}\cos(2\pi x_i)}{2}\right) + 20 + e$$
$$Range = [-100{,}100]$$
$$f_{min} = 0$$
$$\boldsymbol{x}^* = [0{,}0{,}\ldots{,}0]^{Dim} \tag{6.6}$$

Ackley function is widely used test the optimization algorithms. In its two-dimensional form, it is characterized by a nearly flat outer region and a large hole at the center. The function poses a risk to the optimization algorithms to be trapped in one of its many local minima. The complexity of this problem remarkably increases by increasing the dimensions of the problem (see Table 7).

All the best costs from the other algorithms are about 20; however, the HOA algorithm provides good-enough responses (about 1E-14) with highly low SD in a reasonable time. The phrase '*N/A*' for HOA under all conditions also indicates the sustainability and robustness of the results and confirms that these results are not achieved by chance.

Fig. 20 shows the significant difference between the responses obtained from HOA and those from the other methods. None of the algorithms could find a near-optimal solution, whereas HOA finds the optimal response in less than 200 initial iterations (Fig. 21).

F7 (Drop-Waves function) [72]



$$f(x) = -\frac{1+\cos\left(12\sqrt{\sum_{i=1}^{Dim} x_i^2}\right)}{2+0.5\sum_{i=1}^{Dim}(x_i^2)}$$
$$Range = [-100{,}100]$$
$$f_{min} = -1$$
$$\boldsymbol{x}^* = [0{,}0{,}\ldots{,}0]^{Dim} \tag{6.7}$$

The Drop-Wave function is multimodal and highly complex, even in its 2D version.

Table 8 shows the unmatched performance of HOA. Finding the exact solution for this complex problem in high-dimensional

Table 6
Optimization results of F5 obtained by different algorithms in different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
	Best Cost	**0.00**	421161.31	201021.5	955608.4	896710.2	51779.19	6.4812
500	*SD*	0.00	56276.891	5343.534	9460.197	1273.110	2663.457	0.7462
	P-Wlcxn	*NAN*	0.008015	0.009429	0.001288	0.001028	0.004292	0.000622
	Run Time (s)	50	141	9	16	10	289	9
	Best Cost	**0.00**	1250767	451792	2374589	2382712	123814	69.87
1000	*SD*	0.00	91677.34	1975.73	89697.46	258401.7	13540.80	19.1935
	P-Wlcxn	*NAN*	0.095634	0.000801	0.007476	0.006585	0.002746	0.005747
	Run Time (s)	88	252	15	41	17	621	17
	Best Cost	**0.00**	3924005	663114.9	5003697	5659792	249312	55.8
2000	*SD*	0.00	96658.33	9737.795	685917.4	380769.0	98729.1	7.9057
	P-Wlcxn	*NAN*	0.001234	0.006138	0.009009	0.002525	0.008093	0.004507
	Run Time (s)	166	569	28	76	31	2493	33
	Best Cost	**0.00**	9220328	1433908	1.2E+07	1.5E+07	671962.3	82.9
5000	*SD*	0.00	670802.3	60759.83	252127.3	980875.1	25518.16	9.5847
	P-Wlcxn	*NAN*	0.004169	0.004585	0.00123	0.003528	0.008644	0.005891
	Run Time (s)	415	1247	56	166	71	6170	85
	Best Cost	**0.00**	3.0E+07	2953671	2.5E+07	3.1E+07	1570329	1118.547
10000	*SD*	0.00	1960644.7	90509.8	869366.7	1041173	87336.44	20.7947
	P-Wlcxn	*NAN*	0.006547	0.002096	0.16840	0.003805	0.00631	0.005187
	Run Time (s)	828	2364	139	314	176	11273	174

