

OPEN

Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization

Mahdi Azizi¹, Uwe Aickelin², Hadi A. Khorshidi²✉ & Milad Baghalzadeh Shishehgarkhaneh³

In this paper, Energy Valley Optimizer (EVO) is proposed as a novel metaheuristic algorithm inspired by advanced physics principles regarding stability and different modes of particle decay. Twenty unconstrained mathematical test functions are utilized in different dimensions to evaluate the proposed algorithm's performance. For statistical purposes, 100 independent optimization runs are conducted to determine the statistical measurements, including the mean, standard deviation, and the required number of objective function evaluations, by considering a predefined stopping criterion. Some well-known statistical analyses are also used for comparative purposes, including the Kolmogorov–Smirnov, Wilcoxon, and Kruskal–Wallis analysis. Besides, the latest Competitions on Evolutionary Computation (CEC), regarding real-world optimization, are also considered for comparing the results of the EVO to the most successful state-of-the-art algorithms. The results demonstrate that the proposed algorithm can provide competitive and outstanding results in dealing with complex benchmarks and real-world problems.

Abbreviations

GA	Genetic algorithm
CEC	Competitions on evolutionary computation
DE	Differential evolution
N	Number of neutrons
PSO	Particle swarm optimisation
Z	Number of protons
FA	Firefly algorithm
α	Dense and positively charged particles
ACO	Ant colony optimization
β	Negatively charged particles
HS	Harmony search
γ	Photons with higher levels of energy
BBC	Big-bang big-crunch
n	Total number of particles
MVO	Multiverse algorithm
d	Dimension of the considered problem
CGO	Chaos game optimization
x_i^j	The jth decision variable for determining the initial position of the ith candidate
PO	Projectiles optimisation
$x_{i,\min}^j$	Lower bounds of the jth variable in the ith candidate
ALO	Ant lion optimizer
$x_{i,\max}^j$	Upper bounds of the jth variable in the ith candidate
AISA	Adolescent identity search algorithm
EB	Enrichment bound
MGA	Material generation algorithm
NEL	Neutron enrichment level
AOS	Atomic orbital search
NEL^i	Neutron enrichment level of the ith particle

¹Department of Civil Engineering, University of Tabriz, Tabriz, Iran. ²School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia. ³Department of Civil Engineering, Islamic Azad University of Tabriz, Tabriz, Iran. ✉email: hadi.khorshidi@unimelb.edu.au

AHO	Archerfish hunting optimizer
SL ⁱ	Stability level of the ith particle
ICA	Imperialistic competitive algorithm
BS	Particles with best stability levels
WOA	Whale optimization algorithm
WS	Particles with worst stability levels
HHO	Harris hawks optimization
X _i ^{New1}	Newly generated particle in the universe
AOA	Arithmetical optimization algorithm
X _i	Current position vector of the ith particles (solution candidates) in the universe
EVO	Energy valley optimizer
x _j	The jth decision variable
CSA	Cuckoo search algorithm
D _i ^k	Total distance between the ith particle and the kth neighbouring particle
IMODE	Improved multi-operator differential evolution
EBOwithCMAR	Effective butterfly optimizer with covariance matrix adapted retreat phase
SL ⁱ	Stability level of the ith particle
HSES	Hybrid sampling evolution strategy
LSHADE-cnEpSin	LSHADE with an ensemble sinusoidal parameter adaptation
LSHADE-SPACMA	LSHADE with semi-parameter adaptation and covariance matrix adaptation
X _{CP}	Position vector for the centre of particles
j 2020	Developed differential evolution
GSK	Gaining sharing knowledge based algorithm
ES	Evolution strategy
SBS	Socio-behavioural simulation
COM	Classic optimization method
CGS	Combined genetic search
TLBO	Teaching–learning-based optimization
EGWO	Enhanced grey wolf optimizer
JA	Jaya algorithm
IPTR	Interior point trust region
SLP	Sequential linear programming
ECSS	Enhanced charged system search
ABC	Artificial bee colony
GWO	Grey wolf optimizer
r ₁	Random numbers in the range of [0, 1]
X _{Ng}	Position vector of the neighbouring particle around the ith particle
r ₂	Random numbers in the range of [0, 1]
r ₃	Random numbers in the range of [0, 1]
r ₄	Random numbers in the range of [0, 1]
OFE	Objective function evaluations
SD	Standard deviation
KS	Kolmogorov Smirnov
W	Wilcoxon
KW	Kruskal Wallis
D	Dimension
g	Number of inequality constraints
h	Number of equality constraints

In today's competitive world, many efforts are allocated to maximize the overall benefits of nature's limited resources. Recent advances in human knowledge and expertise reveal the need to consider greater accuracy, better performance, and higher construction speeds in the design of real-world systems. It is necessary to develop new methods for design and optimization and implement them on real complex problems to achieve each of the mentioned factors. Optimization is a decision-making process in which the search for a function's minimum and/or maximum values is considered. Optimization algorithms are divided into two categories as exact and approximate algorithms. Exact algorithms can find the optimal solution accurately. Still, in the case of complex optimization problems, they are not efficient enough, and their computational time increases exponentially according to the dimensions of the problem. So, given the limitations of exact methods and the need for precision and speed in identifying appropriate answers, approximate algorithms, like metaheuristics, can find suitable solutions close to the optimal solution in a shorter time that can be used to solve complex problems. In fact, the Greek prefix "meta", shown within the title, is utilized to demonstrate that these algorithms are "higher-level" heuristic algorithms differentiating with problem-specific heuristics¹. The metaphors used by scholars while developing new metaheuristics are as follows: natural evolution, insects, gravitation, electromagnetic force, ecosystem, water, plant, human, birds, and animals. Among them, "insects" is the most often used metaphor for simulating social behaviour to develop practical optimization problems, and bees are the most frequently used insect metaphor, followed by ants². Although metaheuristic algorithms give reasonable results, they do not provide optimum solutions.

Name	Authors	Year	Classifications	Inspiration
GA ³	J Holland	1992	Evolutionary-based	Charles Darwin's theory of natural evolution
Wisdom of Artificial Crowds (WoAC) ¹¹	R Yampolskiy et al	2012	Swarm-based	Human collective intelligence
Bat algorithm (BA) ⁴⁵	X Yang et al	2012	Nature-based	The echolocation behaviour of bats
Hyper-Spherical Search (HSS) algorithm ⁵	H. Karami et al	2014	Evolutionary-based	Space search mechanism
Passing vehicle search (PVS) ³¹	P Savsani et al	2016	Population-based	Vehicle passing on a two-lane highway
Dynamic Virtual Bats Algorithm (DVBA) ³²	A Topal et al	2016	Nature-based	Bat's echolocation behaviour
Crow search algorithm (CSA) ³³	A Askarzadeh	2016	Population-based	The intelligent behaviour of crows
Virus optimization algorithm (VOA) ³⁴	Y Liang et al	2016	Population-based	The behaviour of viruses attacking a living cell
Thermal Exchange Optimization algorithm ²¹	A Kaveh et al	2017	Physics-based	Newton's law of cooling
Queuing search (QS) ²⁵	J Zhang et al	2018	Human-based	Human activities in queuing
Henry gas solubility optimization (HGSO) ¹⁴	F Hashim et al	2019	Physics-based	The behaviour governed by Henry's law
Bear smell search algorithm (BSSA) ⁴⁶	A Ghasemi-Marzbali	2020	Nature-based	The dynamic behaviours of bear
Bald eagle search (BES) algorithm ³⁵	H. A. Alsattar et al	2020	Nature-based	The hunting strategy or intelligent social behaviour of bald eagles
Black Widow Optimization Algorithm (BWO) ⁴⁷	V Hayyolalam et al	2020	Nature-based	The unique mating behaviour of black widow spiders
Jellyfish Search (JS) optimizer ²⁴	J Chou et al	2021	Animal-based	The behaviour of jellyfish in the ocean
Cooperation search algorithm (CSA) ²⁶	Z Feng et al	2021	Population-based	The team cooperation behaviours in modern enterprise
Aquila Optimizer (AO) ²⁷	L Abualigah et al	2021	Population-based	The Aquila's natural habits when catching prey
Capuchin Search Algorithm (CapSA) ²⁸	M Braik et al	2021	Nature-based	The dynamic behaviour of capuchin monkeys
Artificial lizard search optimization (ALSO) ⁴⁸	N Kumar et al	2021	Animal-based	The dynamic foraging behaviour of Agama lizards
Archimedes optimisation algorithm (AOA) ¹⁹	F Hashim et al	2021	Physics-based	Law of physics Archimedes' Principle
Rat Swarm Optimizer (RSO) ¹⁰	G Dhiman et al	2021	Bio-based	The chasing and attacking behaviours of rats
Golden Tortoise Beetle Optimizer (GTBO) ²⁹	O Tarkhaneh et al	2021	Nature-based	The golden tortoise beetle's color-changing habit of attracting the opposite sex
Battle Royale Optimization (BRO) ³⁰	T Rahkar Farshi	2021	Population-based	A genre of digital games knowns as "battle royale
Lichtenberg Algorithm (LA) ²⁰	J Pereira et al	2021	Physics-based	The Lichtenberg figures patterns

Table 1. Historical to cutting-edge metaheuristic optimization algorithms.

Generally speaking, there remain four major classifications of metaheuristics based on the source of their inspiration. The first group is evolutionary algorithms (EAs), robust and flexible metaheuristic search algorithms based on Darwinian evolution that efficiently capture global solutions to complicated optimization problems. EAs replicate the biological course of evolution at the cellular level to create better candidate solutions using selection, crossover, mutation, and reproduction operators. The genetic algorithm (GA) introduced by John Holland and his colleagues in 1992 could be deemed as the so-called EA³. Another prominent algorithm is differential evolution (DE) which was introduced in 1997⁴. In 2014, another evolutionary-based metaheuristic algorithm called hyper-spherical search (HSS) was proposed⁵. The algorithm's core principle was searching the hypersphere's inner space, which is defined by the hypersphere's core and its particles. Swarm-based metaheuristic algorithms are the second category. Particle swarm optimization (PSO) is a well-known and often utilized swarm intelligence algorithm. The basic concept of PSO was inspired by a swarm of flying birds looking for food⁶. Furthermore, in 2012, another swarm-based algorithm named firefly algorithm (FA) was introduced⁷. As its name implies, this algorithm was inspired by the flashing behaviour of fireflies. Ant colony optimization (ACO) is another swarm-based algorithm that uses ants' foraging behaviour. ACO solves a problem by maintaining an updated pheromone trail and relocating these ants in the search space using simple mathematical calculations based on the area's transition probability and total pheromone^{8,9}. Among the other well-known algorithms in this category are rat swarm optimizer (RSO)¹⁰; the wisdom of artificial crowds (WoAC)¹¹; tuna swarm optimization (TSO)¹²; and artificial bee colony (ABC)¹³.

The Big-Bang Big-Crunch (BBCB) algorithm is inspired by the theories of the universe's evolution. Another prominent algorithm is henry gas solubility optimization (HGSO), imitating the behavior governed by Henry's law¹⁴. Moreover, Atomic Orbital Search (AOS) is another algorithm in which the fundamental principles of quantum physics are used as a source of inspiration¹⁵. Several more well-known algorithms in this categorization include the following: Material Generation Algorithm (MGA)^{16,17}, cyber-physical systems (CPS)¹⁸, Archimedes optimization algorithm (AOA)¹⁹, Lichtenberg Algorithm (LA)²⁰, and also Thermal Exchange Optimization algorithm (TEOA)²¹. Finally, the fourth group is human and animal lifestyles-based metaheuristic algorithms. Harris Hawks Optimizer (HHO) is one of the renowned animal behaviour-based algorithms; the cooperative behaviour and pursuit manner of Harris' hawks in nature, known as surprise pounce, is the fundamental inspiration for HHO²². In terms of efficiency, quality of results, and acceptable convergence in dealing with various applications in real-world problems, the HHO has gotten much attention from academics²³. An artificial Jellyfish Search (JS) optimizer has recently been proposed in 2021, inspired by the behaviour of jellyfish in the ocean²⁴. Additionally, queuing search (QS) is another algorithm based on human queueing behaviors²⁵. Table 1 contains more information on the algorithms mentioned above. Additionally, there are other metaheuristic optimization algorithms inspired by different concepts which have been proposed in the last years, including the following: cooperation search algorithm (CSA)²⁶, Aquila Optimizer (AO)²⁷, Capuchin Search Algorithm

(CapSA)²⁸, Golden Tortoise Beetle Optimizer (GTBO)²⁹, Battle Royale Optimization (BRO)³⁰, passing vehicle search (PVS)³¹, Dynamic Virtual Bats Algorithm (DVBA)³², crow search algorithm (CSA)³³, virus optimization algorithm (VOA)³⁴, bald eagle search (BES) algorithm³⁵, Gravitational Search Algorithm (GSA)³⁶, Grey Wolf Optimizer (GWO)³⁷, Teaching–Learning-Based Optimization (TLBO)³⁸, Fire Hawk Optimizer (FHO)^{39,40}, Social Spider Optimization (SSO)⁴¹, League Championship Algorithm (LCA)⁴², and Chaos Game Optimization^{43,44}.

