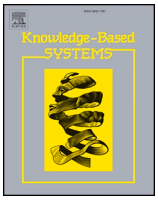
Knowledge-Based Systems 295 (2024) 111850

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Optimization based on the smart behavior of plants with its engineering applications: Ivy algorithm

Mojtaba Ghasemi [a], Mohsen Zare [b], Pavel Trojovský [c,*], Ravipudi Venkata Rao [d], Eva Trojovská [c], Venkatachalam Kandasamy [c]

[a] Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz, Iran
[b] Department of Electrical Engineering, Faculty of Engineering, Jahrom University, Jahrom, Iran
[c] Department of Mathematics, Faculty of Science, University of Hradec Králové, Rokitanského 62, Hradec Králové, Czech Republic
[d] Department of Mechanical Engineering, Sardar Vallabhbhai National Institute of Technology, Ichchanath, Surat, Gujarat, India

ARTICLE INFO

Keywords:
Global optimization
Nature-inspired algorithm
Engineering design
Meta-heuristic algorithms
Ivy algorithm

ABSTRACT

This study presents a powerful and new modeling variant of bio-inspired algorithms, namely, the Ivy algorithm (IVYA), drawn from the growth patterns of Ivy plants. The algorithm simulates the coordinated and ordered population growth and the spreading and evolution of Ivy plants. The growth rate of Ivy plants is modeled using a differential equation and a data-intensive experimental process. The algorithm utilizes the knowledge of nearby Ivy plants to determine the direction of growth. Additionally, the algorithm mimics the behavior of Ivy plants in nature by choosing the closest and most vital neighbor for self-improvement. The IVYA's unique characteristics of preserving population diversity and its simplicity and flexibility allow for easy modification and extension, thus enabling researchers and practitioners to explore various modifications and techniques to enhance its performance and capabilities. These are the basic needs in optimizing engineering problems. The IVYA is compared with ten other algorithms on 26 classical test functions, demonstrating superior performance. Furthermore, the effectiveness of IVYA is shown by solving 12 engineering optimization problems and comparing the results with various optimization algorithms. The experimental results highlight the efficacy and competitiveness of the IVYA in solving optimization problems. Consider that the source code of the IVYA is publicly available at https://www.optim-app.com/projects/ivya.

1. Introduction

Over the past twenty years, there has been an increasing demand to optimize a multitude of control variables while facing diverse constraints in real-world optimization problems [1]. Engineering optimization with various objectives following various constraints and including various complexities, like non-differentiability, cubic, non-linearity, mixed-integers, etc., were optimized using mathematical programming techniques [2], such as the trust-region quadratic-based models [3], Nelder and Mead simplex-based method [4], and also met heuristics, part of the optimization procedure, which is the act of determining the optimum approach for either minimizing or maximizing the objectives of a given problem [5,6]. Of course, along with the many meta-innovation algorithms introduced in recent years, various methods have been successfully introduced and implemented to improve these meta-heuristics, such as the natural survivor method (NSM) [7].

1.1. An introduction to recently proposed algorithms

Nevertheless, because of the enormous magnitude of engineering optimization issues, traditional mathematical programming methods seldom, if ever, offer suitable solutions for many optimization problems [8]. The second set of suggested strategies represents the meta-heuristic optimization algorithms, drawing insights from various natural occurrences like the collective behaviors of birds and other animals in nature. As shown below [9], metaheuristics are typically categorized into four distinct groups:

- **Swarm Intelligence Algorithms (SIAs):** Physical behaviors and patterns seen in nature make the basis of SIAs—for instance, a new optimizer based on butterflies' foraging and mating activities [10].

* Corresponding author: Rokitanského 62, Hradec Králové, 500 03, Czech Republic.
E-mail address: pavel.trojovsky@uhk.cz (P. Trojovský).

https://doi.org/10.1016/j.knosys.2024.111850
Received 18 December 2023; Received in revised form 29 March 2024; Accepted 21 April 2024
Available online 23 April 2024

Golden jackal optimization (GJO) [11] mimics golden jackals' social and friendly hunting treatment. Wild geese algorithm (WGA) draws its principles from the life cycle and behaviors observed in wild geese within their natural habitats [12]. Another algorithm, named the orca predation algorithm (OPA) [13], adopts the hunting behavior of orcas, utilizing strategies such as attacking, driving, and encircling. Aphid–ant mutualism (AAM) algorithm [14] mimics a unique relationship (mutualism) that can be seen between some species of aphids and ants. Barnacle mating habits are the basis of the BMO [15]. The remora optimization algorithm (ROA) [16] shows the parasitic behavior observed in remora fish. The gannet optimization algorithm (GOA) [17] is another algorithm that mathematically expresses the unique behavior of gannets when foraging and has been used to provide exploitation and exploration. Horse herd optimization algorithm (HOA) uses six essential features: roam, defense mechanism, imitation, sociability, hierarchy, and grazing [18]. Integrated optimization algorithm (IOA) [19] combines several operations: role learning, crossover search, wanderer search, leader search, and follower search. Parasitism–predation algorithm (PPA) was developed to tackle the challenges of low convergence and high dimensionality in extensive datasets [20]. Emperor penguin optimizer (EPO) replicates the communal behavior observed in a huddle of penguins [21]. The dwarf mongoose optimization algorithm (DMO) [22] is based on the foraging treatment of dwarf mongoose. Optimizer JAYA is a gradient-free search algorithm [23]. The colony predation algorithm (CPA) [24] draws inspiration from the synchronized hunting behavior observed in natural creatures—the crow search algorithm (CSA) through crows' treatment of hiding food sources [25]. Sailfish optimizer (SO) [26] replicates the coordinated hunting behavior of sailfish in a group setting. Other similar algorithms include, e.g. artificial ecosystem-based optimization (AEO) [27], yellow saddle goatfish algorithm [28], and fitness dependent optimizer (FDO) [29]. The following is a list of other types of inspired algorithms based on the SIAs, summarized in Table 1.

- **Human/social-related Algorithms (HSAs):** Various human behaviors inspire these algorithms [73]. They mimic humans' decision-making strategies, navigating challenges, playing games, and adapting to changing environments. For example, a political algorithm (PO) that mimics the person's political behavior [74], transit search (TS) [75] relies on an established exoplanet exploration method: observing starlight at intervals to detect changes in luminosity, indicating the passage of a planet in front of the star. Queuing search algorithm (QSA) mimics human queuing actions [28]. The volleyball premier league algorithm (VPLA) is based on interactions and competition between volleyball teams [76] and the future search algorithm (FSA) that imitates human behavior in real life [77]. The following is a list of other inspired algorithms based on the HSAs, summarized in Table 2.

- **Optimization Algorithms based on different Theories in different Sciences (TSAs), such as mathematics and physics**

  Fundamental principles and occurrences inspire these algorithms in the different theories in different sciences, such as mathematics and physics. Among them is, for example, fitness-distance balance (FDB) [93], which aims to tackle premature convergence in MHS. It effectively identifies solution candidates with the highest potential for improving the search process from the population. This new method, with its different types like fitness-distance-constraint (FDC) [94], dynamic fitness-distance balance (dFDB) [95], and adaptive fitness-distance balance (AFDB) [96], has been successfully implemented in several optimization algorithms to optimize different problems like the accurate photovoltaic modeling [97], the solar photovoltaic parameter estimation [98], and optimal power flow [99]. heat transfer search (HTS) by the formwork of heat transfer and thermodynamics [100], thermal exchange optimization (TEO) by the

Table 1
Some new optimization algorithms are based on the SIAs used in the recent literature.

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
[30]	Mine blast algorithm (MBA)	MBA draws inspiration from real-world mine bomb explosions. This innovative population-based approach solves engineering design and constrained optimization problems.	2013
[31]	Competitive swarm optimizer (CSO)	The algorithm is inspired by particle swarm optimization (PSO) [32] but takes a different conceptual approach. In CSO, particle updates are not based on their individual or overall best positions. Instead, it introduces a pairwise competition mechanism. A particle that loses the competition adjusts its position based on the behavior of the winning particle.	2015
[33]	Moth-flame optimization (MFO)	The core idea behind MFO is inspired by the navigation strategy of moths known as transverse orientation. This strategy involves moths maintaining a steady angle relative to the moon during flight, enabling them to travel efficiently over long distances in a straight path. However, despite this ability, moths often get trapped in unproductive or dangerous spirals around artificial light sources.	2015
[34]	Pity beetle algorithm (PBA)	PBA is inspired by the collective behavior of the Pityogenes chalcographus beetle, which is adept at finding weakened trees in forests for nesting and food. However, if its population exceeds a certain threshold, it may also attack healthy trees.	2018
[35]	Harris hawks optimization (HHO)	HHO draws inspiration from the cooperative hunting behavior of Harris's hawks, known as the "surprise pounce." These hawks work together to ambush prey from different angles, adapting to dynamic scenarios and prey evasion tactics. This research aims to mimic these behaviors to create an effective mathematical optimization method.	2019
[36]	Squirrel search algorithm (SSA)	SSA is inspired by the dynamic foraging behavior observed in southern flying squirrels and their highly efficient method of locomotion, known as gliding. Gliding is a remarkably efficient mode of travel for these diminutive mammals over extended distances.	2019
[37]	Phasor particle swarm optimization (PPSO)	PPSO derives its foundation from emulating particle control parameters through a phase angle (θ), drawing inspiration from phasor theory within mathematics. Integrating a phase angle (θ) introduces a transformative element to the PSO algorithm, imbuing it with self-adjusting capabilities, trigonometric principles, equilibrium, and a nonparametric meta-heuristic essence.	2019
[38]	Slime mould algorithm (SMA)	SMA draws inspiration from the oscillation pattern observed in slime molds. It introduces a unique mathematical model with adaptive weights to mimic the generation of both positive and negative feedback in slime mold propagation waves.	2020

(*continued on next page*)

Table 1 (*continued*)