Table 7
Optimization results of F6 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
	Best Cost	**4.00E−14**	20.4321	20.815	20.033	20.05759	20.053	21.4357
500	*SD*	7.79E−15	3.26E−02	7.59E−02	9.72E−03	5.75E−02	2.95E−02	2.26E−02
	P-Wlcxn	*NAN*	0.00594	0.003758	0.006807	0.00139	0.007973	0.003961
	Run Time (s)	46	91	10	15	10	264	10
	Best Cost	**1.87E−14**	20.586	20.8542	20.052	20.07	20.072	21.4791
1000	*SD*	2.99E−16	8.40E−02	6.91E−02	6.95E−02	6.26E−02	4.83E−02	3.83E−02
	P-Wlcxn	*NAN*	0.00212	0.007515	0.07487	0.008049	0.0852	0.008502
	Run Time (s)	74	196	17	31	16	583	22
	Best Cost	**1.87E−14**	20.76156	20.9	20.06283	20.06	20.091	21.5
2000	*SD*	8.93E−17	9.07E−02	6.60E−02	2.11E−02	6.13E−02	5.11E−03	7.03E−02
	P-Wlcxn	*NAN*	0.009429	0.004587	0.000971	0.007349	0.002716	0.007783
	Run Time (s)	192	408	28	64	30	1245	35
	Best Cost	**1.90E−14**	20.9122	20.9218	20.084	20.09	20.164	21.5223
5000	*SD*	6.36E−16	5.80E−02	7.88E−02	8.73E−03	9.22E−02	5.43E−02	7.54E−02
	P-Wlcxn	*NAN*	0.000994	0.000847	0.006548	0.009492	0.002676	0.005626
	Run Time (s)	439	823	69	139	75	2389	89
	Best Cost	**7.99E−14**	20.9873	20.9679	20.097	20.101	20.221	21.5667
10000	*SD*	5.89E−15	5.15E−02	5.34E−02	1.69E−02	7.15E−03	8.31E−02	3.30E−02
	P-Wlcxn	*NAN*	0.008895	0.003993	0.09285	0.000792	0.002927	0.005983
	Run Time (s)	785	2342	132	311	160	5423	182

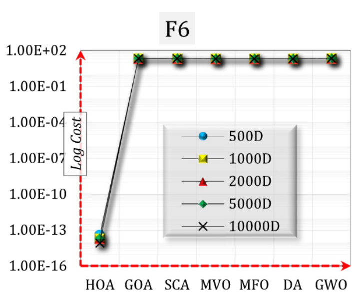


Fig. 20. Obtained best costs of F6.

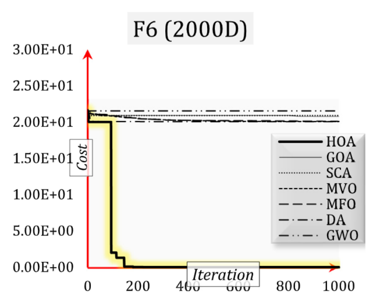


Fig. 21. Convergence curve of F6 in 2000D.