In recent decades, one of the main challenges of artificial intelligence experts has been the applicability of the proposed metaheuristic algorithm in different fields to optimize and improve the overall efficiency of some specific problems. Artificial electric field algorithm for engineering design optimization⁴⁹, black widow algorithm for engineering optimization⁴⁷, earthquake engineering optimization of structures⁵⁰, engineering design optimization with queuing search algorithm²⁵, cuckoo search algorithm for optimization of the travelling salesman problem⁵¹, optimum design of engineering problems with sine cosine grey wolf optimizer⁵², engineering design optimization with self-adaptive Rao algorithm⁵³, design optimization of numerical and engineering optimization problems with improved Harris Hawks optimizer⁵⁴, unconstrained and constrained optimization by hybrid pathfinder optimizer⁵⁵, improved charged system search for optimization of fuzzy controllers⁵⁶, optimum design of structural systems with metaheuristics^{57,58} are some of the most recent research works in the area of applied artificial intelligence.

Even though there are several metaheuristic algorithms, more are always needed. The No Free Lunch (NFL) theory holds that no one method can be used to solve all optimization problems. Thus, the introduction of novel metaheuristic optimization algorithms is continuously ongoing. The creation of new metaheuristics is advantageous to science since they might improve the precision or effectiveness of the optimization procedure for a host of problems⁵⁹. This assertion drives our attempts to suggest a unique metaheuristic algorithm inspired by the cutting-edge physics concepts regarding stability and various forms of decay in particles. Meanwhile, accurate algorithms ensure the most optimal solution, but the problem is that these algorithms are difficult problems. They do not work quickly, and the time to find solutions to complex problems will increase exponentially, and for the hardest and most complex problems, the results of the exact algorithm are not satisfactory. The approximate algorithms should be utilized if the optimal response to the exact algorithm is not achievable in practice. The approximate or metaheuristic algorithms seek the right solution and are close to optimal. This method lowers the calculation time compared to the previous method.

In this paper, Energy Valley Optimizer (EVO) is proposed as a novel metaheuristic algorithm inspired by advanced physics principles regarding stability and different modes of particle decay. Since the basic principles of the decay process through different particles in physics are used as the main idea of the EVO, the originality of this study may be seen from an inspirational standpoint, while the complexity level of the test functions used is also being examined for the first time in this study. The performance of different algorithms must be done in the same conditions and under the same problems, and under various examples, the superiority of each algorithm cannot be confirmed or denied. Hence, 20 unconstrained mathematical test functions are utilized in different dimensions to evaluate the proposed algorithm's performance. For statistical purposes, 100 independent optimization runs are conducted for determining the statistical measurements as the mean, standard deviation, and the required number of objective function evaluations. A predefined stopping criterion is also considered based on a maximum number of 150,000 objective function evaluations and a tolerance of 1×10^{-12} for the global best values of the considered problems. Some well-known statistical analyses, including the Kolmogorov–Smirnov, Wilcoxon, Mann–Whitney, Kruskal–Wallis, and Post-Hoc analysis, are also utilized for comparative purposes. One of the most significant shortcomings of the newly developed metaheuristic algorithms is the simplicity of the evaluation test functions; therefore, two of the latest Competitions on Evolutionary Computation (CEC), the CEC 2020 on bound constraint optimization⁶⁰ and CEC 2020 on real-world optimization⁶¹, and also the “Big O notation” are considered for comparing the results of the EVO to the most successful state-of-the-art algorithms. However, being parameter-free, fast convergence behaviour, and the lowest possible objective function evaluation could be deemed the privileges of the EVO. In stark contrast, the EVO cannot provide exact solutions; in other words, like other metaheuristic algorithms, the EVO is an approximate algorithm.

Since the proposed EVO is an algorithm developed based on some general and advanced principles of physics, the most important aspect of this algorithm is the conformity between the concept and the mathematical model for which the EVO has higher levels of adaptation between these two aspects. The second factor is the complexity of the algorithm while three new position vectors are created in the main search loop so the recent development in computer science regarding software and hardware allow experts to create algorithms with higher levels of complexity. The third factor is the dynamic configuration of the main loop of EVO in which the exploration and exploitation procedures are conducted by searching the variables' and candidate's spaces simultaneously to reach the global best solution. Furthermore, despite the fact that there are several metaheuristic algorithms, more are always needed. Besides, the introduction of novel metaheuristic optimization algorithms is continuously ongoing. The creation of new metaheuristics is advantageous to science since they might improve the precision or effectiveness of the optimization procedure for a host of problems. This assertion drives our attempts to suggest a unique metaheuristic algorithm inspired by the cutting-edge physics concepts regarding particles' stability and decay.

The main contributions of this study are as follows:

- Advanced physics concepts concerning stability and various forms of decay in particles are examined and analyzed to develop a mathematical model of a metaheuristic algorithm.
- A unique physics-inspired algorithm as Energy Valley Optimizer (EVO) is developed using the mentioned model.
- The EVO's solution updating is dependent on the particles' enrichment bound, position vector, and stability level.

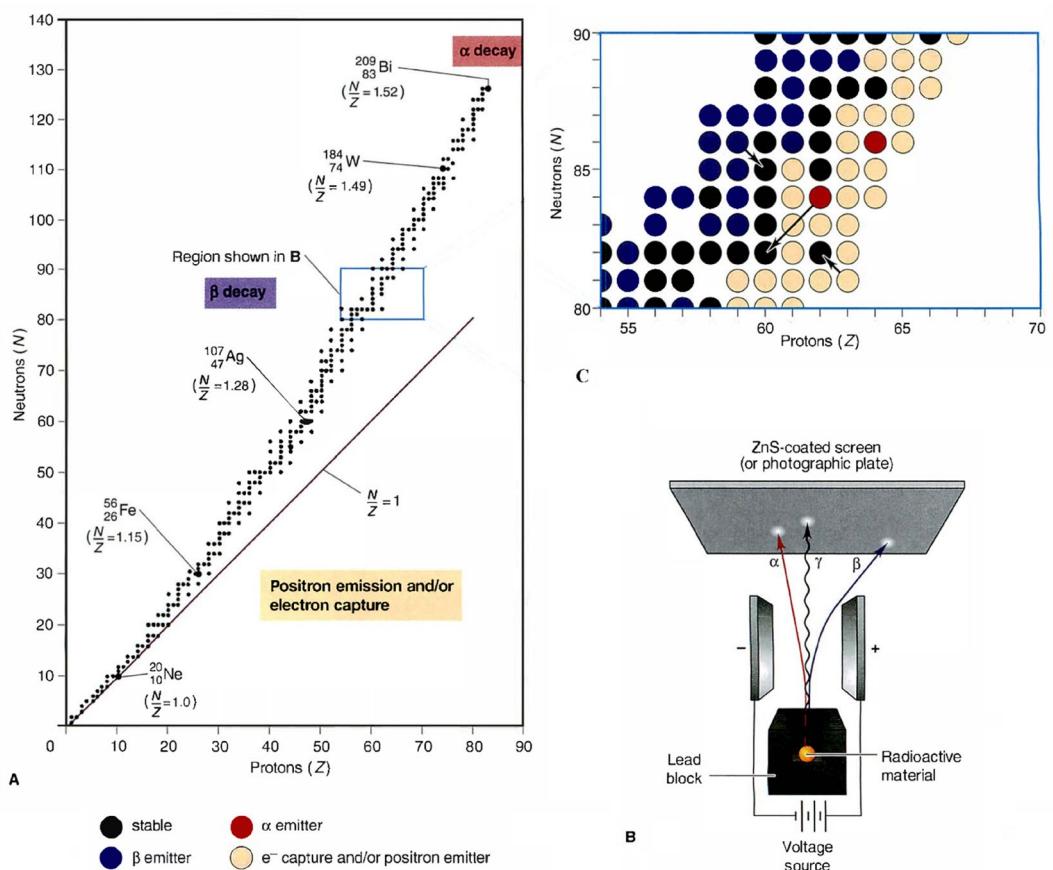


Figure 1. (A) Stability band for particles (B) Process of emission (C) Different types of decay⁶².

- EVO's performance is extensively evaluated against a set of twenty benchmark functions and real-world engineering design problems.
- The proposed EVO is compared to a plethora of cutting-edge metaheuristic algorithms.

The rest of the paper is divided into the following sections. In Sect. 2, the inspiration and mathematical model of the proposed EVO are presented. The numerical investigations, including 20 of the best-known mathematical test functions and statistical analysis, including the Kolmogorov–Smirnov, Wilcoxon, and Kruskal–Wallis analysis are indicated in Sects. 3, and 4 respectively. The CEC 2020 complexity analysis and Big O notation are represented in Sect. 5 while the CEC 2020 real-world constrained optimization problems such as Speed Reducer, Hydro-Static Thrust Bearing, Ten-Bar Truss, and also Rolling Element Bearing are elucidated in Sect. 6. Finally, in Sect. 7, the core findings of this study are presented as concluding remarks.

Energy valley optimizer

Inspiration. Physical reaction refers to colliding two particles or external subatomic particles to produce new particles. In the universe, a great majority of particles are assumed to be unstable except for the stable ones which remain intact indefinitely. The unstable particles tend to emit energy through disintegration or decay, while the overall decay rate is somehow different in various types of particles. In the decaying process, a particle with lower energy is generated while the extra energy is bring-off through the emission process. Energy valley concerns the stability of particles based on their binding energy and interactions with other particles. The direct observation of multiple phenomena has led the experts to extract some valuable patterns for defining the decay in particles. The most crucial challenge in this area is determining the particles' stability bound by considering the number of neutrons (N) and protons (Z) and the N/Z ratio. The $N/Z \approx 1$ refers to the stable, lightweight particle, while for heavier ones, a larger value for N/Z is considered as the stability band. Based on the stability level of the particles, each particle tends to increase its stability level by shifting its N/Z ratio and moving toward the stability band or energy valley. In this regard, the neutron enrichment levels of particles play an essential role in this action. The neutron-rich particles positioned above the stability bound undergo a decay process and require so many neutrons for stability purposes. On the other hand, the neutron-poor particles, which require too few neutrons for stability purposes, tend to undergo electron capture or positron emission to move toward the energy valley or stability band⁶²; in Fig. 1.A, these aspects are illustrated schematically.

In the decay process, a particle with a lower energy level is generated while excessive energy is emitted. There are three types of emissions that determine the decay process in particles with different levels of stability. The

alpha (α) particles denote the dense and positively charged particles identical to the helium. The beta (β) particles are negatively charged particles characterized as electrons with higher speeds. The gamma (γ) rays represent photons with higher levels of energy. In Fig. 1B, the overall behaviour of these types of emissions is illustrated inside the electric field, considering the α particles curve toward the negative plate to a small extent. In contrast, the β particles curve toward the positive plate greatly, and the electric field does not affect the γ rays.

Based on the presented details of the emission process, there are three types of decay, known as alpha, beta, and gamma decay derived from the previously mentioned emission types. In the alpha decay, the loss of an α particle is determined in which the N and Z values in the N/Z ratio are reduced per emission process. In beta decay, the ejection of a β particle is a concern in which the N/Z ratio is increased by reducing the N and increasing the Z values. In gamma decay, omitting a γ photon with higher energy levels from an excited particle is concerned, while no change in N/Z values is necessary for this mode of decay. The schematic presentation of these aspects is presented in Fig. 1C.

Most of the recently developed metaheuristic algorithms are some reasonable higher-level searching techniques formulated by miscellaneous inspirational concepts to provide one or numerous good solutions for a maximization or minimization optimization problem, particularly with imperfect or incomplete information. The searching processes are primarily conducted among some initial candidates determined randomly. The higher-level procedures of the metaheuristics try to improve the overall standing of these candidates in a step-by-step manner. Hence, the principles of the decay process through different particles can be a great starting point for a novel algorithm in which particles' tendency to achieve a stable point can be utilized as the inspirational concept for performance improvement of the solution candidates.

Mathematical model. In this section, the EVO is presented as an optimization algorithm in detail by means of the previously described principles of physics. In the first step, the initialization process is conducted in which the solutions candidates (X_i) are assumed to be particles with different levels of stability in the search space, which is assumed to be a specific part of the universe.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \dots x_1^j \dots x_1^d \\ x_2^1 x_2^2 \dots x_2^j \dots x_2^d \\ \vdots \\ x_i^1 x_i^2 \dots x_i^j \dots x_i^d \\ \vdots \\ x_n^1 x_n^2 \dots x_n^j \dots x_n^d \end{bmatrix}, \begin{cases} i = 1, 2, \dots, n. \\ j = 1, 2, \dots, d. \end{cases} \quad (1)$$

$$x_i^j = x_{i,\min}^j + \text{rand.}(x_{i,\max}^j - x_{i,\min}^j), \quad \begin{cases} i = 1, 2, \dots, n. \\ j = 1, 2, \dots, d. \end{cases} \quad (2)$$

where n denotes on the total number of particles (solution candidates) in the universe (search space); d is the dimension of the considered problem; x_i^j is the j th decision variable for determining the initial position of the i th candidate; $x_{i,\min}^j$ and $x_{i,\max}^j$ represent the lower and upper bounds of the j th variable in the i th candidate; rand is a uniformly distributed random number in the range of $[0, 1]$.