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
		This model helps construct optimal paths for connecting food sources, showcasing remarkable exploratory capabilities and exploitation tendencies.	
[39]	Marine predators algorithm (MPA)	MPA is inspired by the foraging tactics of ocean predators, including Brownian and Lévy movements, and the optimal encounter rate strategy between predator and prey. It closely follows natural rules governing foraging strategies and encounter rates in marine ecosystems, capturing biological interactions in its algorithm.	2020
[40]	Tunicate swarm algorithm	TSA emulates the jet propulsion and swarm dynamics observed in tunicates as they navigate and forage for sustenance.	2020
[41]	Rain optimization algorithm (ROA)	ROA is an innovative metaheuristic algorithm inspired by the behavior of raindrops, which gravitate towards local minima upon reaching the ground. With appropriately tuned parameters, this algorithm can discover both global and local extrema.	2020
[42]	Red fox optimization (RFO)	RFO takes inspiration from a mathematical model depicting the behavioral patterns of red foxes, including their search for food, hunting strategies, population growth dynamics, and evasion tactics against hunters. This model integrates local and global optimization methods and a reproduction mechanism to simulate the complex dynamics of red fox populations.	2021
[43]	Starling murmuration optimizer (SMO)	SMO introduced a dynamic multi-flock assembly and three new exploration tactics: separation, dive, and swirl. Separation enriches population diversity and avoids local peaks using a quantum-inspired mechanism. Diving explores the search space with a fresh quantum-random dive, while Swirl capitalizes on promising regions with a new cohesion force. SMO balances exploration and exploitation by choosing between diving and swirling based on flock quality.	2022
[44]	Flying foxes optimization (FFO)	FFO draws inspiration from the survival tactics employed by flying foxes during heatwaves. It harnesses a Fuzzy Logic (FL) approach to autonomously adjust parameters for each solution, rendering it a parameter-free optimization algorithm.	2022
[45]	Beluga whale optimization (BWO)	BWO consists of three phases: exploration, exploitation, and whale fall, reflecting pair swimming, prey pursuit, and the use of whale carcasses, respectively. The equilibrium coefficient and the likelihood of encountering a whale fall are autonomously adjustable, influencing the balance between exploration and exploitation. Additionally, incorporating Levy flights enhances global convergence during exploitation.	2022
[46]	Wild horse optimizer (WHO)	WHO is inspired by the social behaviors of wild horses, who form herds consisting of a stallion, multiple mares, and their offspring. These horses engage in various activities such as grazing, asserting dominance, and mating. A unique behavior observed in horses is their sense of decency, where foals leave the herd before reaching sexual maturity to prevent incestuous mating. WHO's central concept is derived from this notion of decency observed in horses.	2022
[47]	Honey badger algorithm (HBA)	HBA draws inspiration from the clever foraging habits of honey badgers, aiming to devise an efficient search strategy for solving optimization problems. The honey badger's dynamic approach to searching, involving digging and honey finding, is translated into exploration and exploitation phases within HBA. Additionally, through controlled randomization methods, HBA ensures a diverse population throughout the search process, even as it nears completion.	2022
[48]	Sparrow search algorithm (SSA)	SSA was developed to emulate the anti-predation and foraging behaviors exhibited by sparrows.	2022
[49]	Pufferfish optimization algorithm (POA)	This innovative optimization method draws inspiration from the circular structures found on the ocean floor, created by the pufferfish species Torquigener albomaculosus. The fundamental parameters of this method were derived from observations of the pufferfish's circular structure formation process. The intricate details of this natural phenomenon are thoroughly explored in this study.	2022
[50]	Predator-prey optimization (PPO)	Prey-predator interactions inspire PPO in nature. It features two populations: predators and prey. Energy acquisition depends on body mass and interactions between predators and their common prey. The most successful predator initiates a local search to exploit the environment.	2022
[51]	Dandelion optimizer (DO)	DO simulates dandelion seed flight by wind in three stages: landing, descending, and rising. During rising, seeds spiral up or drift locally. In descending, they steadily adjust direction. In landing, seeds settle randomly to grow, with movement described by Brownian motion and a Levy random walk.	2022
[52]	Artificial rabbits optimization (ARO)	The ARO algorithm draws inspiration from rabbits' survival tactics, specifically their detour foraging and random hiding behaviors. Detour foraging involves rabbits feeding near other nests to avoid predators, while random hiding allows them to select burrows randomly for safety. As energy levels decline, rabbits shift from foraging to hiding to conserve energy.	2022
[53]	White shark optimizer (WSO)	WSO is inspired by the sensory abilities of great white sharks during navigation and foraging. These traits are mathematically modeled to balance exploration and exploitation,	2022

(*continued on next page*)

Table 1 (*continued*)

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
		enabling effective search space optimization.	
[54]	Giant trevally optimizer (GTO)	The GTO algorithm is inspired by the hunting behavior of giant trevally, mainly when targeting seabirds like sooty terns. It models three key steps: simulating trevally movement, selecting optimal hunting areas, and executing the chase and attack, including leaping out of the water to capture prey.	2022
[55]	Nutcracker optimization algorithm (NOA)	NOA is inspired by Clark's nutcrackers, exhibiting two distinct behaviors: seed searching and storage in caches during summer and fall and spatial memory-based cache retrieval using markers in winter and spring. Random exploration of the search space is employed if seeds are not found.	2023
[56]	Genghis khan shark optimizer (GKSO)	The predatory and survival behaviors of the Genghis Khan shark inspire GKSO. It simulates four key activities: hunting (exploration), movement (exploitation), foraging (transition from exploration to exploitation), and self-protection. Mathematical models emulate these activities to optimize agents across different search space regions.	2023
[57]	Dynamic hunting leadership (DHL)	DHL is inspired by nature and offers innovative heuristic techniques for optimization problems. It comprises four variants with distinct leadership strategies, dynamically adjusting the number of leaders for enhanced performance. The evaluation focuses on stability in exploring the search space and exploitation phases, discussing exploration versus exploitation advantages across DHL variants.	2023
[58]	Pilgrimage walk optimization (PWO)	PWO mimics Taiwan's Matsu bobee custom. It simulates devotees' collective movement during Matsu's palanquin procession. It reflects various folk belief activities like the return palanquin ceremony, palanquin robbing, crawling beneath the palanquin, leisure ceremonies, pilgrimage, and venation block casting.	2023
[59]	Leopard seal optimization (LSO)	The hunting strategy of leopard seals inspires LSO. LSO is designed to be a flexible swarm intelligence algorithm capable of solving real-time engineering problems quickly and accurately while avoiding local optima.	2023
[60]	Snow avalanches algorithm (SAA)	SAA has four phases: avalanche, human factors, regional weather, and normal conditions, utilizing only one control parameter. Its key benefits include simplicity, minimal parameters, and easy implementation.	2023
[61]	Humboldt squid optimization algorithm (HSOA)	Humboldt squids' behavior inspires HSOA in hunting, moving, and mating. It involves stages like attacking fish schools, escaping, successful attacks, interspecies attacks, and mating. The algorithm comprises half Humboldt squids and half school fish, promoting cooperation for optimal outcomes.	2023

Table 1 (*continued*)

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
[62]	Sea-horse optimizer (SHO)	SHO is inspired by sea horses' natural behaviors, including movement, predation, and breeding. It replicates sea horse movement patterns and their predation mechanism and incorporates a breeding stage mimicking male pregnancy. These behaviors are mathematically formulated to balance local exploitation and global exploration.	2023
[63]	Termite alate optimization (TAO)	The phototactic behavior of termite alates inspires TAOA. It offers faster convergence with effective exploration and exploitation capabilities, a moderate number of parameters, and computational complexity.	2023
[64]	Goat search algorithms (GSAs)	This article [64] presents an evolutionary algorithm (EA) inspired by the social behavior of village goats. It stimulates the process of a shepherd searching for a lost goat, considering factors like false walks, uniform, and non-uniform steps, goat jumps, periodic walks, and goodness of fit for different walking functions.	2023
[65]	Crested porcupine optimizer (CPO)	CPO is inspired by the defensive behaviors of crested porcupines (CP) and aims to optimize various large-scale problems. CP employs four protective mechanisms, ranging from minor to most aggressive: sight, sound, odor, and physical attack. CPO's exploration behavior mirrors the first two defenses (sight and sound), while its exploitation behavior is reflected in the last two defenses (odor and physical attack).	2024
[66]	Triangulation topology aggregation optimizer (TTAO)	TTAO is based on triangular structures in mathematics and employs two main strategies: generic aggregation and local aggregation. These strategies create multiple triangular structures iteratively, balancing exploration and exploitation. Generic aggregation fosters positive information exchange among diverse structures, while local aggregation strategically places new units around local optimum vertices.	2024
[67]	Hippopotamus optimization (HO)	HO is inspired by the natural behaviors of hippos, offering a novel approach to metaheuristic techniques. HO is defined by a trinary-phase model, encompassing their movements in water, defense mechanisms against predators, and evasion tactics, all mathematically formulated for optimization purposes.	2024
[68]	Electric eel foraging optimization (EEFO)	Electric eels' brilliant group foraging inspires EEFO. It models four main foraging behaviors and balances exploration and exploitation—an energy factor aids in transitioning between global and local search, revealing diverse foraging patterns.	2024
[69]	Super eagle optimization algorithm (SEOA)	SEOA uses a ball security corridor model based on the BSB curve for smooth flight paths and error prevention. It then introduces the super eagle optimization algorithm, mimicking hunting behavior. Two patterns prevent premature convergence, while an information-sharing strategy balances exploration	2024

(*continued on next page*)

Table 1 (*continued*)

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
		and exploitation. Prey uses emotional cues to evade capture and speed up convergence.	
[70]	Puma optimizer (PO)	PO is a new optimization algorithm inspired by pumas' intelligence. It includes unique mechanisms for exploration and exploitation, improving performance across different optimization problems. It also introduces a new intelligent mechanism, a hyper-heuristic for phase change.	2024
[71]	Parrot optimizer (PO)	The Parrot optimizer draws inspiration from specific behaviors observed in trained Pyrrhura Molinae parrots, serving as an efficient optimization method.	2024
[72]	Duck swarm algorithm (DSA)	Duck swarms' foraging behaviors inspire this algorithm. DSA demonstrates superior exploitation, exploration, and local optima avoidance across different test functions. Convergence and population diversity analyses confirm its effectiveness, and sensitivity analysis evaluates its performance.	2024

cooling process [101], Henry gas solubility optimization (HGSO) [102], an optimizer by Coulomb's and Franklin's laws, named the CFA optimizer [103], equilibrium optimizer (EO), a new meta-heuristic approach to modeling control volume mass balance [104], gradient search (GBO), that adopts Newton's method to examine the seek space by adopting several vectors and two major operators [105], and Yin-Yang-pair optimization (YYO) [106]. The following is a list of other types of inspired algorithms based on the TSAs, summarized in Table 3.