Table 8
Optimization results of F7 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	−**1.00**	−4.99E−05	−1.16E−05	−3.1E−04	−2.7E−06	−1.0E−03	−0.1893
	SD	0.00	1.19E−06	7.79E−06	8.56E−05	4.78E−07	2.75E−04	4.88E−02
	P-Wlcxn	*NAN*	0.001261	0.005092	0.007665	0.003257	0.009839	0.008352
	Run Time (s)	48	103	7	14	9	299	9
1000	*Best Cost*	−**1.00**	−7.34E−06	−7.22E−05	−5.6E−05	−1.2E−06	−1.0E−04	−0.1572
	SD	0.00	6.47E−07	2.99E−06	2.38E−06	7.71E−08	2.37E−06	8.10E−02
	P-Wlcxn	*NAN*	0.009319	0.00309	0.007763	0.002054	0.003583	0.004743
	Run Time (s)	96	242	21	33	15	487	15
2000	*Best Cost*	−**1.00**	−2.47E−07	−3.75E−06	−1.2E−07	−6.1E−07	−2.0E−06	−0.10768
	SD	0.00	9.30E−08	8.90E−07	4.53E−09	9.17E−08	5.35E−08	7.17E−03
	P-Wlcxn	*NAN*	0.007383	0.008912	0.007361	0.008819	0.009862	0.004367
	Run Time (s)	179	486	39	67	28	758	32
5000	*Best Cost*	−**1.00**	−8.36E−09	−8.72E−07	−4.8E−08	−2.4E−07	−2.0E−07	−0.05841
	SD	0.00	9.89E−11	6.36E−08	9.58E−09	4.58E−09	4.74E−08	2.99E−03
	P-Wlcxn	*NAN*	0.004609	0.001228	0.34806	0.005216	0.005451	0.002681
	Run Time (s)	434	961	75	156	71	1637	74
10000	*Best Cost*	−**1.00**	−3.72E−10	−5.22E−07	−1.3E−08	−1.2E−07	−1.7E−09	−2.2E−03
	SD	0.00	7.39E−12	8.59E−08	6.31E−09	3.71E−09	5.16E−11	7.25E−05
	P-Wlcxn	*NAN*	0.05167	0.003373	0.000746	0.00648	0.061525	0.006961
	Run Time (s)	854	451	159	332	181	3046	162

Table 9
Benchmark functions for further analysis [73].

Uni-modal				
Function	Dim	Range	f_{min}	$\mathbf{x}^*$
$F8 = \sum_{i=1}^{Dim-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	10,000	[−30,30]	0	$[0, 0, \ldots, 0]^{Dim}$
$F9 = \sum_{i=1}^{Dim} ix_i^4 + rand[0,1)$	10,000	[−1.28,1.28]	0	$[0, 0, \ldots, 0]^{Dim}$
Multi-modal				
Function	Dim	Range	f_{min}	$\mathbf{x}^*$
$F10 = \frac{1}{4000}\sum_{i=1}^{Dim} x_i^2 - \prod_{i=1}^{Dim} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	10,000	[−600,600]	0	$[0, 0, \ldots, 0]^{Dim}$
$F11 = \frac{\pi}{n}\left\{10sin\left(\pi y_1\right)+\sum_{i=1}^{Dim-i}(y_i-1)^2\left[1+10\sin 2\left(\pi y_{i+1}\right)\right]+(y_{Dim}-1)^2\right\} + \sum_{i=1}^{Dim} u\left(x_i, 10, 100, 4\right)$	10,000	[−50,50]	0	$[0, 0, \ldots, 0]^{Dim}$
$y_i = 1+\frac{x_i+1}{4}\ ,\ u\left(x_i,a,k,m\right)=\begin{cases} k(x_i-a)^{Dim} & x_i>a \\ 0 & -a<x_i<a \\ k(-x_i-a)^{Dim} & x_i<-a \end{cases}$				

Table 10
Optimization results of further analysis.

Function		HOA	GOA	SCA	MVO	MFO	DA	GWO
F8(10,000 dimensions)	*Best Cost*	**21.8660**	90809.8	9814.90	86393.6	5644.81	67133.06	45.3480
	SD	1.97409	5674.91	724.889	6955.65	37.9459	6382.94	2.2564
	P-Wlcxn	*NAN*	2.80E−04	8.47E−02	7.41E−03	3.09E−02	8.67E−03	3.50E−04
	Run Time (s)	6037	8624	2650	1863	3410	6749	7308
F9(10,000 dimensions)	*Best Cost*	**4.65e−05**	235.748	377.156	892.019	9055.31	84.5329	42.8476
	SD	7.54E−07	14.7324	27.8552	71.8175	60.8722	8.03730	2.1319
	P-Wlcxn	*NAN*	9.98E−03	4.96E−02	5.85E−02	6.97E−02	9.41E−03	8.51E−04
	Run Time (s)	943	2746	275	405	301	5712	341
F10(10,000 dimensions)	*Best Cost*	**0.00**	19.4430	71.4311	1.53824	73.4271	22.1618	1.88E−01
	SD	1.77E−03	1.46823	5.30308	0.12784	1.14914	1.01347	1.16E−02
	P-Wlcxn	*NAN*	7.72E−04	8.50E−04	2.60E−02	1.43E−02	1.02E−02	7.47E−04
	Run Time (s)	857	2942	234	605	280	5316	196
F11(10,000 dimensions)	*Best Cost*	**4.39e−04**	426066.6	1358.47	261698.0	8354.09	615224.5	78.773
	SD	1.82E−02	39717.21	113.446	23433.05	486.641	35852.01	6.73450
	P-Wlcxn	*NAN*	6.34E−02	3.34E−03	5.72E−03	6.68E−02	5.38E−03	2.59E−04
	Run Time (s)	2419	6578	352	914	408	3492	565