In the second step of the algorithm, the Enrichment Bound (EB) for the particles is determined, which is utilized for considering the differences between the neutron-rich and neutron-poor particles. For this purpose, the objective function evaluation for each of the particles is conducted and determined as the Neutron Enrichment Level (NEL) of the particles. The mathematical presentation of these aspects is as follows:

$$EB = \frac{\sum_{i=1}^n NEL_i}{n}, \quad i = 1, 2, \dots, n. \quad (3)$$

where NEL^i is the neutron enrichment level of the i th particle, and EB is the enrichment bound of the particles in the universe.

In the third step, the stability levels of the particles are determined as follows based on the objective function evaluations:

$$SL_i = \frac{NEL_i - BS}{WS - BS}, \quad i = 1, 2, \dots, n. \quad (4)$$

where SL^i is the stability level of the i th particle, BS and WS are the particles with best and worst stability levels inside the universe equivalent to the minimum and maximum values of so far found objective function values.

In the main search loop of the EVO, if the neutron enrichment level of a particle is higher than the enrichment bound ($NEL_i > EB$), the particle is assumed to have a larger N/Z ratio, so the decay process utilizing alpha, beta, or gamma schemes are in perspective. In this regard, a random number is generated in the range of $[0, 1]$, which mimics the Stability Bound (SB) in the universe. If the stability level of a particle is higher than the stability bound ($SL_i > SB$), the alpha and gamma decay is considered to happen since these two decays are probable for heavier particles with higher stability levels. Based on physics principles regarding alpha decay (Fig. 2), α rays are emitted to improve the product's stability level in the physical reaction. This aspect can be mathematically formulated as one of the position updating schemes of the EVO in which a new solution candidate is generated. For this purpose, two random integers are generated as Alpha Index I in the range of $[1, d]$, which denotes the number

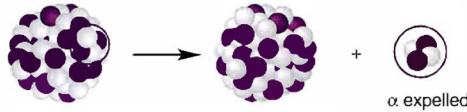
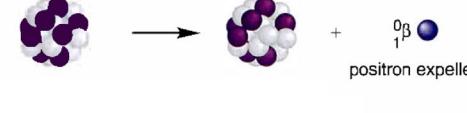
Mode	Emission	Decay Process	
α Decay	α (${}^4_2\text{He}$)	Reactant	Product + 
β Decay [†]	$-{}^0_1\beta$	${}^1_0\text{n}$	\rightarrow ${}^1_1\text{p}$ + ${}^0_{-1}\beta$ β expelled
Positron emission [†]	${}^0_1\beta$	$h\nu$ + high-energy photon	\rightarrow 
Electron capture [†]	x-ray photon	${}^0_{-1}\text{e}$ + ${}^1_1\text{p}$	\rightarrow ${}^1_0\text{n}$ absorbed from low-energy orbital
γ Emission	${}^0_0\gamma$	excited	\rightarrow stable + ${}^0_0\gamma$ γ emission

Figure 2. Different modes of decay⁶².

of emitted rays, and Alpha Index II in the range of [1, Alpha Index I], which defines which α rays to be emitted. The emitted rays are decision variables in the solution candidate, which are removed and substituted by the rays in particle or candidate with the best stability level (X_{BS}). These aspects are mathematically formulated as follows:

$$X_i^{\text{New}1} = X_i \left(X_{BS} \left(x_i^j \right) \right), \begin{cases} i = 1, 2, \dots, n. \\ j = \text{Alpha Index II}. \end{cases} \quad (5)$$

where $X_i^{\text{New}1}$ is the newly generated particle in the universe, X_i is the current position vector of the i th particle (solution candidates) in the universe (search space), X_{BS} is the position vector of the particle with the best stability level, x_i^j is the j th decision variable or emitted ray.

Besides, in gamma decay, γ rays are emitted to improve the excited particles' stability level (Fig. 2), so this aspect can be mathematically formulated as another position-updating process of the EVO in which a new solution candidate is generated. For this purpose, two random integers are generated as Gamma Index I in the range of [1, d], which denotes the number of emitted photons, and Gamma Index II in the range of [1, Gamma Index I], which defines which photons to be considered in the particles. The photons in the particles are decision variables in the solution candidate, which are removed and substituted by a neighboring particle or candidate (X_{Ng}), which mimics the interaction of the excited particles with other particles or even magnetic fields. In this regard, the total distance between the considered particle and the other ones is calculated as follows, and the nearest particle is utilized for this purpose:

$$D_i^k = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \begin{cases} i = 1, 2, \dots, n. \\ k = 1, 2, \dots, n - 1. \end{cases} \quad (6)$$

where D_i^k is the total distance between the i th particle and the k th neighbouring particle, and (x_1, y_1) and (x_2, y_2) denote the coordinates of the particles in the search space.

Using these actions, the position updating process for generating the second solution candidate in this phase is conducted as follows:

$$X_i^{\text{New}2} = X_i \left(X_{Ng} \left(x_i^j \right) \right), \begin{cases} i = 1, 2, \dots, n. \\ j = \text{Gamma Index II}. \end{cases} \quad (7)$$

where $X_i^{\text{New}2}$ is the newly generated particle in the universe, X_i is the current position vector of the i^{th} particle (solution candidate) in the universe (search space), X_{Ng} is the position vector of the neighbouring particle around the i^{th} particle, and x_i^j is the j^{th} decision variable or emitted photon.

If the stability level of a particle is lower than the stability bound ($SL_i \leq SB$), beta decay is considered to happen because this type of decay happens in more unstable particles with lower stability levels. Based on the physics principles regarding beta decay (Fig. 2), β rays are expelled from the particles to improve the stability level of the particle, so a big jump in the search space is supposed to be conducted due to the higher levels of instability in these particles. In this regard, a position updating process is conducted for the particles in which a controlled movement toward the particle or candidate with the best stability level (X_{BS}) and the centre of particles (X_{CP}) is performed. These aspects of the algorithm mimic the particles' tendency to reach the stability band in which most of the known particles are positioned near this band, and most of them have higher levels of stability (Fig. 1a and b). These aspects are mathematically formulated as follows:

$$X_{CP} = \frac{\sum_{i=1}^n X_i}{n}, i = 1, 2, \dots, n. \quad (8)$$

$$X_i^{\text{New}1} = X_i + \frac{(r_1 \times X_{BS} - r_2 \times X_{CP})}{SL_i}, i = 1, 2, \dots, n. \quad (9)$$

where $X_i^{\text{New}1}$ and X_i are the upcoming and current position vectors of the i^{th} particles (solution candidates) in the universe (search space), X_{BS} is the position vector of the particle with the best stability level, X_{CP} is the position vector for the centre of particles, SL_i is the stability level of the i^{th} particle, r_1 and r_2 are two random numbers in the range of $[0, 1]$ which determine the amount of particles' movement.

In order to improve the exploitation and exploration levels of the algorithm, another position updating process is conducted for the particles employing beta decay in which a controlled movement toward the particle or candidate with the best stability level (X_{BS}) and a neighbouring particle or candidate (X_{Ng}) is performed while the stability level of the particle does not affect the movement process. These aspects are mathematically formulated as follows:

$$X_i^{\text{New}2} = X_i + (r_3 \times X_{BS} - r_4 \times X_{Ng}), i = 1, 2, \dots, n. \quad (10)$$

where $X_i^{\text{New}2}$ and X_i are the upcoming and current position vectors of the i^{th} particle (solution candidates) in the universe (search space), X_{BS} is the position vector of the particle with the best stability level, X_{Ng} is the position vector of the neighbouring particle around the i^{th} particle, and r_3 and r_4 are two random numbers in the range of $[0, 1]$ which determine the amount of particles' movement.

If the neutron enrichment level of a particle is lower than the enrichment bound ($NEL_i \leq EB$), the particle is assumed to have a smaller N/Z ratio, so the particle tends to undergo electron capture or positron emission to move toward the stability band. In this regard, a random movement in the search space is determined for considering these sorts of movements as follows:

$$X_i^{\text{New}} = X_i + r, i = 1, 2, \dots, n. \quad (11)$$

where X_i^{New} and X_i are the upcoming and current position vectors of the i^{th} particles (solution candidates) in the universe (search space), and r is a random number in the range of $[0, 1]$ which determines the amount of particles' movement.

At the end of the main loop of the EVO, there are only two newly generated position vectors for each of the particles as $X_i^{\text{New}1}$ and $X_i^{\text{New}2}$ if the enrichment level of the particle is higher than the enrichment bound, while for the particle with a lower enrichment level, only X_i^{New} is generated as a new position vector. At each state, the newly generated vectors are merged with the current population, and the best particles participate in the following search loop of the algorithm. A boundary violation flag is determined for the decision variables which go beyond the predefined upper and lower bounds, while a maximum number of objective function evaluations or a maximum number of iterations can be utilized as a termination criterion. The pseudo-code of the EVO is presented in Fig. 3, while the flowchart of this algorithm is provided in Fig. 4.

Overall, one of the primary areas of physics is studying the interactions of particles, which focuses on the unique properties of particles and the elements that make them up as well as their interactions with other particles. While the overall rate of decay varies across different kinds of particles, unstable particles tend to demonstrate emission by disintegration or decay. Determining the bound stability of particles by taking into account the number of neutrons (N) and protons (Z), and determination of the N/Z ratio is the toughest challenge in this field. The neutron-rich particles located above the stability bound, however, experience decay and need many neutrons to be stable. Consequently, the principles of the decay process via various particles can be a fantastic starting point for a unique algorithm in which the solution candidates' performance can be improved by taking inspiration from particles' propensity to reach a stable point.

Considering the new things of this algorithm, the main loop of the algorithm includes three position updating process. Two of these procedures occur in decision variables for which the exploration process is conducted while one position updating process occurs in the solution candidates for which the exploitation is satisfied. The challenging part of this algorithm is the fact that the exploration part may lead the algorithm to local optimum solutions but the other part tries to tune the previous solutions to reach the global best candidate.

```

Procedure Energy Valley Optimizer (EVO)
    Determine initial positions of solution candidates ( $X_i$ ) as particles in the search space
    Evaluate fitness values for initial solution candidates as Neutron Enrichment Level ( $NEL_i$ )
    while Iteration (Number of Function Evaluation) < Maximum number of iterations (Function Evaluations)
        Determine Enrichment Bound (EB) of the particles
        Determine the particle with the best stability level ( $X_{BS}$ ).
        for  $i=1:n$ 
            Determine Stability Level ( $SL_i$ ) of the  $i$ th particle
            Determine Neutron Enrichment Level ( $NEL_i$ ) of the  $i$ th particle
            if  $NEL_i > EB$ 
                Determine Stability Bound (SB) of the particles
                if  $SL_i > SB$ 
                    Generate Alpha Index I and II
                    for  $j=1: Alpha$  Index II
                         $X_i^{New1} = X_i \left( X_{BS} \left( x_i^j \right) \right)$ 
                    end
                    Generate Gamma Index I and II
                    Determine a neighbouring particle ( $X_{Ng}$ )
                    for  $j=1: Gamma$  Index II
                         $X_i^{New2} = X_i \left( X_{Ng} \left( x_i^j \right) \right)$ 
                    end
                    else if  $SL_i \leq SB$ 
                        Determine Centre of Particles ( $X_{CP}$ )
                         $X_i^{New1} = X_i + (r_1 \times X_{BS} - r_2 \times X_{CP}) / SL_i$ 
                        Determine a neighbouring particles ( $X_{Ng}$ )
                         $X_i^{New2} = X_i + (r_3 \times X_{BS} - r_4 \times X_{Ng})$ 
                    end
                    else if  $NEL_i \leq EB$ 
                         $X_i^{New} = X_i + r$ 
                    end
                end
            end while
            Return the particle with the best stability level ( $X_{BS}$ ).
    end Procedure

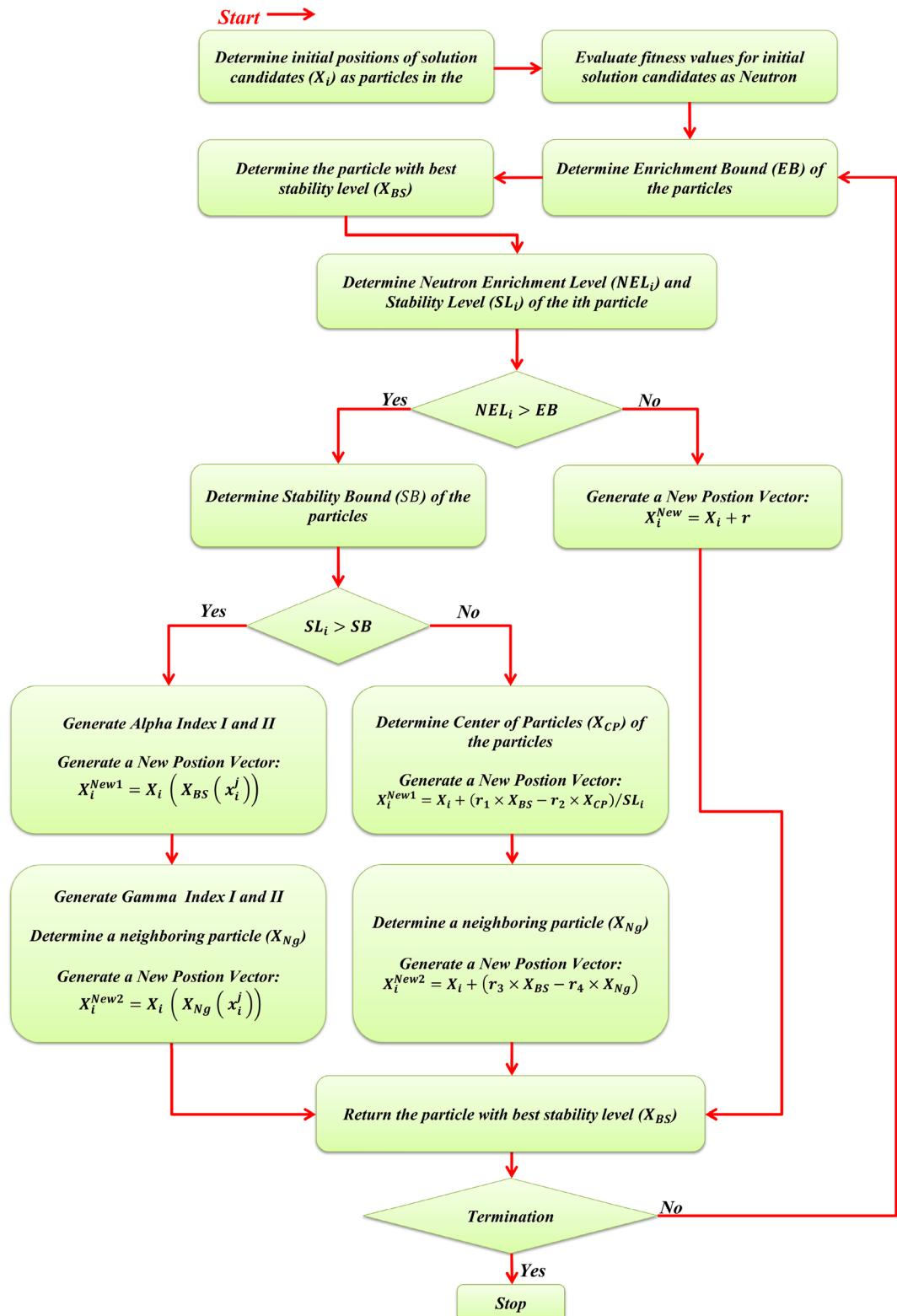
```