- **Evolutionary algorithms (EAs):** EAs are inspired by the natural evolution process, with a multitude of algorithms falling under this category [124], abnormal division of tumor cells due to a decrease in or an incomplete molecular adhesion motivated introduction of the invasive tumor growth optimization algorithm (ITGO) [125], and genetic algorithms (GAs) are rooted in Darwin's theory of natural evolution, driving their fundamental approach [126]. The following is a list of other types of inspired algorithms based on the EAs, summarized in Table 4.

1.2. Plant intelligence to optimization: an introduction to the plant-based optimization algorithms (PBOAs)

Optimization algorithms leveraging artificial intelligence techniques aim to find the best possible solution for a given problem by mimicking intelligent behavior. These algorithms efficiently explore and exploit search spaces to enhance performance and achieve optimal results. Utilizing machine learning, evolutionary computation, and other AI methodologies, they adapt and evolve over time, adapting to changing environments and complex constraints. Such algorithms play a vital role in various fields, including engineering, finance, healthcare, and more, enabling automated decision-making and problem-solving at scale [138, 139]. As Barbara McClintock suggested, plant intelligence links intelligence with fitness [140]. The perception versus reality of plant life is shifting dramatically as the study of plant intelligence gains scientific traction, replacing passivity with dynamism [141]. The plant root cap likely has a systemic structure for responses. Intelligent decisions optimize plant phenotype in changing environments, controlled by the cambium [142]. Spontaneous behavior, numerical ability, and error correction suggest intentionality. Volatile organic compounds act as

Table 2
Some new optimization algorithms are based on the HSAs used in the recent literature.

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
[78]	Human evolutionary model (HEM)	HEM utilizes expert consensus to determine the optimal parameters for intelligent evolution. It resolves disagreements among experts through Mediative Fuzzy Logic (MFL).	2007
[79]	Brain storm optimization (BSO)	The collective behavior of humans inspires BSO during brainstorming. Like natural selection, speciation occurs as the population diverges into separate species. Similarly, solutions in BSO diverge into multiple clusters. New solutions are generated through the mutation of individuals or combinations of individuals.	2011
[80]	Social group optimization (SGO)	It draws inspiration from human social behavior in tackling complex problems.	2016
[81]	Human learning optimization (HLO)	HLO utilizes three learning operators (random, individual, and social) to search for optimal solutions, mimicking human learning mechanisms. However, HLO assumes uniform learning abilities, which is unrealistic as human IQ scores follow a Gaussian distribution. Diverse HLO (DHLO) introduced Gaussian distribution and dynamic adjustment strategies to address this, diversifying learning abilities and enhancing algorithm robustness.	2017
[82]	Supply-demand-based optimization (SDO)	SDO is inspired by economic supply-demand mechanisms, simulating consumer and producer supply.	2019
[83]	Poor and rich optimization (PRO)	PRO is inspired by the ongoing efforts of both the rich and the poor to improve their wealth and status, with the rich aiming to widen the wealth gap and the poor seeking to narrow it by emulating the rich, acknowledging the potential for socioeconomic mobility.	2019
[84]	Billiards-inspired optimization algorithm (BOA)	BOA's optimization mimics billiards: solution candidates act as balls, while the best solutions serve as pockets. Vector algebra dictates ball positions in the search space when they collide.	2020
[85]	Dynastic optimization algorithm (DOA)	DOA is derived from the social dynamics observed within human lineages.	2020
[86]	Human urbanization algorithm (HUA)	HUS draws inspiration from human behavior in urbanization and life improvement. It employs various strategies, such as combined searching in broad and restricted scopes, population management, and concentration of agents during the search process.	2020
[87]	Nomadic people optimizer (NPO)	NPO mimics the migration behavior of groups searching for essential resources. It employs a multi-swarm approach, resembling clans led by leaders, to find optimal solutions based on their positions.	2020
[88]	Human felicity algorithm (HFA)	The essence of HFA lies in pursuing societal happiness, achieved through collective thought shifts. It starts with a diverse population categorized into elites, disciples, and ordinary individuals. Societal thinking evolves through elite influence, personal experiences, and abrupt shifts triggered by war or famine.	2022

(*continued on next page*)

Table 2 (*continued*)

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
[89]	War strategy optimization (WSO)	WSO draws inspiration from military troop movements in warfare, optimizing soldier positions dynamically through attack and defense strategies. It introduces novel weight updates and relocates weaker soldiers for improved convergence and robustness.	2022
[90]	Numeric crunch algorithm (NCA)	NCA harnesses population distribution and a novel hyperbolic acceleration function to drive convergence. It ensures exploration and exploitation within each generation by leveraging population behavior around its members and dynamically adjusting boundaries.	2023
[91]	Human evolutionary optimization (HEO)	HEOA divides the global search into human exploration and development phases. Logistic Chaos Mapping is used for initialization. In the exploration phase, an initial search occurs, followed by the development phase, where the population is categorized into leaders, explorers, followers, and losers, each using different search strategies.	2024
[92]	Partial reinforcement optimizer (PRO)	PRO is influenced by the partial reinforcement effect (PRE) theory in psychology, where learners are intermittently rewarded for strengthening specific behaviors. The timing and selection of reinforcement significantly impact the learner's response rate and strength during the learning process.	2024

communication signals. Game theory explains intelligent interactions, and experience drives adaptation against stresses [143].

The notion of plant intelligence has been a common and influential inspiration for researchers in recent years, particularly in designing various PBOAs [138], for example, the tree growth algorithm in which trees compete to reach sun's rays [144], inspiration by flora's growth pattern to structure artificial flora algorithm (AFA) [145], Table 5 summarizes these innovative algorithms to introduce readers to optimization algorithms based on plant intelligence.

Evolutionary algorithms pose a challenge in employing some differential terms for updating population members. While these terms enable the algorithm to converge faster, mutation methods become necessary to avoid premature convergence in local optima. Such methods provide the proposed algorithms with differential evolution (DE) [159] capabilities. If it is imagined that moving toward the optimal solutions requires a stimulus to move population members, using a differential term can be known as an internal stimulus. The ideas of EAs should be designed so that this stimulus is not solely dependent on the position difference between two population members to maintain population diversity throughout the optimization process. In other words, since it is an external driving force, it can be hoped that the proposed algorithm can maintain population diversity in the entire optimization process. This fact is particularly crucial in solving complex engineering optimization problems. This concept has inspired us to present Ivy algorithm (IVYA) from a different perspective. Also, according to [108, 160], no unique search method can optimize all test functions in various fields. A specific optimizer might produce reasonable solutions for some situations but perform poorly on others. As a result, a novel optimization technique called the IVYA is modeled and explained herein. The Ivy optimization algorithm, introduced for the first time in this article, features a unique search structure compared to other optimization algorithms inspired by tree intelligence. It employs a distinct growth

Table 3
Some new optimization algorithms are based on the TSAs used in the recent literature.

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
[107]	Lightning attachment procedure optimization (LAPO)	LAPO mimics the lightning attachment process, including downward and upward leader movements, unpredictable trajectory, and branch fading. It doesn't need parameter tuning and avoids local optima.	2017
[102]	Henry gas solubility optimization (HGSO)	HGSO imitates the behavior described by Henry's law, which relates the amount of a gas dissolved in a liquid to the gas's partial pressure above the liquid.	2019
[108]	Lévy flight distribution (LFD)	The LFD algorithm draws inspiration from Lévy flight random walks, especially in navigating vast and unfamiliar search domains such as wireless sensor networks (WSNs).	2020
[105]	Gradient-based optimizer (GBO)	GBO, inspired by Newton's method, utilizes two operators, GSR and LEO, along with vectors for exploration. GSR employs gradient-based techniques to enhance exploration and convergence. LEO helps evade local optima.	2020
[109]	Lichtenberg algorithm (LA)	Inspired by radial intra-cloud lightning and Lichtenberg figures, LA harnesses fractal power. It stands out as a hybrid algorithm blending population-based and trajectory-based search methods.	2021
[110]	Light spectrum optimizer (LSO)	The inspiration behind LSO is the dispersion of light at various angles as it passes through raindrops, resulting in the meteorological phenomenon known as the colorful rainbow spectrum. Three distinct experiments were carried out To validate the proposed algorithm.	2022
[111]	Circle search algorithm (CSA)	CSA is influenced by circle geometry, focusing on key attributes like diameter, center, perimeter, and tangent lines. The ratio of radius to tangent segment impacts exploration and exploitation in CSA.	2022
[112]	Elastic deformation optimization algorithm (EDOA)	EDOA draws on physics principles, specifically Newton's second motion law and Hooke's elasticity law, integrating a novel parameter adaptive adjustment for improved exploitation and exploration of the search space.	2022
[113]	Homonuclear molecules optimization (HMO)	HMO mimics atomic arrangements inspired by Bohr's atomic model and homonuclear molecule structures. It initializes atoms and electrons as searching agents. In each iteration, atoms adjust positions based on the best electron to form homonuclear molecules.	2022
[114, 115]	Special relativity search (SRS)	The SRS is based on particle interactions in an electromagnetic field, utilizing the Lorentz force and angular frequency. It incorporates special relativity to determine particle coordinates, employing length contraction and time dilation concepts in its main equation.	2022
[116]	Quadratic interpolation optimization (QIO)	QIO is inspired by GQI, a math-based method addressing traditional quadratic interpolation limitations. It exploits optimization spaces using	2023

(*continued on next page*)

Table 3 (*continued*)