spaces with no standard deviation of the responses in a reasonable runtime presents the superiority of HOA, compared to the other algorithms. On the other hand, the other strong algorithms reached no near-optimal solution.

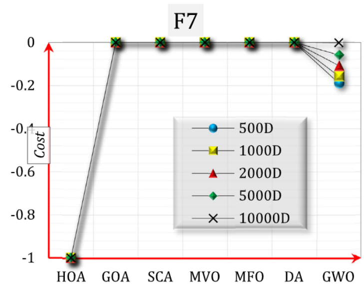


Fig. 22. Obtained best costs of F7.

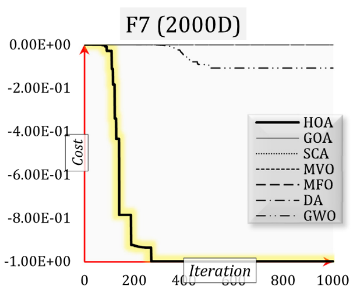


Fig. 23. Convergence curve of F7 in 2000D.

Fig. 22 demonstrates that the best responses of GOA, SCA, MVO, MFO, and DA are not even close to (-1). GWO has found answers other than zero, although they are not compelling. Fig. 23 shows the fast and excellent performance of HOA to solve this highly complex problem at 2000 dimensions.

6.3. Further analysis

Two more uni-modal and two multi-modal functions are examined with these methods in 10,000 dimensions to prove the effectiveness of the proposed HOA method to deal with high dimensional problems. The functions are presented in Table 9 and the results are provided in Table 10.

According to Table 10 the best cost and SD obtained by HOA was better than the other algorithms. The *P*-value of the Wilcoxon test in some cases was more than 0.05, shows that they were achieved by chance, although the run time of some algorithms, including SCA, MVO and MFO was shorter than HOA.

According to the results presented in Sections 6.1 and 6.2, HOA provides highly competitive results for the uni-modal and multi-modal benchmark functions.

The favorable performance of this algorithm can be explained regarding the following reasons:

- Regular classification of search agents based on their location and fitness;
- In each classification, with the improvement of the particle state, the number of search particles decreases, and thus the leadership process in the set is better;
- Adjustment with the horse herd's performance using their six natural behaviors; and
- Use of highly quick sorting in MATLAB, which consequently reduces the computational cost of the algorithm.

Accordingly, this algorithm outperforms GOA, SCA, MVO, MFO, DA, and GWO in analyzing eleven test functions in high-dimensional spaces. The establishment of harmonious relationships between the movements of horses leads to the highest and the best exploitation of horses' behavior, including that the HOA algorithm has some advantages in exploration and exploitation to find the best solution for the highly complex problems.