Figure 3. Pseudo-code of the EVO.

Mathematical test functions

In this section, 20 of the well-known mathematical test functions are utilized to evaluate the EVO's performance as a novel metaheuristic algorithm. These functions are as Ackley 1, Alpine 1, Chung Reynolds, Exponential, Inverted cosine wave, Pinter, Rastrigin, Salomon, Schwefel 1.2, Schwefel 2.21, Schwefel 2.25, Sphere, Step 3, Trigonometric 2, W / Wavy, and Xin-She Yang 6 functions while the first 10 functions are considered with 50 dimensions and the other half are with 100 dimensions. The global best for the fourth, fifth and eighteenth functions are as -1, -49 and 1 respectively while for the rest of them, the global best is as 0. For statistical purposes, 100 independent optimization runs are conducted to determine the statistical measurements as the mean, standard deviation, and the required number of objective function evaluations. A predefined stopping criterion is also considered based on a maximum number of 150,000 objective function evaluations and tolerance of 1×10^{-12} for the global best values of the supposed problems.

The parameter settings of the optimization algorithms are shown in Table 2, and all tests to evaluate the EVO's performance were conducted with 50 populations using a PC with the detailed parameters shown in Table 3. In Table 4, the best, mean and Standard Deviation (SD) of results for EVO and other alternative algorithms, including the Ant Colony Optimization (ACO), Harmony Search (HS), Firefly Algorithm (FA), Multiverse Optimizer (MVO), Interior Search Algorithm (ISA), and Cuckoo Search Algorithm (CSA) in dealing with these benchmark mathematical functions are presented. Based on the provided results in these tables, it is obvious that the proposed EVO can outperform the other algorithms in most cases. Considering the required Objective Function Evaluations (OFE), ACO, FA, HS, and CSA required a mean of 150,000 OFE while the MVO with 149,900, ISA with 141,263.90, and EVO with 43,060.28 have better performance. Meanwhile, the same random state in each of the considered algorithms for each optimization run is set to a fixed state to have an unbiased and fair judgment regarding the overall performance of the EVO.

**Figure 4.** Flowchart of the EVO.

The convergence curves of 100 optimization runs conducted by the EVO in dealing with the mathematical test functions are illustrated in Fig. 5. The best and worst runs, alongside the mean of all runs, are highlighted. By considering the convergence history of EVO, it can be concluded that the proposed algorithm is capable of

Algorithm	Parameter	Value
ACO	<i>Q</i>	1
	<i>Alpha</i> (α)	1
	<i>Beta</i> (β)	1
	<i>rho</i>	0.05
HS	<i>bw</i>	0.2
	<i>HMCR</i>	0.95
	<i>PAR</i>	0.3
FA	<i>Alpha</i> (α)	0.25
	<i>Beta</i> (β)	0.2
	<i>Gamma</i> (γ)	1
MVO	<i>WEP_{max}</i>	1
	<i>WEP_{min}</i>	0.2
ISA	<i>Alpha</i> (α)	0.2
CSA	<i>Pa</i>	0.25
	<i>nd</i>	0.15

Table 2. The parameter settings of the optimization algorithms.

Name		Detailed settings
Hardware	CPU	CORE i7
	Frequency	2.8 GHz
	RAM	8 GB
	Hard drive	2 TB

Table 3. Details of the utilized system in optimization process in the current study.

performing fast optimization procedures in most of the considered problems, while the algorithm does not need to conduct predefined 150,000 objective function evaluations and is capable of reaching the tolerance of 1×10^{-12} in a swift way.

Statistical analysis

In this section, the results of the EVO and other alternative metaheuristic algorithms in dealing with the mathematical test functions are utilized for conducting a comprehensive statistical analysis. In this regard, four of the well-known statistical tests as the Kolmogorov Smirnov (KS) test for normality control, Wilcoxon (W) signed ranks test for comparing in a two-by-two procedure the summation and mean of the metaheuristics' ranks, and the Kruskal Wallis (KW) test for evaluating the overall rankings of different metaheuristic algorithms by comparing the mean of the metaheuristics' ranks. In Table 5, the results of the KS test are provided in which the p-value of this test is less than 0.05, so the hypothesis in the normal distribution of data is satisfied, and the non-parametric statistical tests can be utilized for further investigations.

In Fig. 6, the results of the W test are presented in which the mean ranks of different metaheuristic algorithms are provided and compared in a two-by-two manner regarding the best results, while the metaheuristics with a smaller mean of ranks are superior to the other algorithm. The EVO can provide better results with smaller means of ranks in most cases based on the results. In Fig. 7, the results of the KW statistical test, including the mean of ranks by considering all of the data sets, are presented in which the EVO is capable of outranking the other algorithms in all of the considered data sets.

CEC 2020 complexity analysis & Big O notation

In most of the recently developed metaheuristic algorithms, the computational efforts of the algorithms in dealing with complex optimization problems have been of significant concern due to the increasing interest of artificial intelligence experts to provide computationally efficient algorithms for optimization purposes. In this regard, the computational time procedure of the CEC 2020 benchmark suit on bound constrained⁶⁰ is utilized in which T_0 represents the run time of a predefined mathematical process, T_1 is the computational time of G1 function by considering 200,000 objective function evaluations, T_2 is the computational time of the proposed algorithm (EVO) for 200,000 objective function evaluations of G1 function, and \bar{T}_2 is the mean of five individual T_2 . In Table 6, the computational time of the proposed EVO algorithm and other approaches are provided in which the capability of the EVO in competing with other algorithms is in perspective.

One of the well-known procedures for evaluating the computational complexity of the algorithms is the “Big O notation” which is frequently used in computer science and is adopted in this paper for further investigations on EVO algorithm. By considering NP and D as the total number of initial solution candidates and the dimension of

No.		Alternative metaheuristic algorithms						
		ACO	FA	HS	MVO	CSA	ISA	EVO
F_1	Best	20.70461	20.38123	3.655346	0.069087	3.424855	1.41E-05	0
	Mean	20.97241	20.50565	4.372899	3.203908	9.174743	1.56	0.125987
	SD	0.086512	0.050224	0.313316	6.133982	3.58599	4.09E-01	1.25982
F_2	Best	75.36033	5.423342	0.408871	1.955532	9.844303	7.06E-07	0
	Mean	95.46201	8.546599	0.767617	7.655639	20.88773	4.92E-04	5.47E-13
	SD	6.486329	1.509117	0.150857	3.045156	4.369272	1.16E-03	5.47E-12
F_3	Best	19,933,773	1.09E+09	31,448.5	0.006724	0.044259	0	0
	Mean	75,539,630	1.72E+09	67,983.71	0.024618	6.177559	1.28E-08	0
	SD	34,603,873	3.04E+08	24,225.86	0.010664	8.326967	9.02E-08	0
F_4	Best	-0.79993	-0.99974	-0.99117	-1	-1	-1	-1
	Mean	-0.65791	-0.99967	-0.98724	-0.99999	-1	-1	-1
	SD	0.063929	3.13E-05	0.002225	1.65E-06	1.08E-06	5.64E-09	0
F_5	Best	-4.85263	-13.5479	-39.3724	-13.6182	-15.0676	-22.634	-49
	Mean	-2.99902	-10.0326	-36.6063	-8.73447	-11.5189	-12.5943	-48.5529
	SD	0.411566	1.273413	1.04247	1.508705	1.058497	2.991059	4.469695
F_6	Best	15,547.54	4403.94	651.3584	2841.421	1009.47	0.055643	0
	Mean	18,151.53	5395.454	1500.041	6914.513	3839.607	930.2275	2.94E-06
	SD	1093.599	411.362	584.8083	2225.258	1408.859	957.1977	2.94E-05
F_7	Best	511.8067	112.6812	15.96008	110.5245	210.6031	51.73782	0
	Mean	587.6085	155.6537	21.72945	211.2711	265.203	131.404	3.64E-07
	SD	25.18078	15.83632	2.781115	42.39096	18.5789	38.00434	3.63E-06
F_8	Best	10.05457	18.11648	2.599891	0.599873	1.399914	0.599873	0
	Mean	12.34569	21.19853	3.134814	0.830873	2.029959	1.104873	0.009265
	SD	0.966472	0.822021	0.27839	0.099184	0.226778	0.190361	0.081186
F_9	Best	8.49E-07	5.35E-11	1.08E-09	0	1.43E-12	0	0
	Mean	2.98E-03	2.54E-05	1.72E-05	9.25E-09	1.67E-08	0	0
	SD	7.70E-03	7.88E-05	3.90E-05	1.75E-08	3.39E-08	0	0
F_{10}	Best	77.50366	71.56383	14.6766	0.460693	3.379566	0.450844	0
	Mean	91.58769	83.56584	17.55798	1.319453	7.234517	1.909486	0.503791
	SD	2.434241	4.747136	1.236382	0.505237	1.954402	0.958307	4.897183
F_{11}	Best	56.83658	34.3526	3.065453	0.088745	0.023951	0.000271	0
	Mean	68.01628	38.76511	3.687095	0.148389	0.111745	0.074045	1.10E-06
	SD	3.374358	1.686266	0.247853	0.023546	0.058802	0.114694	1.10E-05
F_{12}	Best	333,012.9	284.4142	3805.585	0.424549	53.86077	0.013213	0
	Mean	474,307.6	471.6474	5370.455	0.904045	113.9655	2.278516	0.259769
	SD	44,714.95	90.6102	635.0582	0.30938	29.47293	5.882736	2.597688
F_{13}	Best	1.05E+09	5.38E+08	8,042,041	0.188584	3122.341	0.018164	0
	Mean	1.42E+09	6.99E+08	12,104,315	0.501756	18,977.06	340.8279	0.001432
	SD	1.12E+08	58,432,546	1,604,566	0.160563	8842.065	457.0865	0.01432
F_{14}	Best	71,819.64	15.43768	954.7067	0.132864	10.13445	12.38795	46.96118
	Mean	97,864.79	19.82098	1262.973	0.269682	49.44706	22.25387	73.26776
	SD	11,197	1.451369	142.6338	0.075034	40.36055	4.672117	10.17666
F_{15}	Best	70,339.89	16.00331	888.4887	0.130584	12.04421	16.55338	26.27337
	Mean	97,836.47	19.52131	1240.962	0.279098	54.91552	24.86934	72.60386
	SD	10,159.17	1.460158	139.9793	0.072831	56.50644	4.299523	11.2144
F_{16}	Best	1872.414	0.527636	85.29447	0.008924	0.171064	3.42E-12	0
	Mean	2199.699	0.668883	108.5625	0.013944	0.89789	2.23E-09	0
	SD	134.2236	0.058032	10.19812	0.002377	0.567639	1.20E-08	0
F_{17}	Best	223,877	130,660	8707	36	173	166	22
	Mean	268,117.9	148,673.9	10,812.77	84.51	299.23	559.02	32.41
	SD	13,451.02	7049.15	969.7187	31.88759	65.20935	160.5713	4.878514
F_{18}	Best	5,588,993	5,588,993	195,795.3	867.3468	3839.556	1924.694	262.1536
	Mean	6,702,584	6,702,584	267,328.7	1195.725	6439.726	5520.177	420.8609
	SD	337,357.7	337,357.7	25,817.66	140.2667	1114.555	2013.876	839.9876