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
		GQI minimizers to identify and utilize promising regions effectively.	
[117]	Sinh Cosh optimizer (SCHO)	The SCHO was formulated drawing upon the mathematical properties of Cosh and Sinh. SCHO is comprised of four main stages: two distinct phases involving exploration and exploitation, an approach centered on bounded search, and an adaptive switching mechanism.	2023
[118]	Kepler optimization algorithm (KOA)	KOA is inspired by Kepler's laws governing planetary motion. It employs four fundamental operators: gravitational force, mass, orbital velocity, and position, which influence the trajectories of planets around the Sun.	2023
[119]	Propagation search algorithm (PSA)	PSA is inspired by wave propagation in electrical transmission lines. It models current and voltage as search agents, with the transmission line's propagation constant guiding exploitation and exploration.	2023
[120]	PID-based search algorithm (PSA)	PSA utilizes an iterative PID algorithm to drive the population towards optimal states by adjusting system deviations continuously. It is mathematically formulated and widely applicable for optimizing diverse search spaces.	2023
[121]	Rime optimization algorithm (RIME)	The RIME algorithm simulates ice growth processes to optimize exploration and exploitation. It refines selection mechanisms and updates populations to enhance performance.	2023
[122]	Hyperbolic sine optimizer (HSO)	HSO utilizes individual population members to explore solution spaces with distinct phases and hyperbolic function convergence. This approach enhances speed, simplifies parameter adjustment, tackles slow convergence, and excels in high-dimensional optimization.	2024
[123]	Newton-Raphson-based optimizer (NRBO)	NRBO is influenced by Newton-Raphson's approach, utilizing the Newton-Raphson search rule (NRSR), the TAO (trap avoidance operator), and matrix groups to explore the search space and achieve optimal outcomes thoroughly.	2024

equation, "growth and evolution Eq. (2)", which sets it apart from conventional optimization methods. Notably, the algorithm incorporates the innovative concept of intertwining two trees, developing a novel search equation. This departure from traditional approaches, particularly those based on plant intelligence, offers a fresh perspective on optimization techniques.

Following this, we introduce the IVYA as part of this approach, the organic and erratic behavior of the Ivy plant, and its ability to grow, move, and intertwine intelligently upwards. Moreover, the IVYA is utilized to tackle challenging engineering optimization tasks.

Below are the main paper's contributions:

1. We compared its findings against ten widely recognized, established, and novel meta-heuristic optimizers to illustrate the proposed Ivy optimizer's performance and demonstrate the benefit of developing this algorithm.

Table 4
Some new optimization algorithms are based on the EAs used in the recent literature.

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
[127]	Find-fix-finish-exploit-analyze (F3EA)	F3EA approaches optimization like a battle, with iterative steps: fFind, fix, finish, exploit, and analyze. It introduces new selection and mutation operators, models fixing as 1D optimization, artificially triggers action on selected solutions, and exploits opportunities through mating.	2019
[128]	Coronavirus optimization algorithm (COA)	COA models coronavirus spread from patient zero, considering reinfection probability and social measures. Initially exponential, the infected population decreases with time and measures. COA has preset parameters and auto-termination, with a parallel multivirus version exploring wider areas. It's also combined with deep learning for parameter optimization.	2020
[129]	Artificial ecosystem-based optimization (AEO)	AEO mimics the behaviors of living organisms: production, consumption, and decomposition, inspired by the flow of energy in the earth's ecosystem.	2020
[130]	Coronavirus optimization algorithm (COVIDOA)	COVIDOA mimics coronavirus hijacking human cells, inspired by its frameshifting replication technique.	2022
[131]	Coronavirus metamorphosis optimization algorithm (CMOA)	CMOA, drawing inspiration from COVID-19, employs synchronous motion and jump vectors to optimize across diverse scenarios more effectively.	2023
[132]	Termite life cycle optimizer (TLCO)	TLCO draws from termite colonies' life cycles and animal movement strategies. Workers, soldiers, and reproductive termites carry out specialized colony growth and survival tasks.	2023
[133]	Guided learning strategy (GLS)	GLS assesses population dispersion via the standard deviation of recent generations' locations, guiding algorithm exploration or exploitation accordingly. It enhances various algorithms' performance by adapting to their needs.	2024
[134]	Love evolution algorithm (LEA)	LEA, inspired by stimulus–value–role theory, undergoes three phases: stimulus, value, and role. Both partners evolve, benefiting from these phases regardless of relationship outcomes.	2024
[135]	Walrus optimizer (WO)	WO draws inspiration from walruses' behaviors in various activities such as migration, breeding, roosting, escaping, gathering, and feeding, guided by signals indicating safety and danger.	2024
[136]	Fitness-distance balance-based EA (FDB-EA)	The EAs were enhanced with the FDB method to address the complex optimization problem with numerous local solution traps and both discrete and continuous variables. FDB improves search effectiveness by balancing exploration and diversity, similar to natural selection.	2024
[137]	Symmetric projection optimizer (SPO)	The SPO algorithm leverages the projection search, rapidly fitting a projection plane using the symmetry of two points, and detecting minima	2024

(*continued on next page*)

Table 4 (*continued*)

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
		based on fitness function values, expediting the identification of regions with potential extreme values.	

2. In contrast to the established comparative algorithms, our proposed technique exhibits improved optimization speed and suitable computational complexity.
3. The new approach outperforms many prior enhanced algorithms regarding optimization outcomes for standard test functions.
4. Compared to many other proposed algorithms, the proposed approach has greater optimization power and is better suited for various engineering optimization tasks.

The subsequent sections of this paper are structured as follows. Section 2 proposes the Ivy algorithm. Section 3 exhibits the solutions derived from the optimization standard test functions. The section also compares the treatment of the Ivy algorithm with some modern optimizers. In Section 4, the IVYA has been applied to various types of nonlinear engineering optimization functions. Finally, Section 5 processes several conclusions for the proposed IVYA.

2. Ivy algorithm

This section describes the steps to establish the IVYA and the framework of this algorithm in detail.

The IVYA algorithm is a population-based optimization approach that operates through an iterative process, harnessing the collective search capability of the population members. IVYA population consists of individuals resembling a community of Ivy plants. The position of each member within the search space presents potential values for the problem's decision variables. In mathematical terms, each population member, i.e., an Ivy plant, is represented as a vector and serves as a potential solution for the problem.

2.1. A brief introduction to an Ivy plant growth pattern

Ivy is a plant that creeps along the ground and produces flowers in summer. Hedera is the scientific name for the group of plants that Ivy belongs to, and they are typically found in Europe, Asia, North Africa, and the Pacific regions. About 15 different species of this plant exist, and they all start their growth by creeping along the ground until they reach a surface such as a tree, rock, wall, or fence. When they reach this climbing-clinging point, Ivy can access sunlight directly, increasing its growth rate. Sometimes, Ivy can even grow to be several tens of meters tall! Climbing plants are great for filling empty areas without much greenery because they grow by hanging or attaching to walls. They have long branches that can reach far from the roots and soil. One cool thing about these plants is that you can easily direct their growth in any direction. This makes it easy to guide the branches and create unique wall patterns. This plant is an evergreen ground cover that can quickly adapt to various environmental conditions. Its flexible stems can grow and hide under a green fence within months or years. According to growth conditions, climbing plants are typically classified into the following two categories:

1. Plants that are better suited for growing indoors and cannot thrive outside.
2. Plants that are climbing plants that are specific to outdoor environments and may not be suitable for growing indoors.

Table 5
Some new optimization algorithms are based on the PBOAs used in the recent literature.

Ref.	The name of the algorithm	Inspirational origin or motivator	Year
[146]	Invasive weed optimization (IWO)	IWO imitates weed colonization's resilience, adjustment, and unpredictability in a concise yet potent optimization algorithm.	2006
[147]	Plant growth optimization (PGO)	PGO constructs an artificial plant growth model based on plant growth attributes, encompassing leaf expansion, spatial distribution, phototropism, and branching.	2008
[148]	Artificial plant optimization algorithm (APOA)	APOA introduces a shrinkage coefficient to confine branch dimensions within bounds. A new function verifies whether particles are within the feasible region. One-dimensional optimization methods are applied to generate new positions, ensuring branches remain within the feasible region.	2011
[149]	Flower pollination algorithm (FPA)	The drawing inspiration is from the evolutionary process of pollination in flowers.	2012
[150]	Root mass optimization (RMO)	RMO mimics plant root growth dynamics.	2013
[151]	Root growth algorithm (RGA)	A mathematical framework and architecture were devised to model plant root growth, focusing on soil-root interactions.	2014
[152]	Tree-seed algorithm (TSA)	TSA is a heuristic optimization method mimicking tree-seed interactions. Trees represent potential solutions in an n-dimensional space, with seeds generated from them. Seeds replace trees based on quality, guided by a parameter called search tendency (ST), over multiple iterations.	2015
[153]	Plant propagation algorithm (PPA)	PPA is inspired by the seasonal production of fruits and seeds, which resembles a restaurant serving customers. This analogy reflects birds and animals feeding on ripe fruit, aiding seed dispersion.	2015
[154]	Sunflower optimization (SFO)	SFO draws inspiration from the movement patterns of sunflowers.	2019
[155]	Carnivorous plant algorithm (CPA)	CPA is inspired by the adaptive strategies carnivorous plants employ to thrive in challenging environments.	2021
[156]	Poplar optimization algorithm (POA)	POA imitates the reproduction methods of poplar trees, incorporating both sexual and asexual propagation. It outlines how these methods are executed by individuals and integrates a mutation strategy inspired by the backtracking search algorithm to maintain diversity.	2022
[157]	Growth optimizer (GO)	GO is primarily inspired by how individuals learn and reflect during their social growth. Learning involves acquiring knowledge from the external environment, while reflection entails assessing one's own shortcomings and adjusting learning strategies to facilitate personal development.	2023
[158]	Orchard algorithm (OA)	OA is inspired by fruit gardening, integrating actions like irrigation, fertilization, pruning, and grafting to cultivate a productive orchard. It balances exploration and exploitation through individual and collective behaviors, employing operators such as annual growth, pruning, and grafting for efficient search space navigation.	2023

2.2. The Ivy algorithm by modeling the Ivy plant growth pattern with assumptions

This subsection explains in detail the concept of the proposed IVYA, which simulates the different life stages of Ivy, including growing, rising, and spreading in the swarm of Ivy plants. Generally, the IVYA parts are as shown:

- Creation of primary populations of Ivy as candidate solutions.
- The steps of researching and population search in the proposed IVYA:

  Step 1: Coordinated and ordered population growth.
  Step 2: Growth to obtain a source of sunlight for Ivy plants.
  Step 3: Spreading and evolution of Ivy plants.