7. Conclusion

The present study proposed a fast and robust optimization algorithm, which was inspired by the general behaviors of horses at different ages, and employed to solve highly complex optimization problems. The HOA algorithm was benchmarked using seven well-known test functions at high dimensions, and it was found out that this algorithm was highly efficient in terms of exploration and exploitation. The results (namely the best cost, SD, *p*-value of Wilcoxon test, and CPU runtime) confirmed the ability of HOA to deal with these challenging problems having lots of unknown variables in high-dimensional spaces. These results also demonstrated a considerable development of the benchmark function at 500, 1000, 2000, 5000, and 10000 dimensions, in comparison to the existing solutions, implying the applicability of the suggested algorithm in dealing with highly complex and challenging problems. A highly favorable balance between the HOA components could justify the extraordinary ability of the proposed algorithm. This balance increases the efficiency of computational complexity by the algorithm. The adult α horses begin to have a local search with ultra-high precision around the global optimum. The β horses also search for the other close situations around the α horses, with a desire to move toward them; however, the γ horses are less interested in moving toward the α horses. They have a great desire to search new places and find the new possible global optimum locations. The young δ horses are accustomed to irregularities as such they are highly convenient options for the random search phase because of their certain behavioral characteristics. The behavioral coefficients determine six important features in horses, including grazing, hierarchy, sociability, imitation, defense mechanism, and roam. The inclusion of these parameters leads to the emergence of a coherent algorithm, which can find the optimum solution during the shortest possible time. The HOA development process is on progress to solve the multi-objective optimization problems in future studies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank to Dr. **Seyed Ali Mirjalili** for his inestimable helps and comments on this work.

Appendix. MATLAB Implementation of important parts of HOA

```
w=0.95;          % Reduction factors
wg=w;wh=w;ws=w;
wi=w;wd=w;wr=w;

pN=0.1;          % Percent of best horses
qN=0.2;          % Percent of worst horses

g_Alpha=1.50;    % Grazing of Alpha horses
d_Alpha=0.5;     % Defense Mechanism of Alpha horses

g_Beta=1.50;     % Grazing of Beta horses
h_Beta=0.90;     % Hierarchy of Beta horses
s_Beta=0.20;  % Sociability of Beta horses
d_Beta=0.20;     % Defense Mechanism of Beta horses

g_Gamma=1.50;    % Grazing of Gamma horses
h_Gamma=0.50;    % Hierarchy of Gamma horses
s_Gamma=0.10; % Sociability of Gamma horses
i_Gamma=0.30;    % Imitation of Gamma horses
d_Gamma=0.10;    % Defense Mechanism of Gamma horses
r_Gamma=0.05;    % Roam of Gamma horses

g_Delta=1.50;    % Grazing of Delta horses
r_Delta=0.10;    % Roam of Delta horses
```

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

```
% Calculating Alpha Horses Velocity
Horse.Velocity.Alpha=g_Alha*(0.95+0.1*rand)*(Horse.Best.Position-Horse.Position)
-d_Alpha*rand.*(BadPos-Horse.Position);
```

```
% Calculating Beta Horses Velocity
Horse.Velocity.Beta=g_Beta*(0.95+0.1*rand)*(Horse.Best.Position-
Horse.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_Gamma*(0.95+0.1*rand)*(Horse.Best.Position-
Horse.Position)...
+h_Gamma*rand.*(GlobalBest.Position-Horse.Position)...
+s_Gamma*rand.*(MeanPos-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.Position-
Horse.Position)...
+r_Delta.*(Horse.Velocity);
```

References

[1] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99.

[2] C. Darwin, On the Origin of Species, Vol. 1859, Routledge, 2004.

[3] K. Deb, An efficient constraint handling method for genetic algorithm, Comput. Methods Appl. Mech. Engrg. 186 (2–4) (2000) 311–338.

[4] F. Abbattista, N. Abbattista, L. Caponetti, An evolutionary and cooperative agents model for optimization, in: Proceedings of 1995 IEEE International Conference on Evolutionary Computation, IEEE, 1995, pp. 668–671.