Continued

No.		Alternative metaheuristic algorithms						
		ACO	FA	HS	MVO	CSA	ISA	EVO
F_{19}	Best	0.857868	0.47082	0.083849	0.504413	0.401207	0.318924	0
	Mean	0.892505	0.541511	0.106005	0.580524	0.494181	0.5283	0.002477
	SD	0.009596	0.024226	0.008691	0.031767	0.027936	0.047698	0.023212
F_{20}	Best	0.000205	2.35E-05	2.79E-06	1.99E-05	1.28E-06	7.16E-06	9.65E-06
	Mean	0.018252	1.44E-03	1.85E-03	2.03E-03	1.73E-03	1.15E-03	1.13E-03
	SD	0.021241	1.14E-03	2.03E-03	1.80E-03	1.51E-03	1.02E-03	1.37E-03

Table 4. The best, mean and standard deviation of different metaheuristics in dealing with mathematical functions.

the optimization problem, respectively, the computational complexity of generating position vectors and calculating objective function are as $O(NP \times D)$ and $O(NP) \times O(F(x))$ regarding the fact that $F(x)$ represents the objective function of the optimization problem. In the main loop of the EVO, each line has a computational complexity of $MxItr$ as the total number of iterations. The position updating process for each of the solution candidates in this loop can be conducted in two phases as if $NEL_i > EB$, two new position vectors are created in one of the $SL_i > SB$ and $SL_i \leq SB$ subphases so the computational complexity of $O(MxItr \times NP \times D \times 2)$ is determined in this phase. Regarding $NEL_i \leq EB$, only one new position vector is generated, so the complexity of $O(MxItr \times NP \times D)$ is concerned in this case. For objective function evaluation in these two phases, the complexity of $O(\text{MaxIter} \times NP \times D \times 3) \times O(F(x))$ and $O(\text{MaxIter} \times NP \times D) \times O(F(x))$ are determined, respectively.

CEC 2020 real-world constrained optimization

Most of the time, the capability of the metaheuristic algorithms is considered through real-world optimization problems in which some sort of design constraints alongside the bound constraints should be handled for having feasible solutions. For this purpose, the real-world constraint optimization problems of CEC 2020⁶¹ are utilized in this paper. In Table 7, a summary of these engineering design problems is presented, while the complete mathematical formulations can be found in the literature. The schematic presentation of these problems is also presented in Figs. 8, 9, 10, 11. A total number of 30 independent optimization runs have been conducted using 20,000 function evaluations for statistical purposes. For constraint handling purposes, however, the well-known penalty approach with static co-efficient is utilized in the current study.

In Tables 8 and 9, the best and statistical results of EVO and other algorithms in dealing with the speed reducer problem are presented in which the optimum design variables alongside the design constraints are provided. Based on the results of best optimization runs conducted by different methods, EVO can provide 2994.42, which is the best among other approaches. Besides, EVO provides the means and worst of 2994.44 and 2994.46, respectively, which are better than other methods' results.

The best and statistical results of different algorithms, including the proposed EVO in dealing with the hydrostatic thrust bearing design problem, are provided in Tables 10 and 11, respectively, including the optimum design variables and the design constraints. In dealing with the problem, EVO can provide the problem, EVO can provide 1619.55, which is the best among other approaches, while the best so far found in other approaches is for CGO, which calculated 1621.24. Meanwhile, EVO provides the means and worst of 1730.09 and 1899.34, respectively, demonstrating some superiority compared to other approaches.

Tables 12 and 13 provide the best and statistical results of different algorithms, including the proposed EVO, in dealing with the ten-bar truss design problem. It can be concluded that the proposed EVO can provide 524.92, which is much better than the previously reported results of 529 and 530. In addition, EVO provides better means, and worst values demonstrate this novel algorithm's superiority compared to other approaches.

Tables 14 and 15 present the best and statistical results of EVO and other algorithms in dealing with the rolling element bearing design problem (as a maximization problem), in which the optimum design variables and the design constraints are provided. Based on the best optimization runs conducted by different methods, EVO can provide 81,859.74, while the ALO with 85,546.63 is the best algorithm in providing the best result in this case. However, the EVO is capable of competing with the ALO and other approaches by providing mean and worst values of 81,110.32 and 80,212.09, respectively.

Conclusions

Energy Valley Optimizer (EVO) is proposed as a novel metaheuristic algorithm inspired by the advanced principles of physics regarding stability and different modes of decay in particles. For evaluation purposes, 20 unconstrained mathematical test functions with 100 independent optimization runs and the maximum number of 150,000 objective function evaluations alongside the latest Competitions on Evolutionary Computation (CEC), including the CEC 2020 on real-world optimization are utilized. The key results and main findings of this research paper are summarised as follows:

- In dealing with the unconstrained mathematical test functions, EVO can outrank the other alternative metaheuristic algorithm and converge to the global best solutions in most cases.

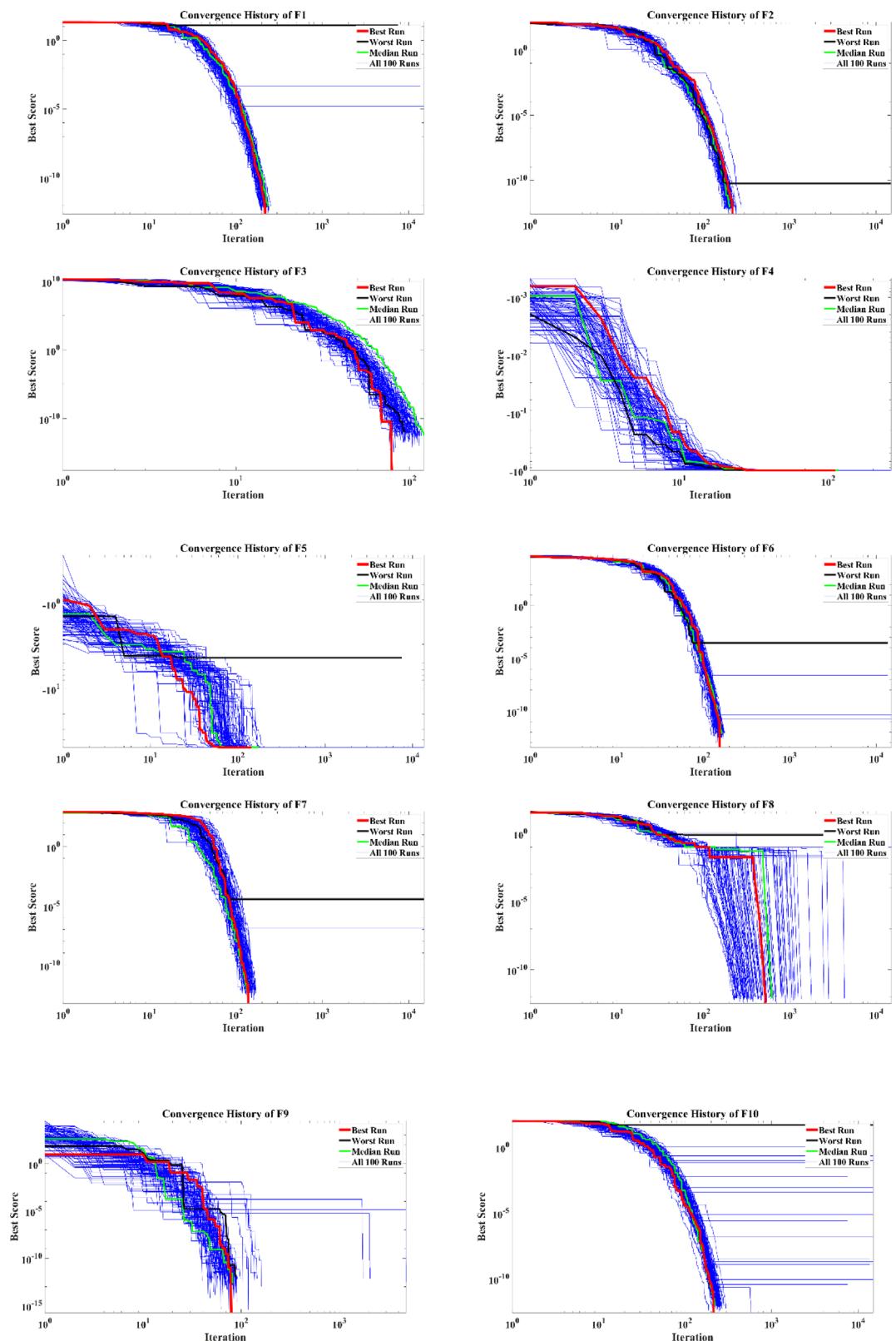
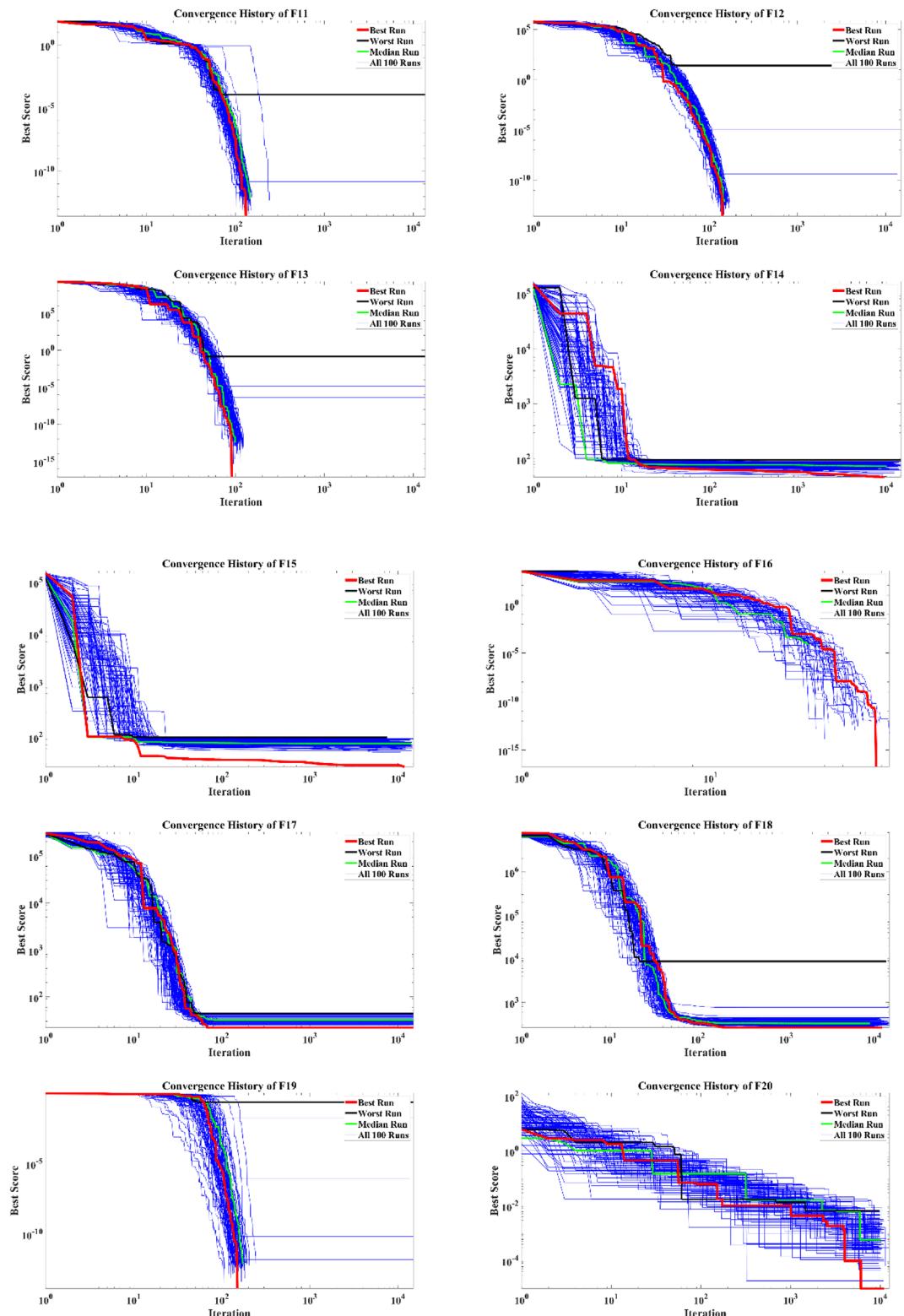
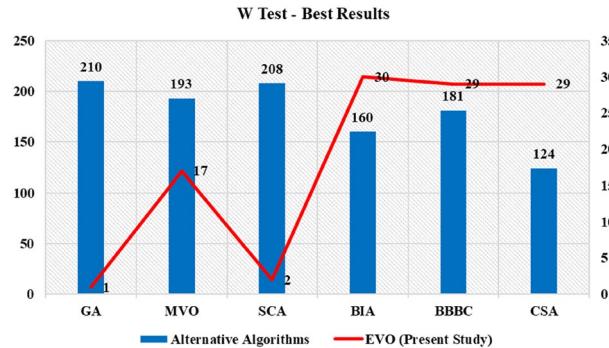
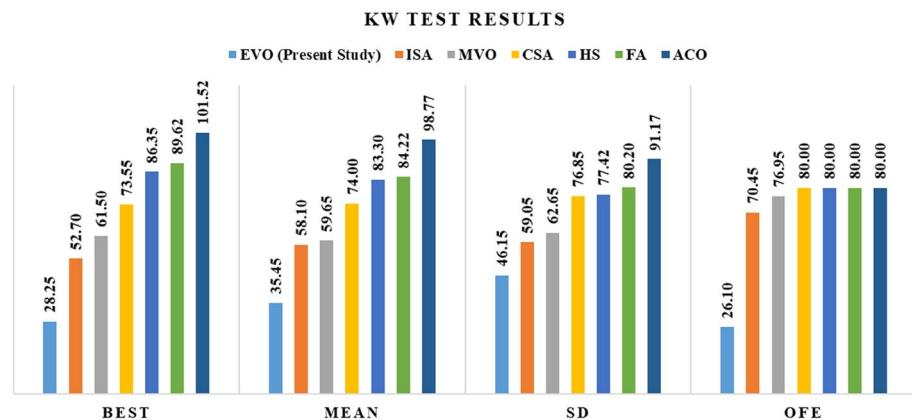