- Survivor selection

At the beginning, a swarm of Ivy trees is generated so that the location vector of the i-th Ivy is denoted as I_i, and its growth velocity Gv_i is determined. Next, the Ivy trees are arranged according to their fitness value, ranging from the best solution to the worst solution.

In this experimental inspiration, every Ivy applies knowledge of its nearby Ivy in the ordered swarm of plants and is given direction of growth via this member.

2.2.1. The formation of primary populations of Ivy trees as candidates for solutions

We denote $Npop$ and D the number of the population members and the number of the problem's decision variables, respectively. Hence, the i-th population member has the form $I_i = (I_{i1}, \ldots, I_{iD})$, where $i = 1, 2, \ldots, Npop$, and the total population of Ivy plants we denote $\overrightarrow{I} = (I_1, \ldots, I_i, \ldots, I_{Npop})$.

At the algorithm's start, the IVYA population's initial position in the search space is randomly determined using Eq. (1).

$$I_i = I_{min} + rand(1, D) \odot (I_{max} - I_{min}),\ i = 1, \ldots, Npop, \tag{1}$$

where a vector of the dimension D of uniformly distributed random numbers from the interval $[0,\ 1]$ is denoted by $rand(1, D)$. I_{max} and I_{min} are the upper and lower bound of the search space, respectively, and the Hadamard product of two vectors (also referred to as the element-wise product and denoted in Matlab as ".∗") is represented by the operation "⊙".

2.2.2. The steps of researching and population search in the proposed IVYA

Step 1. Coordinated and ordered population growth

As mentioned above, Ivy is a creeping plant that grows over time. In this article, we have assumed the growth rate Gv of an Ivy plant as a function of time given by the differential equation:

$$\frac{dGv(t)}{dt} = \psi \cdot Gv(t) \cdot \varphi(Gv(t)), \tag{2}$$

where Gv, and φ are the growth rate, growth velocity, and correction factor that indicates the deviation from growth, respectively. In the proposed algorithm, Eq. (2) is modeled, based on a data-intensive experimental and simulation process, by the following difference equation for the growth velocity $Gv_i(t)$ of the member I_i

$$\Delta Gv_i(t+1) = rand^2 \odot (N(1, D) \odot \Delta Gv_i(t)), \tag{3}$$

where vectors $\Delta Gv_i(t)$ and $\Delta Gv_i(t+1)$ represent the growth rates in a discrete-time system (at the time instants t and $t+1$), $rand$ is a uniform random real number from the interval $[0,\ 1]$ (i.e., $rand \in U[0, 1]$ and $rand^2$ is a random number from the random variable with the PDF equal to $1/(2\sqrt{x})$). $N(1, D)$ represents a random vector of the dimension D, whose components are random numbers from the standard Gaussian (Normal) distribution.

Step 2. Growth as a means of obtaining a source of sunlight for Ivy plants

For the overall well-being of Ivy trees in nature, finding an attaching surface (e.g., a wall, rock, or tree) soon for climbing toward sunlight is vital. In wild forests, young Ivy can appropriately choose the direction of its growth towards the nearest tree, and very often, it is also another older Ivy that has already found its support. In this way, Ivy can gradually fill a large continuous forest area. Fortunately for the other tree species in the forest, this climbing of young Ivy over older Ivy leads to the fact that only the strongest of the entire group of young and old Ivy survive, nearly regardless of their age [161,162].

The simulation of this behavior of Ivy in nature leads in the proposed algorithm to the fact that the i-th member of the population I_i chooses its closest most vital neighbor (by the value of the fitness function) I_{ii} as a means of self-improvement (see Fig. 1). If we denote $\overrightarrow{I}^S = [I_1^s, \ldots, I_j^s, \ldots, I_{Npop}^s]$ sorted $\overrightarrow{I}$ (from the best to the worst member, i.e., $I_{Best} = I_1^s$), then

$$I_{ii} = \begin{cases} I_{j-1}^s, & I_i = I_j^s; \\ I_i, & I_i = I_{Best}. \end{cases} \tag{4}$$

The following equation describes how member I_i uses member I_{ii} to climb and move logically in the direction of the light source

$$I_i^{new} = I_i + |N(1, D)| \odot (I_{ii} - I_i) + N(1, D) \odot \Delta Gv_i,\ i = 1, 2, \ldots, Npop, \tag{5}$$

with

$$\Delta Gv_i = \begin{cases} I_i \oslash (I_{max} - I_{min}), & Iter = 1; \\ rand^2 \odot (N(1, D) \odot \Delta Gv_i), & Iter > 1, \end{cases} \tag{6}$$

where $|N(1, D)|$ is the vector, whose components are absolute values from components of the vector $N(1, D)$, and the operation "$\overrightarrow{u} \oslash \overrightarrow{v}$" is the Hadamard division of a vector $\overrightarrow{u}$ by a vector $\overrightarrow{v}$ (also called the element-wise division and denoted in Matlab as "./").

Step 3. Spreading and evolution of Ivy plants

After the phase of a global wandering of the member I_i through search space to the closest, most vital neighbor I_{ii}, there is a phase in which member I_i tries to follow the best member directly I_{Best} of the entire population, which is equivalent to finding better optimal solutions around the member I_{Best}. This phase is mathematically formulated

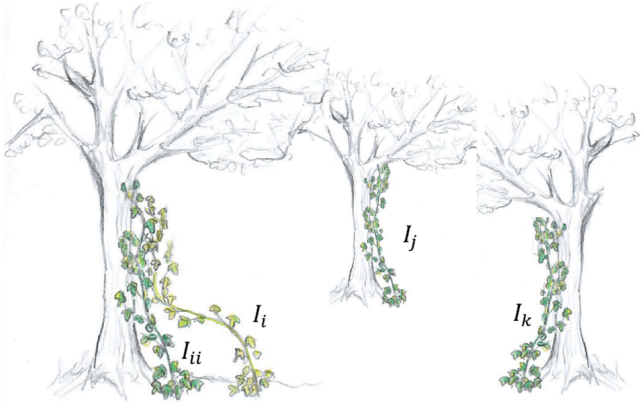


Fig. 1. The i-th member of the population I_i chooses his closest, most vital neighbor I_{ii}.

in the *Iter*-th step of the IVYA as follows:

$$I_i^{new} = I_{Best} \odot (rand(1,D) + N(1,D) \odot \Delta Gv_i). \quad (7)$$

Subsequently, the new value of the growth rate ΔGv_i^{new} of the current member I_i^{new} is computed by the following formula (which is entirely analogous to the formula used in the initialization step for the computation of ΔGv_i)

$$\Delta Gv_i^{new} = I_i^{new} \oslash (I_{max} - I_{min}) \ . \quad (8)$$

2.2.3. Survivor selection

Hoflacher and Bauer [48] studied the ability of the youngest and oldest (basal) parts of Ivy plants to absorb sunlight during the alternation of two growth phases of Ivy trees, namely the phase of formation of "juvenile leaves" (which is typical of the stage of rapid upward growth) and "adult leaves" (which is typical for the stage of expansion of new branches in width and flowering).

To model these two alternating phases in the life of Ivy trees, "climbing" and "expanding," in the proposed IVYA, we used the following decision approach. When the objective function value $f(I_i)$ of the member I_i is less than the multiple of $f(I_{best})$ by the parameter $\beta = (2 + rand)/2$. Then, the Ivy tree starts to expand branches and leaves in width (given by Eq. (5)). Otherwise, the Ivy grows upwards and climbs (given by Eq. (7)).

In IVYA, to model Ivy plants' behavior, after finishing each iteration, the swarm $\vec{I}$ from the previous algorithm, the cycle is merged with the novel-generated swarm in the current iteration $\vec{I}^{new}$, i.e., we get the list $\vec{I}^{Merged} = \{\vec{I}, \vec{I}^{new}\}$, then the fitness value ranks these merged swarm members from the most favorable to the least favorable, i.e., we get the vector $\vec{I}^{M/S} = [I_1^{M/S}, I_2^{M/S}, \ldots, I_{Npop}^{M/S}, I_{Npop+1}^{M/S}, \ldots, I_{2 \cdot Npop}^{M/S}]$. Subsequently, the first *Npop* top-ranked members (considering the constant algorithm population size equal to*Npop*) are chosen as final members of the current population, i.e., $\vec{I} = [I_1^{M/S}, I_2^{M/S}, \ldots, I_{Npop}^{M/S}]$.

Algorithm 1: The Ivy algorithm:

1: Put amounts of the parameters of the Ivy algorithm: I_{min} , I_{max}, $Iter_{max}$, and $Npop$. Set $Iter = 1$.

2: Compute the initial population of the ivy plants $\vec{I} = (I_1, I_2, \ldots, I_i, \ldots, I_{Npop})$ by Eq. (1).

3: Compute ΔGv_i by Eq. (6), i.e., $\Delta Gv_i = I_i \oslash (I_{max} - I_{min})$ for $i = 1,2, \ldots, Npop$.

4: Calculate values of the objective function of all members of the initial population and resort $\vec{I}$ by fitness values (from the best to the worst member) and set $I_{Best} = I_1$.

5: while $Iter \leq Iter_{max}$ **do**

6: for $i = 1$ **to** $Npop$ **do**

7: Select the member I_{ii} by Eq. (4).

8: Compute $\beta = (2 + rand)/2$ and $\Delta Gv_i = rand^2 \cdot (N(1,D) \odot \Delta Gv_i)$;

9: if $f(I_i) < \beta \cdot f(I_{Best})$

10: $I_i^{new} = I_i + |N(1,D)| \odot (I_{ii} - I_i) + N(1,D) \odot \Delta Gv_i$;

11: else if

12: $I_i^{new} = I_{Best} \odot (rand(1,D) + N(1,D) \odot \Delta Gv_i)$;

13: end if

14: Evaluate the fitness of I_i^{new}**;**

15: Compute $\Delta Gv_i^{new} = I_i^{new} \oslash (I_{max} - I_{min})$;

16: Add I_i^{new} as the i-th component of the vector $\vec{I}^{new}$ and set $\Delta Gv_i = \Delta Gv_i^{new}$.