[5] A.D. Belegundu, J.S. Arora, A study of mathematical programming methods for structural optimization. Part I: Theory, Internat. J. Numer. Methods Engrg. 21 (9) (1985) 1583–1599.

[6] E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, J. Mech. Des. 112 (2) (1990) 223–229.

[7] R. Hedayatzadeh, F.A. Salmassi, M. Keshtgari, R. Akbari, K. Ziarati, Termite colony optimization: A novel approach for optimizing continuous problems, in: 2010 18th Iranian Conference on Electrical Engineering, IEEE, 2010, pp. 553–558.

[8] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.

[9] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization, in: Problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, Springer-Verlag, IFSA, Citeseer, 2007.

[10] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995, pp. 39–43.

[11] B. Niu, H. Wang, Bacterial Colony Optimization, Vol. 2012, Discrete Dynamics in Nature and Society, 2012.

[12] E.-G. Talbi, Metaheuristics: From Design to Implementation, John Wiley & Sons, 2009.
[13] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT press, 1992.
[14] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[15] F. Glover, Tabu search—Part I, ORSA J. Comput. 1 (3) (1989) 190–206.
[16] J.R. Koza, Genetic Programming: On the Programming of Computers By Means of Natural Selection, MIT press, 1992.
[17] I. Rechenberg, J.M. Zurada, R.J. Marks II, C. Goldberg, Evolution strategy, in computational intelligence: Imitating life, in: Computational Intelligence Imitating Life, IEEE Press, Piscataway, 1994.
[18] N.J. Radcliffe, P.D. Surry, Formal Memetic Algorithms, in: AISB Workshop on Evolutionary Computing, Springer, 1994, pp. 1–16.
[19] R.G. Reynolds, An introduction to cultural algorithms, in: Proceedings of the Third Annual Conference on Evolutionary Programming, World Scientific, 1994, pp. 131–139.
[20] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[21] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (2) (1999) 82–102.
[22] Y.K. Kim, J.Y. Kim, Y. Kim, A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines, Appl. Intell. 13 (3) (2000) 247–258.
[23] A. Sinha, D.E. Goldberg, A Survey of Hybrid Genetic and Evolutionary Algorithms, Vol. 2003004, IlliGAL report, 2003.
[24] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 4661–4667.
[25] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[26] E. Cuevas, A. Echavarría, M.A. Ramírez-Ortegón, An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation, Appl. Intell. 40 (2) (2014) 256–272.
[27] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[28] F. MiarNaeimi, G. Azizyan, M. Rashki, Multi-level cross entropy optimizer (MCEO): An evolutionary optimization algorithm for engineering problems, Eng. Comput. 34 (4) (2018).
[29] H. Du, X. Wu, J. Zhuang, Small-world optimization algorithm for function optimization, in: International Conference on Natural Computation, Springer, 2006, pp. 264–273.
[30] R.A. Formato, Central force optimization: A new metaheuristic with applications in applied electromagnetics, in: Progress in Electromagnetics Research, PIER 77, 2007, pp. 425–491.
[31] M.-H. Tayarani-N, M.R. Akbarzadeh-T, Magnetic optimization algorithms a new synthesis, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 2659–2664.
[32] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: A gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[33] A. Kaveh, S. Talatahari, A novel heuristic optimization method: Charged system search, Acta Mech. 213 (3–4) (2010) 267–289.
[34] A.Y.S. Lam, V.O.K. Li, Chemical-reaction-inspired metaheuristic for optimization, IEEE Trans. Evol. Comput. 14 (3) (2010) 381–399.
[35] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inf. Sci. 222 (2013) 175–184.
[36] F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, Curved space optimization: A random search based on general relativity theory, 2012, arXiv preprint arXiv:1208.2214.
[37] A. Kaveh, T. Bakhshpoori, Water evaporation optimization: A novel physically inspired optimization algorithm, Comput. Struct. 167 (2016) 69–85.