Figure 5. Convergence curves of the EVO for the mathematical functions.

**Figure 5.** (continued)

- EVO can converge to the global best solution with the lowest objective function evaluations, demonstrating this novel algorithm's efficiency from a computational point of view.
- Based on the W statistical test results, the EVO can provide better results with smaller means of ranks compared to other algorithms in a two-by-two manner.
- The results of the KW statistical test, including the mean of ranks, demonstrate that the EVO can out-ranked the other algorithms in all of the considered data sets.

Main algorithm	Data type	Alternative metaheuristic algorithms					
		ACO	FA	HS	MVO	CSA	ISA
EVO	Best	8.42E-06	8.42E-06	8.42E-06	4.15E-05	8.42E-06	7.25E-04
	Mean	1.83E-04	1.83E-04	7.25E-04	2.32E-02	2.57E-03	2.32E-02
	SD	2.32E-02	5.91E-02	5.91E-02	1.35E-01	5.91E-02	4.97E-01
	OFE	8.42E-06	8.42E-06	8.42E-06	8.42E-06	8.42E-06	4.15E-05

Table 5. The KS test results (*p*-values) for different metaheuristic algorithms.**Figure 6.** The W test results (mean of ranks) for different metaheuristic algorithms.**Figure 7.** The KW test results including mean of the ranks for different metaheuristic algorithms.

Metaheuristics	Results (sec)			
	T ₀	T ₁	\hat{T}_2	$(\hat{T}_2 - T_1)/T_0$
IMODE ⁶³	0.01117	0.2235	0.3330	0.9780
j2020 ⁶⁴	0 *	0.0465	0.1818	Inf **
GSK ⁶⁵	0.0411	1.12E-05	1.76E-05	1.58E-04
EVO (Present Study)	0.0186	0.0163	3.2271	172.6236

Table 6. The computational time of different algorithms in dealing with the CEC 2020 procedure. *It is a rounded value. ** unitless.

- Considering the constrained design examples of the CEC 2020 on real-world problems, the EVO can reach better solutions than other algorithms from the literature.
- Based on the best optimization runs conducted by different methods in dealing with the speed reducer problem, EVO can provide 2994.42, which is the best among other approaches.

No. (CEC No.)	Name	D	g	h
H ₁ (RC15)	Speed Reducer	7	11	0
H ₂ (RC25)	Hydro-Static Thrust Bearing	4	7	0
H ₃ (RC27)	Ten-Bar Truss	10	3	0
H ₄ (RC28)	Rolling Element Bearing	10	9	0

Table 7. Real-world constrained optimization problems. D, Dimensions; g, Number of inequality constraints; h, Number of equality constraints.

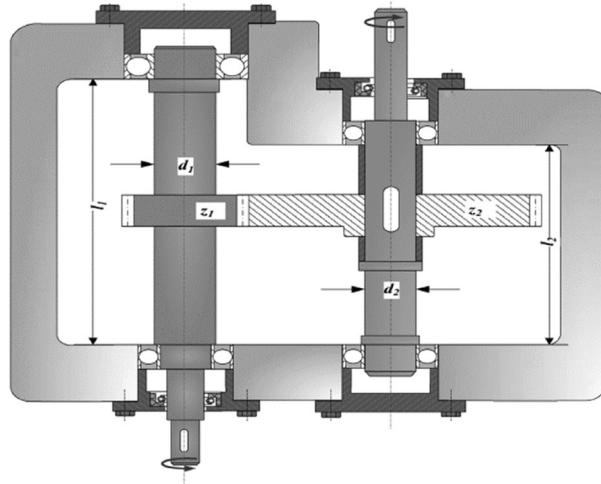


Figure 8. Schematic view of the speed reducer problem⁶⁶.

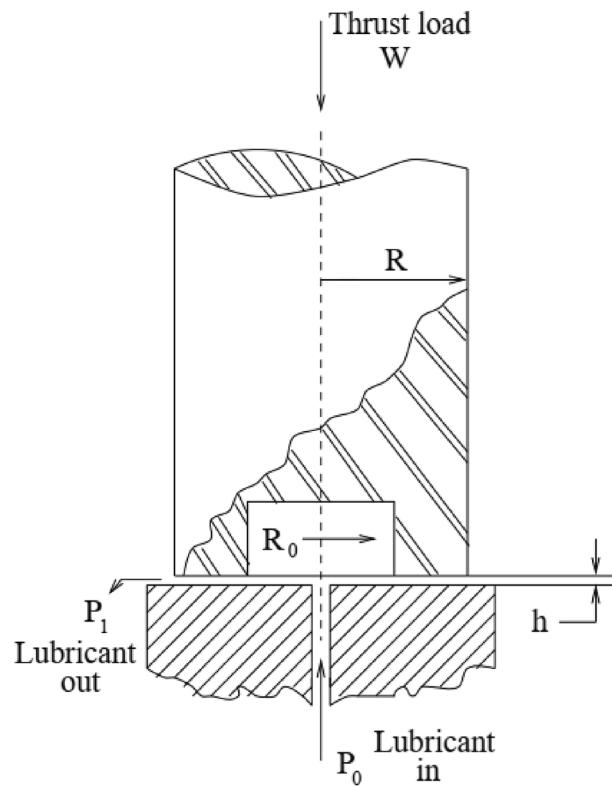


Figure 9. Schematic view of the hydro-static thrust bearing⁶⁶.

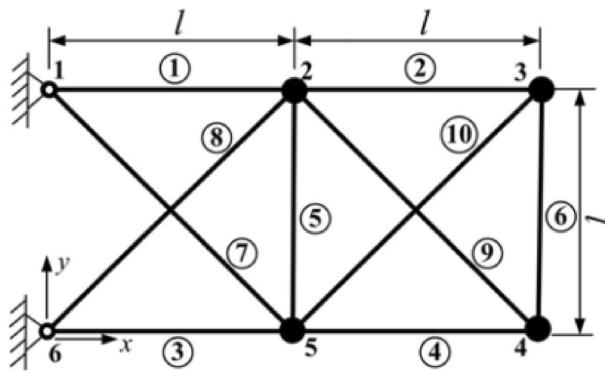


Figure 10. Schematic view of the ten-bar truss problem⁶⁶.

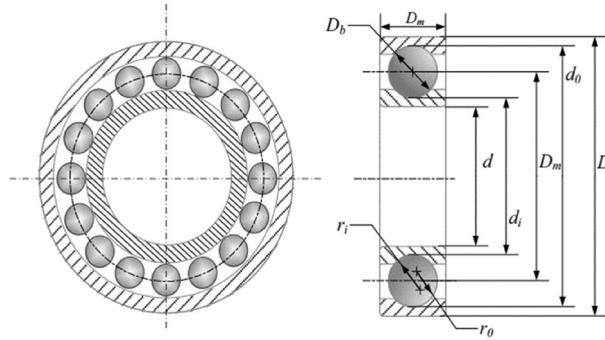


Figure 11. Schematic view of the rolling element bearing problem⁶⁶.

	ES ⁶⁷	SBS ⁶⁸	CSA ⁶⁹	DE ⁷⁰	MGA ¹⁶	CGO ⁶⁶	Present Study (EVO)
Best	3025.005000000	3008.080000000	3000.981000000	2994.471066000	2994.443649000	2994.438869000	2994.4259994948
b	3.5061630000	3.5061220000	3.5015000000	3.5000000000	3.5000066840	3.5000079560	3.500002974
m	0.7008310000	0.7000060000	0.7000000000	0.7000000000	0.7000000000	0.7000006560	0.7000000004
z	17.0000000000	17.0000000000	17.0000000000	17.0000000000	17.0000000000	17.0000008100	17.0000005194
l_1	7.4601810000	7.5491260000	7.6050000000	7.3000000000	7.3000000000	7.3005419270	7.3000041520
l_2	7.9621430000	7.8593300000	7.8181000000	7.7153199100	7.7153272500	7.7153576930	7.7153347121
d_1	3.3629000000	3.3655760000	3.3520000000	3.3502146700	3.3505953620	3.3505423910	3.3505420375
d_2	5.3090000000	5.2897730000	5.2875000000	5.2866544700	5.2866584470	5.2866579300	5.2866555457
$g_1(x)$	-0.0777000000	-0.0755000000	-0.0743000000	-0.0739152	-2.1550556750	-2.1551222770	-2.1550034019
$g_2(x)$	-0.2013000000	-0.1994000000	-0.1983000000	-0.1979985	-98.1359464800	-98.1371022200	-98.1350729737
$g_3(x)$	-0.4741000000	-0.4562000000	-0.4349000000	-0.9999967	-1.9253722120	-1.9242737610	-1.9251203297
$g_4(x)$	-0.8971000000	-0.8994000000	-0.9008000000	-0.9999995	-18.3099259200	-18.3096983400	-18.3098233913
$g_5(x)$	-0.0110000000	-0.0132000000	-0.0011000000	-0.6668526000	-0.0535900310	-0.0004371520	-0.0010647971
$g_6(x)$	-0.0125000000	-0.0017000000	-0.0004000000	-0.0000000	-0.0019196020	-0.0016664740	-0.0005189180
$g_7(x)$	-0.7022000000	-0.7025000000	-0.7025000000	-0.7025000000	-28.1000000000	-28.0999882900	-28.0999996296
$g_8(x)$	-0.0006000000	-0.0017000000	-0.0004000000	-0.0000000	-0.0000095500	-0.0000066800	-0.0000004220
$g_9(x)$	-0.5831000000	-0.5826000000	-0.5832000000	-0.5833333	-6.9999904520	-6.9999933180	-6.9999995780
$g_{10}(x)$	-0.0691000000	-0.0796000000	-0.0890000000	-0.0513257	-0.3741069580	-0.3747283410	-0.3741910957
$g_{11}(x)$	-0.0279000000	-0.0179000000	-0.0130000000	-0.0000000	-0.0000029600	-0.00000340000	-0.0000136118

Table 8. Best results of different approaches for the speed reducer problem. teeth module (m), face width (b), length of the first shaft between bearings (l_1), the diameter of the first shaft (d_1), number of teeth on pinion (z), length of the second shaft between bearings (l_2), the diameter of the second shaft (d_2). ES, Evolution strategy; SBS, Socio-behavioural simulation; CSA, Cuckoo search algorithm; DE, Differential evolution.

Approaches	Best	Mean	Worst	Std-Dev
ES ⁵⁷	3025.0050000000	3088.7778000000	3078.5918000000	NA
SBS ⁶⁸	3008.0800000000	3012.1200000000	3028.2800000000	NA
CSA ⁶⁹	3000.9810000000	3007.1997000000	3.0090000000	4.9634000000
DE ⁷⁰	2994.4710660000	2994.4710660000	2994.4710660000	0.0000000000
MGA ¹⁶	2994.4388690000	2994.4706500000	2996.5582370000	4.72E-16
CGO ⁶⁶	2994.4436490000	2994.4653970000	2995.5049330000	0.1102820000
Present Study (EVO)	2994.4259994948	2994.4438541726	2994.4633295494	0.0115141369

Table 9. Statistical results for the speed reducer problem considering different approaches.