17: end for

18: Merge populations $\vec{I}$ and $\vec{I}^{new}$ (i.e., create the list $\vec{I}^{Merged} = \{\vec{I}, \vec{I}^{new}\}$).

19: Denote $\vec{I}^{M/S}$ the sorted version of $\vec{I}^{Merged}$ by fitness values (from the best to the worst member) and choose the best $Npop$ members from $\vec{I}^{M/S}$ as the next generation of the proposed IVYA (i.e., $\vec{I} = (I_1^{M/S}, I_2^{M/S}, \ldots, I_{Npop}^{M/S})$) and set $I_{Best} = I_1$.

20: $Iter = Iter + 1$;

21: end while

22. Show the optimal solution I_{Best}, founded by the Ivy optimizer.

Fig. 2. The suggested Ivy optimization algorithm pseudo-code.

In final, Fig.2 shows the IVYA's pseudo-code.

2.3. Computational complexity of the IVYA

Be aware that initialization, fitness assessment, and population update are the three operations that account for the majority of the IVYA's computational complexity. It should be noted that the initialization method has an $O(Npop)$ computational cost for $Npop$ persons. The search for the better location and updating the position vector of the entire swarm make up the updating mechanism, which shows a computational complexity of $O(Iter_{max} \cdot Npop \cdot D) + O(Iter_{max} \cdot Npop)$, that $Iter_{max}$ shows the generations of IVYA, and D shows the problem's dimension. Consequently, the proposed IVYA's cost is calculated by $O((D \cdot Iter_{max} + 1 + Iter_{max}) \cdot Npop)$.

Algorithm 1: The Ivy algorithm:

3. Simulation analysis

In this part of the paper, the implementation results of the suggested IVYA are provided and contrasted with those of the standard and numerous other recently proposed optimizers.

3.1. Effectiveness analysis of the IVYA optimizer

3.1.1. Selection of test functions

As indicated in Tables 6 and 7 for the conventional and real-parameter test functions, respectively, 26 benchmark test functions of dimension 30 were selected to assess the effectiveness of the IVYA. The data and details for these test functions are available in [5,163,[180]]. Among these, F1–F6 are conventional single-peak functions with just global and no local optimums, which are better suited to testing the algorithm's convergence rate. Traditional multipeak functions are F7–F12. The local extremum positions of the multipeak function are numerous. As indicated in Table 7, the unimodal functions (F13 to F17), basic multimodal functions (F18 to F24), and expanded multimodal functions (F25 and F26) are among the shifted real-parameter problems. For the conventional and real-parameter test functions, respectively, the size of function evaluations has been chosen to be 60,000 and 300,000, which is the same as that of [5,163].

3.1.2. Effectiveness analysis of different sizes of Ivy's swarm in the Ivy optimizer

Table 8 shows IVYA results for the functions over 30 separate iterations using the criteria of standard deviation (Std.) and average (Mean) values. The table includes the mean value of Friedman's test and the ranking of algorithms overall, showcased in its last two rows. The IVYA is suitable for selecting population 60, according to the findings in this table. However, a much deeper view of this table shows that IVYA was powerful in identifying suitable and optimal results for all five populations from 30 to 90. When selecting the initial population size for an algorithm, it's important to balance adequate coverage of the problem space and the number of iterations required. If the population size is too low, the algorithm may not be able to cover enough of the search space to find a solution. Conversely, if the population size is too high, population members may not be updated frequently enough, which can negatively impact the search process. Based on our findings IVYA, a population size between sixty and ninety is generally suitable for achieving optimal algorithm behavior.

Additionally, test functions F14, F16, F19, F21, F24, and F26 are shown in Fig. 3 with their characteristics of IVYA's convergence, demonstrating the algorithm's quick and appropriate convergence. Fig. 4 shows the Friedman test [164–166] for each test function with five populations from 30 to 90 is provided below.

3.1.3. IVYA's convergence with exploration and exploitation analysis

The convergence and trajectory are employed in Fig. 5 to accredit the performance of the IVYA. The qualitative measures, including the topology of the problem space, are shown in this picture. The 1st, 2nd, and 3rd columns include the solutions' explore histories, convergence curves, and trajectory curves, respectively. As observed, the population dispersion is low, and the population members converged at an acceptable rate for the first four functions, which are more straightforward. However, in the following functions, which are among the functions (shifted-rotated complex), the population dispersion in the problem's feasible space is high, and the problem's dynamics are preserved through the final iterations. This problem demonstrates the accuracy of algorithm modeling because natural evolution and other processes are continuously evolving. Changes in the variables illustrate the algorithm's ability to pass through local optima. For instance, in the graphs on the F13 function, the two search variables underwent changes that revolved around the optimal points, after which these variables

Table 6
The traditional benchmark functions.

Type of test function	Used range	F_{min}
The traditional unimodal test functions:		
$F_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]$	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le D\}$	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{D} [x_i + 0.5\text{AptCommand230B};^2$	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{D} i x_i^4 + random(0,\ 1)$	$[-1.28, 1.28]$	0
The traditional multimodal test functions:		
$F_8(x) = \sum_{i=1}^{D} -x_i \sin\sqrt{\lvert x_i \rvert}$	$[-500, 500]$	$-418.9829 \bullet D$
$F_9(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\frac{x_i}{\sqrt{i}} + 1$	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{D}(10\sin(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2) + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$, where	$[-50, 50]$	0
$u(x_i, a, k, n) = \begin{cases} k(x_i - a)^n, & x_i > a; \\ 0, & -a \le x_i \le a; \\ k(-x_i - a)^n, & x_i < -a, \end{cases}$ and $y_i = 1 + (x_i + 1)/4$.		

Table 7
The real-parameter shifted functions with $F_{min} = 0$.

Function name	Definition of the test function	Search range
Usual parameters of unimodal test functions:		
Shifted Sphere	$F_{13}(x) = \sum_{i=1}^{D} z_i^2$, with the shifted global optimum in $o = [o_1, o_2, \ldots, o_D]$. And $z = x - o$ for the all shifted functions and $z = (x - o)\cdot M$.	$[-100, 100]^D$
Shifted Schwefel's Problem 1.2	$F_{14}(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2$, $z = x - o$.	$[-100, 100]^D$
Shifted Rotated High Conditioned Elliptic	$F_{15}(x) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2$, $z = (x - o)\cdot M$, where M is an orthogonal matrix.	$[-100, 100]^D$
Shifted Schwefel's Problem 1.2 with Noise in Fitness	$F_{16}(x) = (\sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2)\cdot(1 + 0.4\cdot\lvert N(0,1)\rvert)$.	$[-100, 100]^D$
Schwefel's Problem 2.6 with Global Optimum on Bounds	$F_{17}(x) = \max\{\lvert A_i x - B_i\rvert\}$, A is a matrix of the type $D \times D$, its the i-th row is denoted as A_i, $B_i = A_i \cdot o$, and o is a vector of the type $D \times 1$, where o_i are randomly generated reals from $[-100, 100]$.	$[-100, 100]^D$
Usual parameters of multimodal test functions:		
Shifted Rosenbrock's	$F_{18} = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$, $z = x - o + 1$.	$[-100, 100]^D$
Shifted Rotated Griewank's without Bounds	$F_{19} = \frac{1}{4000}\sum_{i=1}^{D}(z_i)^2 - \prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1$, $z = (x - o)\cdot M$ and $M = M'(1 + 0.3\lvert N(0,1)\rvert)$, with matrix M' of the linear transformation, orthogonal matrix with the condition number equal to 3.	$[-600, 600]^D$
Shifted Rotated Ackley's with Global Optimum on Bounds	$F_{20} = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)\right) + 20 + e$, $z = (x - o)\cdot M$.	$[-32, 32]^D$
Shifted Rastrigin's	$F_{21} = \sum_{i=1}^{D}(z_i^2 - 10\cos(2\pi z_i) + 10)$, $z = x - o$.	$[-5, 5]^D$
Shifted Rotated Rastrigin's	$F_{22} = \sum_{i=1}^{D}(z_i^2 - 10\cos(2\pi z_i) + 10)$, $z = (x - o)\cdot M$.	$[-5, 5]^D$
Shifted Rotated Weierstrass	$F_{23} = \sum_{i=1}^{D}(\sum_{k=0}^{20} 0.5^k\cos(2\pi 3^k(z_i + 0.5))) - D\sum_{k=0}^{20} 0.5^k\cos(2\pi 3^k)$.	$[-0.5, 0.5]^D$
Schwefel's Problem 2.1	$F_{24} = \sum_{i=1}^{D}(A_i - B_i(x))^2$, $A_i = \sum_{j=1}^{D}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j)$, $B_i(x) = \sum_{j=1}^{D}(a_{ij}\sin x_j + b_{ij}\cos x_j)$, a_{ij}, b_{ij} are random integers from the interval $[-100, 100]$, α_j are random reals from $[-\pi, \pi]$.	$[-\pi, \pi]^D$
Usual parameters of expanded multimodal test functions:		
Shifted Expanded Griewank's plus Rosenbrock's	$F_{25} = F_8(F_2(z_1, z_2)) + \cdots + F_8(F_2(z_{D-1}, z_D)) + F_8(F_2(z_D, z_1))$, $z = x - o + 1$.	$[-3, 1]^D$
Shifted Rotated Expanded Scaffer's F6	$F_{26} = F_6(z_1, z_2) + \cdots + F_6(z_{D-1}, z_D) + F_6(z_D, z_1)$, $z = (x - o)\cdot M$.	$[-100, 100]^D$

converged.

A critical performance measure for optimization algorithms is balancing exploration and exploitation indices during optimization. These two values are essential for optimizing algorithms and are associated with different search strategies. Exploration involves an extensive search and investigation in the solution space to discover new and potentially better solutions [167]. Typically, this search strategy leads to solutions focused on finding the global optimum or regions of higher quality. In contrast, exploitation focuses on optimizing local regions of the solution space by improving and optimizing existing solutions through local searches around feasible solutions. Effective solutions are usually exploited using a local search [168].

Fig. 6. illustrates changes in exploration and exploitation during IVYA iterations for objective functions F14 to F26. With a well-designed search process, algorithms such as IVYA can be effectively enhanced using improvement methods. Effective methods to update population positions are crucial for maintaining population diversity and preventing premature convergence. The calculations in this section are based on what was presented in [167].