[38] H. Varaee, M.R. Ghasemi, Engineering optimization based on ideal gas molecular movement algorithm, Eng. Comput. 33 (1) (2017) 71–93.
[39] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: A nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[40] A. Kaveh, M.I. Ghazaan, A new meta-heuristic algorithm: Vibrating particles system, Sci. Iran. Trans. A Civ. Eng. 24 (2) (2017) 551.
[41] X.L. Li, A New Intelligent Optimization-Artificial Fish Swarm Algorithm (Doctor thesis), Zhejiang University of Zhejiang, China, 2003.
[42] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical report-tr06, Erciyes university, engineering faculty, computer., 2005.
[43] M. Roth, Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks, 2005.
[44] M. Eusuff, K. Lansey, F. Pasha, Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization, Eng. Optim. 38 (2) (2006) 129–154.
[45] A. Mucherino, O. Seref, Monkey search: A novel metaheuristic search for global optimization, in: AIP Conference Proceedings, American Institute of Physics, 2007, pp. 162–173.
[46] Y. Shiqin, J. Jianjun, Y. Guangxing, A dolphin partner optimization, in: Intelligent Systems, 2009. GCIS'09. WRI Global Congress On, IEEE, 2009, pp. 124–128.
[47] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, 2010, arXiv preprint arXiv:1003.1409.
[48] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010, pp. 65–74.
[49] A. Askarzadeh, A. Rezazadeh, A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer, Int. J. Energy Res. 37 (10) (2013) 1196–1204.
[50] W.-T. Pan, A new fruit fly optimization algorithm: Taking the financial distress model as an example, Knowl.-Based Syst. 26 (2012) 69–74.
[51] B. Wang, X. Jin, B. Cheng, Lion pride optimizer: An optimization algorithm inspired by lion pride behavior, Sci. China Inf. Sci. 55 (10) (2012) 2369–2389.
[52] A.H. Gandomi, A.H. Alavi, Krill herd: A new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.
[53] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[54] A.H. Gandomi, X.S. Yang, A.H. Alavi, Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.
[55] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, Swarm Evol. Comput. 17 (2014) 14–24.
[56] S. Mirjalili, The Ant Lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[57] S. Mirjalili, Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073.
[58] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[59] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[60] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[61] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[62] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[63] G. Azizyan, F. Miarnaeimi, M. Rashki, N. Shabakhty, Flying squirrel optimizer (FSO): A novel SI-based optimization algorithm for engineering problems, Iran. J. Optim. 11 (2) (2019) 177–205.
[64] C. Li, X. Yu, T. Huang, G. Chen, X. He, A generalized hopfield network for nonsmooth constrained convex optimization: Lie derivative approach, IEEE Trans. Neural Netw. Learn. Syst. 27 (2) (2015) 308–321.
[65] C. Li, X. Yu, T. Huang, X. He, Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2017) 2407–2418.
[66] G.H. Waring, Horse behavior, in: The Behavioral Traits and Adaptations of Domestic and Wild Horses, Including Ponies, Noyes Publications, Mill Road, 1983.
[67] S.M. McDonnell, The Equid Ethogram: A Practical Field Guide to Horse Behavior, Eclipse Press, 2003.
[68] M.A. Levine, Domestication and early history of the horse, in: The Domestic Horse: The Origins, Development and Management of Its Behaviour, 2005, pp. 5–22.
[69] K. Krueger, J. Heinze, Horse sense: Social status of horses (Equus Caballus) affects their likelihood of copying other horses' behavior, Anim. Cogn. 11 (3) (2008) 431–439.
[70] F. Bogner, A comprehensive summary of the scientific literature on horse assisted education in Germany, 2011.
[71] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver press, 2010.
[72] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimization problems, 2013, arXiv preprint arXiv:1308.4008.
[73] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Jadavpur University, Nanyang Technological University, Kolkata, 2010, pp. 341–359.