	COM ⁷¹	CGS ⁷²	EA ⁷³	TLBO ³⁸	MGA ¹⁶	CGO ⁶⁶	Present Study (EVO)
Best	2288.2268000000	2161.4215000000	1950.2860000000	1625.4427600000	1623.9809380000	1621.2461750000	1619.5525106188
R	7.1550000000	6.7780000000	6.2710000000	5.9557805026	5.9632415160	5.9634400230	5.9567022667
R ₀	6.6890000000	6.2340000000	12.9010000000	5.3890130519	5.3959079890	5.3955878610	5.3896315105
μ	0.0000083210	6.096 E-06	0.0000056050	0.0000053586	0.0000053800	0.0000053600	0.0000053812
Q	9.1680000000	3.8090000000	2.9380000000	2.2696559728	2.2822425050	2.2648221880	2.2751206437
$g_1(x)$	-11,086.7430000000	-8329.7681000000	-2126.8673400000	-0.0001374735	-144.9586796000	-9.0788651780	-10.5323413309
$g_2(x)$	-402.4493000000	-177.3527000000	-68.0396000000	-0.0000010103	-1.1948020210	-2.5136231960	-0.2665016758
$g_3(x)$	-35.0571960000	-10.6845430000	-3.7051910000	-0.0000000210	-0.3724500270	-0.0021106440	-0.3581746345
$g_4(x)$	-0.0015420000	-0.0006520000	-0.0005590000	-0.0003243625	-0.0003291500	-0.0003248340	-0.0003275759
$g_5(x)$	-0.4660000000	-0.5440000000	-0.6660000000	-0.5667674507	-0.5673335270	-0.5678521610	-0.5670707561
$g_6(x)$	-0.0001440000	-0.0007170000	-0.0008050000	-0.0009963614	-0.0009963550	-0.0009963660	-0.0009963610
$g_7(x)$	-563.6444010000	-83.6182210000	-849.7186830000	-0.0000090762	-4.1442588760	-15.3591184600	-2.8341205598

Table 10. Comparison of the best solutions for the hydro-static thrust bearing design problem. bearing step radius (R), recess radius (R₀), oil viscosity (μ), flow rate (Q). COM Classic optimization method, CGS Combined genetic search, EA Evolutionary algorithm, TLBO Teaching–learning-based optimization.

Approaches	Best	Mean	Worst	Std-Dev
EGWO ⁷⁴	1625.4646700000	1627.7441980000	1650.6987470000	3.8155469730
JA ⁷⁵	1625.4427100000	1796.8936700000	2104.3776000000	0.2100000000
TLBO ³⁸	1625.4427600000	1797.7079800000	2096.8012000000	0.1900000000
MGA ¹⁶	1621.2461750000	1739.1567290000	1992.9613050000	0.1100000000
CGO ⁶⁶	1621.2461750000	1706.0414310000	1981.1732950000	64.4989571200
Present Study (EVO)	1619.5525106188	1730.0997954158	1899.3444719503	72.9271039971

Table 11. Statistical results of different approaches for the hydro-static thrust bearing design problem. EGWO, Enhanced grey wolf optimizer; JA, Jaya algorithm.

- EVO can provide 1619.55 for the hydro-static thrust bearing design problem, which is the best among other approaches, while the best so far found in other approaches is for CGO, which calculated 1621.24.
- The proposed EVO can provide 524.92 for the ten-bar truss design problem, which is much better than the previously reported results of 529 and 530.
- Regarding the rolling element bearing design problem, EVO can provide 81,859.74, as the best optimum solution alongside mean and worst values of 81,110.32 and 80,212.09, respectively.

Based on the results and conducted analysis, the main reason for the superiority of the EVO algorithm compared to other mentioned metaheuristics algorithms is threefold: parameter-free, fast convergence behavior, and the lowest possible objective function evaluation. The proposed EVO should be tested for future studies utilizing complex optimization problems in different fields, including real-size engineering design problems.

	UPTR ⁷⁶	SLP ⁷⁷	TLBO ⁷⁸	ECSS ⁷⁹	MGA ¹⁶	CGO ⁶⁶	Present study (EVO)
Best	544.700000000	534.570000000	530.760000000	529.250000000	529.1204229000	526.0057179000	524.9297706988
A ₁	36.380000000	35.148000000	35.494000000	39.569000000	36.764160000	35.927110000	0.0035129553
A ₂	12.941000000	13.169000000	14.777000000	16.740000000	16.298970000	13.157320000	0.0014623757
A ₃	35.764000000	37.690000000	36.203000000	34.361000000	37.943780000	34.432280000	0.0034910422
A ₄	18.314000000	19.556000000	15.387000000	12.994000000	16.510870000	15.595150000	0.0014785733
A ₅	3.002000000	1.087000000	0.645100000	0.645000000	0.659000000	0.658000000	0.0000646513
A ₆	5.433000000	4.844000000	4.589600000	4.802000000	4.574890000	4.628350000	0.0004607633
A ₇	20.989000000	18.314000000	23.211000000	26.182000000	22.940230000	21.377600000	0.0023593717
A ₈	24.140000000	27.415000000	24.561000000	21.260000000	22.631850000	27.024300000	0.0023952278
A ₉	9.753000000	12.562000000	12.482000000	11.766000000	10.878920000	13.622330000	0.0012999784
A ₁₀	18.102000000	12.106000000	12.324000000	11.392000000	11.536430000	11.000350000	0.0011962322

Table 12. Comparison of the best solutions for the ten-bar truss design problem. cross-sectional areas of the truss bars (A₁, A₂, A₃, A₄, A₅, A₆, A₇, A₈, A₉, A₁₀). IPTR, Interior Point Trust Regio; SLP, Sequential Linear Programming; ECSS, Enhanced Charged System Search.

Approaches	Best	Mean	Worst	Std-Dev
MGA ¹⁶	529.1204229000	534.6843574000	548.0179132000	26.3365167500
CGO ⁶⁶	526.0057179000	531.3864551793	534.1262390000	2.8915658990
Present Study (EVO)	524.9297706988	528.3155150968	532.0617359800	1.9129280976

Table 13. Statistical results of the MGA method for the ten-bar truss bearing design problem.

	TLBO ³⁸	ABC ⁸⁰	GWO ⁸⁰	ALO ⁸⁰	MGA ¹⁶	CGO ⁶⁶	Present Study (EVO)
Best	81,859.740000000	85,428.249500000	85,529.083000000	85,546.637700000	83,912.879830000	83,918.492530000	81,859.7415974169
D _m	21.425590000	125.659900000	125.709000000	125.718000000	125.0002787000	125.000000000	125.7190556147
D _b	125.719100000	21.408620000	21.423160000	21.425242000	21.8745119200	21.875000000	21.4255902408
Z	11.000000000	11.000000000	11.000000000	11.000000000	10.7770658300	10.7770090500	10.6955328415
f _i	0.515000000	0.515000000	0.515000000	0.515000000	0.5150008220	0.515000000	0.515000000
f ₀	0.515000000	0.515000000	0.529322000	0.5157018000	0.5150029930	0.515000000	0.515000000
K _{Dmin}	0.424266000	0.427166000	0.420867000	0.4541646000	0.4059083530	0.400000000	0.4631829367
K _{Dmax}	0.633948000	0.668849000	0.633296000	0.6464928000	0.6555880200	0.6462005260	0.6999265065
ε	0.300000000	0.300000000	0.300224000	0.3000122000	0.3000041550	0.300000000	0.300000000
e	0.0688580000	0.0713860000	0.0200000000	0.0638003000	0.0775449260	0.0501524450	0.0634315198
ζ	0.7994980000	0.6000000000	0.6194320000	0.6107592000	0.6000000000	0.6000000000	0.6042131085

Table 14. Comparison of the best solutions for the rolling element bearing design problem. pitch diameter (D_m), ball diameter (D_b), total number of balls (Z), inner raceway curvature coefficient (f_i), and the outer raceway curvature coefficient (f₀). ABC Artificial bee colony, GWO Grey wolf optimizer, ALO Ant lion optimizer.

Approaches	Best	Mean	Worst	Std-Dev
TLBO ³⁸	81,859.740000000	81,438.987000000	80,807.855100000	0.660000000
ABC ⁸⁰	85,428.249500000	85,121.754400000	83,859.085100000	362.570000000
GWO ⁸⁰	85,529.083000000	83,395.084900000	43,543.450800000	8224.500000000
ALO ⁸⁰	85,546.637700000	84,032.863600000	73,872.816400000	3121.800000000
MGA ¹⁶	83,912.879830000	83,892.256470000	83,711.213170000	23.658410000
CGO ⁶⁶	83,918.492530000	83,916.597490000	83,829.800000000	10.535800000
Present Study (EVO)	81,859.7415974169	81,110.3219736893	80,212.0966120356	522.1000790005

Table 15. Statistical results of different approaches for the rolling element bearing design problem.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 1 July 2022; Accepted: 30 December 2022

Published online: 05 January 2023

References

- Boussaïd, I., Lepagnot, J. & Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **237**, 82–117. <https://doi.org/10.1016/j.ins.2013.02.041> (2013).
- Hussain, K., Salleh, M. N. M., Cheng, S. & Shi, Y. Metaheuristic research: a comprehensive survey. *Artif. Intell. Rev.* **52**(4), 2191–2233. <https://doi.org/10.1007/s10462-017-9605-z> (2019).
- Holland, J. H. Genetic algorithms and adaptation. In *Adaptive Control of Ill-Defined Systems* (eds Selfridge, O. G. *et al.*) 317–333 (Springer US, 1984). https://doi.org/10.1007/978-1-4684-8941-5_21.
- Storn, R. & Price, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**, 341–359. <https://doi.org/10.1023/A:1008202821328> (1997).
- Karami, H., Sanjari, M. J. & Gharehpetian, G. B. Hyper-Spherical Search (HSS) algorithm: a novel meta-heuristic algorithm to optimize nonlinear functions. *Neural Comput. Appl.* **25**, 1455–1465. <https://doi.org/10.1007/s00521-014-1636-7> (2014).
- Eberhart, R. & Kennedy, J. in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. 39–43.
- Yang, X.-S. Nature-inspired mateheuristic algorithms: success and new challenges. *J. Comput. Eng. Inform. Technol.* <https://doi.org/10.4172/2324-9307.1000e101> (2012).
- Dorigo, M., Maniezzo, V. & Colorni, A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)* **26**(1), 29–41. <https://doi.org/10.1109/3477.484436> (1996).
- Ahmed, Z. E., Saeed, R. A., Mukherjee, A. & Ghorpade, S. N. Energy optimization in low-power wide area networks by using heuristic techniques. In *LPWAN Technologies for IoT and M2M Applications* 199–223 (Elsevier, 2020). <https://doi.org/10.1016/B978-0-12-818880-4.00011-9>.
- Dhiman, G., Garg, M., Nagar, A., Kumar, V. & Dehghani, M. A novel algorithm for global optimization: rat swarm optimizer. *J. Ambient. Intell. Humaniz. Comput.* **12**, 8457–8482. <https://doi.org/10.1007/s12652-020-02580-0> (2021).
- Yampolskiy, R. V., Ashby, L. & Hassan, L. Wisdom of artificial crowds—A metaheuristic algorithm for optimization. *J. Intell. Learn. Syst. Appl.* **4**(2), 10. <https://doi.org/10.4236/jilsa.2012.42009> (2012).
- Xie, L. *et al.* Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization. *Comput. Intell. Neurosci.* **2021**, 9210050. <https://doi.org/10.1155/2021/9210050> (2021).
- Karaboga, D. & Basturk, B. in *Foundations of Fuzzy Logic and Soft Computing*. (eds Patricia Melin *et al.*) 789–798 (Springer Berlin Heidelberg).
- Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. & Mirjalili, S. Henry gas solubility optimization: a novel physics-based algorithm. *Futur. Gener. Comput. Syst.* **101**, 646–667. <https://doi.org/10.1016/j.future.2019.07.015> (2019).
- Azizi, M. Atomic orbital search: a novel metaheuristic algorithm. *Appl. Math. Model.* **93**, 657–683. <https://doi.org/10.1016/j.apm.2020.12.021> (2021).
- Talatahari, S., Azizi, M. & Gandomi, A. H. Material generation algorithm: a novel metaheuristic algorithm for optimization of engineering problems. *Processes* **9**, 859 (2021).
- Azizi, M., Shishehgarkhaneh, M. B. & Basiri, M. Optimum design of truss structures by Material Generation Algorithm with discrete variables. *Decis. Anal.* **3**, 100043. <https://doi.org/10.1016/j.dajour.2022.100043> (2022).
- Hosseini, E., Ghafoor, K. Z., Emrouznejad, A., Sadiq, A. S. & Rawat, D. B. Novel metaheuristic based on multiverse theory for optimization problems in emerging systems. *Appl. Intell.* **51**, 3275–3292. <https://doi.org/10.1007/s10489-020-01920-z> (2021).
- Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S. & Al-Atabany, W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **51**, 1531–1551. <https://doi.org/10.1007/s10489-020-01893-z> (2021).
- Pereira, J. L. J. *et al.* Lichtenberg algorithm: a novel hybrid physics-based meta-heuristic for global optimization. *Expert Syst. Appl.* **170**, 114522. <https://doi.org/10.1016/j.eswa.2020.114522> (2021).
- Kaveh, A. & Dadras, A. A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Adv. Eng. Softw.* **110**, 69–84. <https://doi.org/10.1016/j.advengsoft.2017.03.014> (2017).
- Heidari, A. A. *et al.* Harris hawks optimization: algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872. <https://doi.org/10.1016/j.future.2019.02.028> (2019).
- Alabool, H. M., Alarabiati, D., Abualigah, L. & Heidari, A. A. Harris hawks optimization: a comprehensive review of recent variants and applications. *Neural Comput. Appl.* **33**, 8939–8980. <https://doi.org/10.1007/s00521-021-05720-5> (2021).
- Chou, J.-S. & Truong, D.-N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **389**, 125535. <https://doi.org/10.1016/j.amc.2020.125535> (2021).
- Zhang, J., Xiao, M., Gao, L. & Pan, Q. Queuing search algorithm: a novel metaheuristic algorithm for solving engineering optimization problems. *Appl. Math. Model.* **63**, 464–490. <https://doi.org/10.1016/j.apm.2018.06.036> (2018).
- Feng, Z.-K., Niu, W.-J. & Liu, S. Cooperation search algorithm: a novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. *Appl. Soft Comput.* **98**, 106734. <https://doi.org/10.1016/j.asoc.2020.106734> (2021).
- Abualigah, L. *et al.* Aquila optimizer: a novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **157**, 107250. <https://doi.org/10.1016/j.cie.2021.107250> (2021).
- Braik, M., Sheta, A. & Al-Hiary, H. A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. *Neural Comput. Appl.* **33**, 2515–2547. <https://doi.org/10.1007/s00521-020-05145-6> (2021).
- Tarkhaneh, O., Alipour, N., Chapnevis, A. & Shen, H. Golden tortoise beetle optimizer: a novel nature-inspired meta-heuristic algorithm for engineering problems. (2021).
- Rahkar Farshi, T. Battle royale optimization algorithm. *Neural Comput. Appl.* **33**, 1139–1157. <https://doi.org/10.1007/s00521-020-05004-4> (2021).
- Savsanı, P. & Savsanı, V. Passing vehicle search (PVS): a novel metaheuristic algorithm. *Appl. Math. Model.* **40**, 3951–3978. <https://doi.org/10.1016/j.apm.2015.10.040> (2016).
- Topal, A. O. & Altun, O. A novel meta-heuristic algorithm: dynamic virtual bats algorithm. *Inf. Sci.* **354**, 222–235. <https://doi.org/10.1016/j.ins.2016.03.025> (2016).
- Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Comput. Struct.* **169**, 1–12. <https://doi.org/10.1016/j.compstruc.2016.03.001> (2016).
- Liang, Y.-C. & Juarez, J. R. C. A novel metaheuristic for continuous optimization problems: virus optimization algorithm. *Eng. Optim.* **48**(1), 73–93. <https://doi.org/10.1080/0305215X.2014.994868> (2016).