Table 8
The Mean and Std. solutions for all the functions.

Function	$Npop = 30$ Mean Std.	$Npop = 45$ Mean Std.	$Npop = 60$ Mean Std.	$Npop = 75$ Mean Std.	$Npop = 90$ Mean Std.
F1 to F4	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00
F5	2.40E + 01 4.51E-01	2.48E + 01 9.71E-01	2.49E + 01 7.51E-01	2.50E + 01 1.94E-01	2.56E + 01 4.81E-01
F6	2.18E-01 3.25E-01	1.00E-01 1.52E-01	2.53E-02 2.71E-02	6.75E-03 2.55E-03	4.71E-02 9.86E-02
F7	1.59E-05 1.15E-05	1.29E-05 1.43E-05	1.46E-05 1.23E-05	2.01E-05 1.43E-05	1.59E-05 1.28E-05
F8	−3.09E + 03 4.79E + 02	−3.30E + 03 4.71E + 02	−3.57E + 03 6.61E + 02	−3.16E + 03 3.74E + 02	−3.50E + 03 5.18E + 02
F9	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00
F10	8.88E-16 0.00E + 00	8.88E-16 0.00E + 00	8.88E-16 0.00E + 00	8.88E-16 0.00E + 00	8.88E-16 0.00E + 00
F11	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00	0.00E + 00 0.00E + 00
F12	1.40E-02 1.87E-02	4.18E-03 6.27E-03	1.11E-03 1.57E-03	2.68E-03 6.01E-03	2.77E-03 7.01E-03
F13	4.09E-28 6.34E-28	2.56E-25 9.07E-25	1.31E-23 5.89E-23	5.57E-21 2.11E-21	1.48E-20 2.35E-20
F14	8.37E-02 1.48E-02	7.43E-03 4.46E-03	1.85E-02 2.59E-02	6.37E-02 3.49E-02	1.37E-01 6.29E-02
F15	1.10E + 06 3.33E + 05	1.19E + 06 3.20E + 05	1.54E + 06 3.83E + 05	1.64E + 06 6.14E + 05	1.73E + 06 2.66E + 05
F16	2.57E + 04 5.25E + 03	2.17E + 04 4.44E + 03	1.03E + 04 4.72E + 03	1.16E + 04 5.23E + 03	7.50E + 03 3.31E + 03
F17	1.30E + 04 4.32E + 03	1.10E + 04 3.74E + 03	2.59E + 03 2.62E + 03	3.64E + 03 2.40E + 03	3.85E + 03 2.23E + 03
F18	5.82E + 02 4.35E + 02	6.27E + 02 4.42E + 02	8.17E + 02 4.14E + 02	8.73E + 02 4.59E + 02	8.54E + 02 4.30E + 02
F19	1.92E-02 1.28E-02	2.20E-02 1.75E-02	2.39E-02 9.29E-03	1.46E-02 9.25E-03	1.28E-02 1.21E-02
F20	2.06E + 01 3.67E-01	2.07E + 01 3.65E-01	2.01E + 01 3.49E-02	2.05E + 01 3.90E-01	2.06E + 01 3.63E-01
F21	1.78E + 02 2.54E + 01	1.61E + 02 2.89E + 01	1.17E + 02 3.14E + 01	1.60E + 02 3.41E + 01	1.59E + 02 2.80E + 01
F22	3.27E + 02 8.39E + 01	3.59E + 02 1.14E + 02	1.02E + 02 7.92E + 01	1.23E + 02 1.01E + 02	1.96E + 02 6.57E + 01
F23	3.05E + 01 5.12E + 00	2.93E + 01 3.57E + 00	1.59E + 01 4.34E + 00	1.93E + 01 4.45E + 00	1.59E + 01 3.49E + 00
F24	7.40E + 03 6.90E + 03	1.51E + 04 1.88E + 04	1.98E + 04 7.95E + 03	1.36E + 04 1.57E + 04	9.21E + 03 8.44E + 03
F25	2.68E + 01 1.43E + 01	2.63E + 01 1.54E + 01	5.88E + 00 1.57E + 00	3.10E + 01 2.26E + 01	2.22E + 01 1.31E + 01
F26	1.29E + 01 6.08E-01	1.34E + 01 3.60E-01	1.20E + 01 6.33E-01	1.34E + 01 4.16E-01	1.33E + 01 5.03E-01
MFr	3.2692	3.2115	2.3269	3.1731	3.0192
Final rank	5	4	1	3	2

Fig. 7 presents the average values recorded in Fig. 6. As depicted in this figure, the balance between the exploration and exploitation values seems entirely appropriate.

3.2. A competitive study with new algorithms

We have employed a number of the optimizers introduced in recent studies to demonstrate the optimization power of the IVYA under competitive analysis in equal conditions. To this goal, the optimization problems shown in Tables 6 and 7 and effectively applied in numerous works have been chosen as criteria for evaluating the effectiveness of the suggested Ivy method. All optimization methods run on a single machine, with each test function getting 30 executions and each function having 30 dimensions. For the conventional and real-parameter test

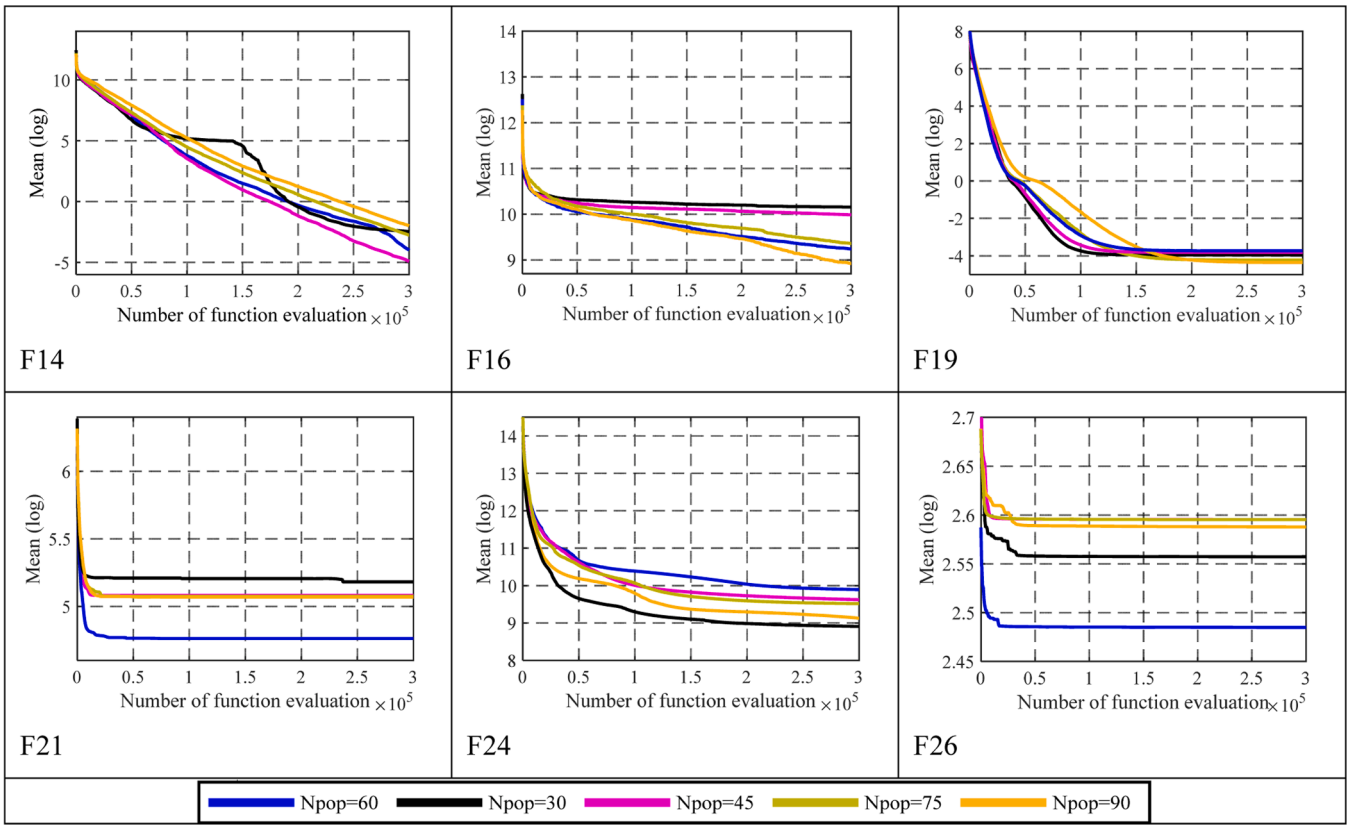


Fig. 3. Convergence behavior of the Ivy optimization algorithm.

functions, the corresponding number of function evaluations are 60,000 and 300,000, respectively. Each algorithm's control parameters and population size were chosen according to the primary reference of the algorithm, as given in Table 9.

The optimal solutions of the competitive metaheuristics have been shown in Table 10 by three measurement criteria (Std., Mean, and the winner of 30 independent executions) for every optimizer's optimal function. The IVYA has successfully obtained the most excellent Mean values in the more test functions, as is evident from these tables. Under comparable circumstances, the suggested IVYA was able to outperform the ten recently introduced metaheuristic algorithms in terms of optimizing conventional and real-parameter test functions. In general, the proposed IVYA has outperformed in comparison with the Rao-1, JAYA, AO, PSO, AOA, BBO, GWO, RSA, SCA, and WOA algorithms in the more test functions, demonstrating its supremacy and great value as a developing optimization method. The IVYA algorithm has been proven to perform better or equally well than other algorithms in seven of the fourteen first functions and five of the twelve functions in the F15 to F26 range. This balanced approach, which includes simple and shifted-rotated functions, indicates that the algorithm can comprehensively cover the problem space through the ability to search. Hence, the proposed algorithm can be implemented with various improvement methods to address different optimization problems.

The Friedman test [164–166] is applied in the experiment to rank each algorithm's performance. This test rates the value of each algorithm from lowest to highest and determines whether there is a substantial difference between the IVYA and the comparable optimizers. The positions achieved in this test are given in Table 11. They rely on the Mean indicator, showcasing the mean values derived from the solutions produced by the IVYA and various other metaheuristics. With a mean rank of 2.4038 for the Friedman test, the proposed IVYA is definitely the best approach from the comparisons when looking at this table.