35. Alsattar, H. A., Zaidan, A. A. & Zaidan, B. B. Novel meta-heuristic bald eagle search optimisation algorithm. *Artif. Intell. Rev.* **53**, 2237–2264. <https://doi.org/10.1007/s10462-019-09732-5> (2020).
36. Rashedi, E., Nezamabadi-pour, H. & Saryazdi, S. G. S. A. A gravitational search algorithm. *Inform. Sci.* **179**, 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004> (2009).
37. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007> (2014).
38. Rao, R. V., Savsani, V. J. & Vakharia, D. P. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **43**, 303–315. <https://doi.org/10.1016/j.cad.2010.12.015> (2011).
39. Azizi, M., Talatahari, S. & Gandomi, A. H. Fire Hawk optimizer: a novel metaheuristic algorithm. *Artif. Intell. Rev.* <https://doi.org/10.1007/s10462-022-10173-w> (2022).
40. Shishehgarkhaneh, M. B., Azizi, M., Basiri, M. & Moehler, R. C. BIM-based resource tradeoff in project scheduling using fire hawk optimizer (FHO). *Buildings* **12**, 1472 (2022).
41. Luque-Chang, A., Cuevas, E., Fausto, F., Zaldívar, D. & Pérez, M. Social spider optimization algorithm: modifications, applications, and perspectives. *Math. Probl. Eng.* **2018**, 6843923. <https://doi.org/10.1155/2018/6843923> (2018).
42. Husseinzadeh Kashan, A. League Championship algorithm (LCA): an algorithm for global optimization inspired by sport championships. *Appl. Soft Comput.* **16**, 171–200. <https://doi.org/10.1016/j.asoc.2013.12.005> (2014).
43. Talatahari, S. & Azizi, M. Chaos game optimization: a novel metaheuristic algorithm. *Artif. Intell. Rev.* **54**, 917–1004. <https://doi.org/10.1007/s10462-020-09867-w> (2021).
44. Azizi, M., Aickelin, U., Khorshidi, H. A. & Shishehgarkhaneh, M. B. Shape and size optimization of truss structures by Chaos game optimization considering frequency constraints. *J. Adv. Res.* <https://doi.org/10.1016/j.jare.2022.01.002> (2022).
45. Yang, X. S. & Hosseini Gandomi, A. Bat algorithm: a novel approach for global engineering optimization. *Eng. Comput.* **29**, 464–483 (2012).
46. Ghasemi-Marzbali, A. A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm. *Soft. Comput.* **24**, 13003–13035. <https://doi.org/10.1007/s00500-020-04721-1> (2020).
47. Hayyolalam, V. & Pourhaji Kazem, A. A. Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **87**, 103249. <https://doi.org/10.1016/j.engappai.2019.103249> (2020).
48. Kumar, N., Singh, N. & Vidyarthi, D. P. Artificial lizard search optimization (ALSO): a novel nature-inspired meta-heuristic algorithm. *Soft. Comput.* **25**, 6179–6201. <https://doi.org/10.1007/s00500-021-05606-7> (2021).
49. Anita, A. Y. & Kumar, N. Artificial electric field algorithm for engineering optimization problems. *Expert Syst. Appl.* **149**, 113308. <https://doi.org/10.1016/j.eswa.2020.113308> (2020).
50. Vatin, N., Ivanov, A. Y., Rutman, Y. L., Chernogorskiy, S. & Shvetsov, K. Earthquake engineering optimization of structures by economic criterion. *Mag. Civil Eng.* **76**, 67–83. <https://doi.org/10.18720/MCE.76.7> (2017).
51. Ouaarab, A., Ahiod, B. & Yang, X.-S. Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Comput. Appl.* **24**, 1659–1669. <https://doi.org/10.1007/s00521-013-1402-2> (2014).
52. Gupta, S., Deep, K., Moayed, H., Foong, L. K. & Assad, A. Sine cosine grey wolf optimizer to solve engineering design problems. *Eng. Comput.* **37**, 3123–3149. <https://doi.org/10.1007/s00366-020-00996-y> (2021).
53. Rao, R. V. & Pawar, R. B. Self-adaptive multi-population rao algorithms for engineering design optimization. *Appl. Artif. Intell.* **34**, 187–250. <https://doi.org/10.1080/08839514.2020.1712789> (2020).
54. Kamboj, V., Nandi, A., Bhaduria, A. & Sehgal, S. An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Appl. Soft Comput.* **89**, 106018. <https://doi.org/10.1016/j.asoc.2019.106018> (2019).
55. Qi, X., Yuan, Z. & Song, Y. A hybrid pathfinder optimizer for unconstrained and constrained optimization problems. *Comput. Intell. Neurosci.* **2020**, 5787642. <https://doi.org/10.1155/2020/5787642> (2020).
56. Azizi, M., Ghasemi Seyyed Arash, M., Ejlali Reza, G. & Talatahari, S. Optimization of fuzzy controller for nonlinear buildings with improved charged system search. *Struct. Eng. Mech.* **76**, 781–797 (2020).
57. Alekseytsev, A. Metaheuristic optimization of building structures with different level of safety. *J. Phys. Conf. Ser.* **1425**, 012014. <https://doi.org/10.1088/1742-6596/1425/1/012014> (2019).
58. Khondoker, M. T. H. Automated reinforcement trim waste optimization in RC frame structures using building information modeling and mixed-integer linear programming. *Autom. Construct.* **124**, 103599. <https://doi.org/10.1016/j.autcon.2021.103599> (2021).
59. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82 (1997).
60. Yue, C. T. *et al.* *Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*. Technical Report (Nanyang Technological University, Singapore, 2020).
61. Kumar, A. *et al.* A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evolut. Comput.* **56**, 100693. <https://doi.org/10.1016/j.swevo.2020.100693> (2020).
62. Silberberg, M. *Principles of General Chemistry* 3rd edn. (McGraw-Hill Education, 2012).
63. Sallam, K. M., Elsayed, S. M., Chakrabortty, R. K. & Ryan, M. J. in *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8.
64. Brest, J., Maučec, M. S. & Bošković, B. in *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8.
65. Mohamed, A. W., Hadi, A. A., Mohamed, A. K. & Awad, N. H. in *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8.
66. Talatahari, S. & Azizi, M. Optimization of constrained mathematical and engineering design problems using chaos game optimization. *Comput. Ind. Eng.* **145**, 106560 (2020).
67. Mezura-Montes, E., Coello, C. & Landa-Becerra, R. Engineering optimization using simple evolutionary algorithm. (2003).
68. Akhtar, S., Tai, K. & Ray, T. A socio-behavioural simulation model for engineering design optimization. *Eng. Optim.* **34**, 341–354. <https://doi.org/10.1080/03052150212723> (2002).
69. Gandomi, A. H., Yang, X.-S. & Alavi, A. H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **29**, 17–35 (2013).
70. Zhang, M., Luo, W. & Wang, X. Differential evolution with dynamic stochastic selection for constrained optimization. *Inf. Sci.* **178**, 3043–3074. <https://doi.org/10.1016/j.ins.2008.02.014> (2008).
71. Siddall, J. N. *Optimal Engineering Design: Principles and Applications* (CRC Press, 1982).
72. Deb, K. & Goyal, M. Optimizing engineering designs using a combined genetic search. In *Proc. International Conference on Genetic Algorithms*. 521–528 (1997).
73. Hernandez-Aguirre, A., Botello, S., Coello, C. & Lizárraga, G. Use of multiobjective optimization concepts to handle constraints in single-objective optimization. In *Genetic and Evolutionary Computation — GECCO 2003: Genetic and Evolutionary Computation Conference Chicago, IL, USA, July 12–16, 2003 Proceedings, Part I* (eds Cantú-Paz, E. *et al.*) 573–584 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003). https://doi.org/10.1007/3-540-45105-6_69.
74. Şahin, I., Dörterler, M. & Gokce, H. Optimization of hydrostatic thrust bearing using enhanced grey wolf optimizer. *Mechanika* **25**, 480–486 (2019).
75. Rao, R. V. & Waghmare, G. G. A new optimization algorithm for solving complex constrained design optimization problems. *Eng. Optim.* **49**, 60–83. <https://doi.org/10.1080/0305215X.2016.1164855> (2017).
76. Yu, Z. *et al.* Optimal design of truss structures with frequency constraints using interior point trust region method. *Proc. Rom. Acad. - Math. Phys. Tech. Sci. Inf. Sci.* **15**(2), 165–173 (2014).

77. Lamberti, L. & Pappalettere, C. Move limits definition in structural optimization with sequential linear programming. Part I: optimization algorithm. *Comput. Struct.* **81**, 197–213. [https://doi.org/10.1016/S0045-7949\(02\)00442-X](https://doi.org/10.1016/S0045-7949(02)00442-X) (2003).
78. Baghlan, A. & Makiabadi, M. H. Teaching-learning-based optimization algorithm for shape and size optimization of truss structures with dynamic frequency constraints. *Iran. J. Sci. Technol. Trans. A Sci.* **37**, 409–421 (2013).
79. Kaveh, A. & Zolghadr, A. Shape and size optimization of truss structures with frequency constraints using enhanced charged system search algorithm. *Asian J. Civil Eng. (Build. Hous.)* **12**, (2011).
80. Yildiz, A. R., Abderazeck, H. & Mirjalili, S. A comparative study of recent non-traditional methods for mechanical design optimization. *Arch. Comput. Methods Eng.* **27**, 1031–1048 (2020).

Author contributions

M.A. contributed to the study design, developed the methods, implemented the experiments, wrote and revised the article. U.A., H.A.K. and M.B.S. provided critical review on the outcomes, revised the article. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.A.K.

Reprints and permissions information is available at www.nature.com/reprints.

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023