Fig. 8 depicts the convergence speed of the studied methods to solve functions F13 to F26 (real-parameter test functions) for the number of function evaluations NFEs $\in [1,\ 3 \times 10^5]$. Based on the solutions to the problems, the proposed Ivy method shows adequate power to flee the local optimal position. These diagrams show that the result achieved via IVYA has a better and lower curve than the studied methods in most test problems in the first iterations, indicating an exceptional superiority of the IVYA. The graphs showing the convergence of the IVYA algorithm reveal that the search process stays in most iterations, except when the algorithm finds a suitable answer quickly. This demonstrates the algorithm's capability to maintain population diversity even after multiple iterations. One contributing factor is the population updating approach, which does not solely rely on differential terms. This approach has played a significant role in keeping the population diverse.

Fig. 9 shows a graphical representation of the statistical analysis of Table 10's results. In this figure, "SR" shows an algorithm's total rank, "Nb" and "Nw" show the number of times an algorithm gets the best answer and the worst answer, and "Mr" shows the algorithm's average rank.

The results of a comparison study between the IVYA algorithm and several others are presented in Table 12. The study followed the Wilcoxon signed-rank test [164,176] with a significance level of 0.05. An algorithm is considered significantly better than the other one if the p-value is less than the significance level and its 0.95 confidence interval does not include zero. Based on these criteria, IVYA outperforms all algorithms except for BBO. When comparing IVYA with BBO, the p-value is more significant than 0.05, and the confidence interval includes the zero value. However, a further analysis of the confidence interval reveals that the preceding interval to zero is wider than the following one. This analysis indicates that IVYA provides better solutions than BBO.

4. Optimization of mechanical design

By examining how well the planned IVYA handles engineering

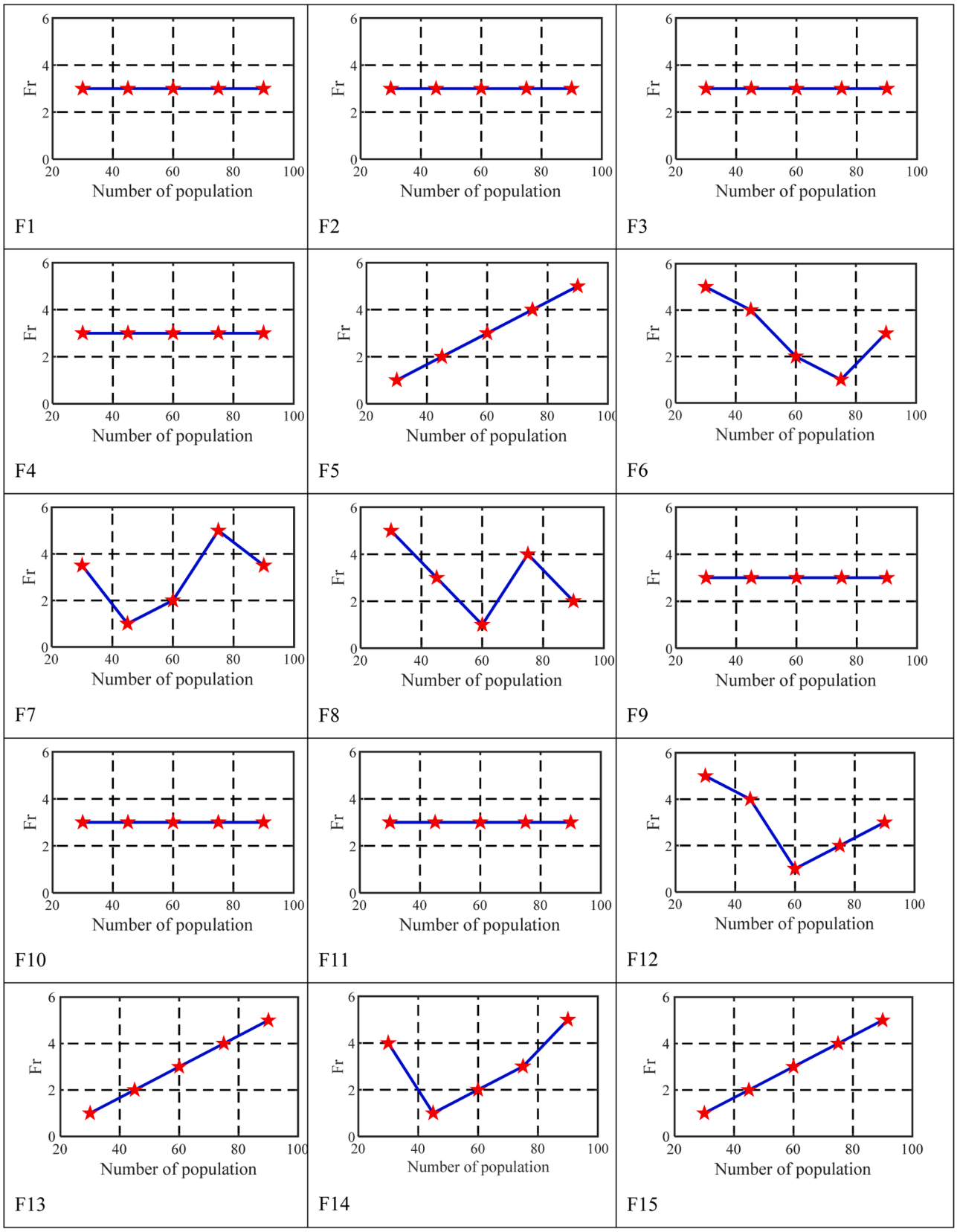


Fig. 4. Friedman test for each test function by the IVYA for all five populations.

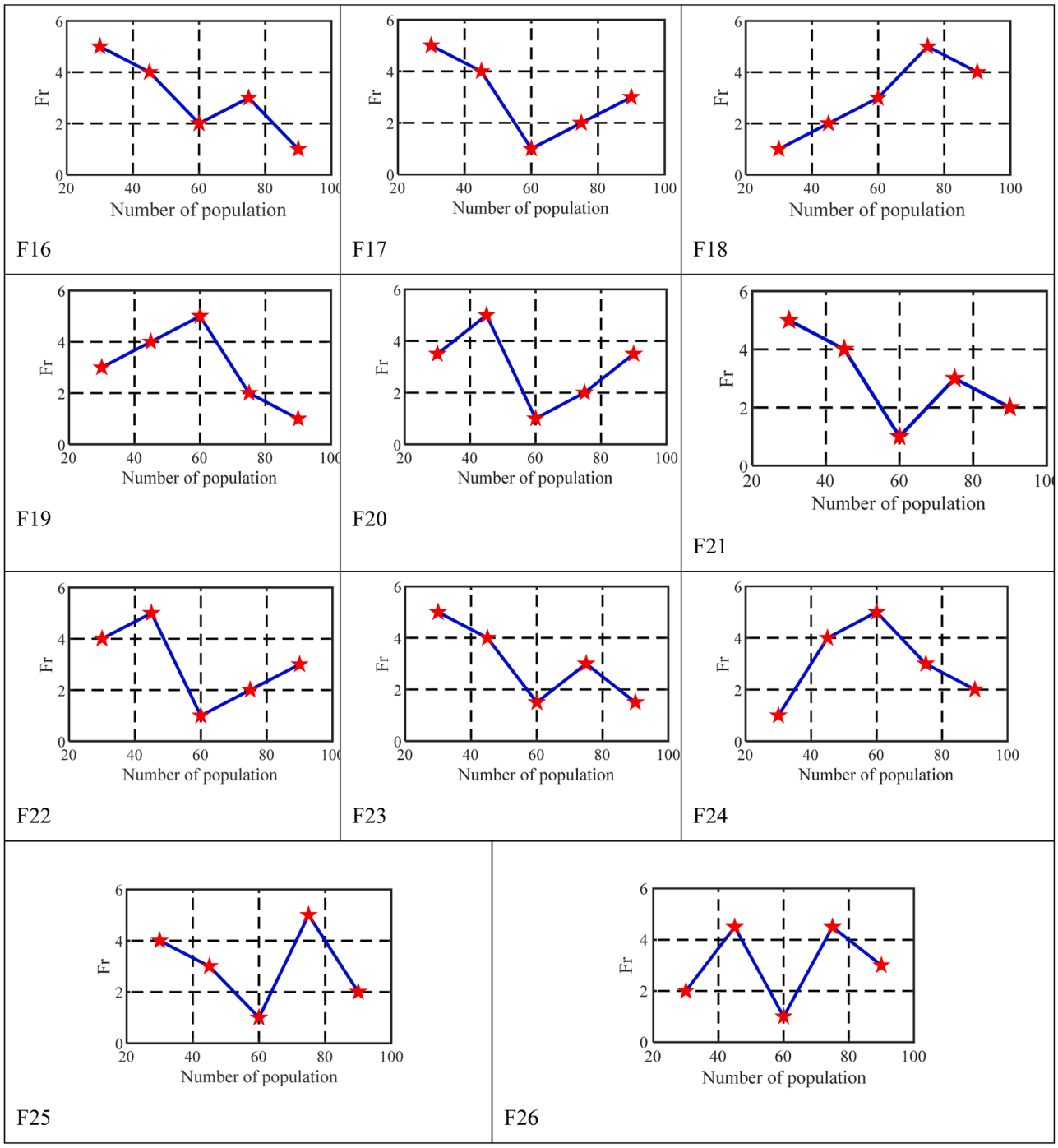


Fig. 4. (*continued*).

optimization issues, this part confirms how well it performs in real-world applications. The restrictions of the optimization problem are handled in this study using the static penalty approach [53]:

$$\theta(X) = f(X) \pm \left[\sum_{i=1}^{m} l_i \cdot \max(0, t_i(X))^{\alpha} + \sum_{j=1}^{n} o_j \cdot \left| U_j(X) \right|^{\beta} \right] \tag{9}$$

where $\theta(X)$ means objective function, o_j and l_j show two enormous penalty numbers. $U_j(X)$ and $t_i(X)$ show problem limits. The given exponents α and β are from the set$\{1,2\}$.

4.1. Three-bar truss problem

This problem's challenge is creating a truss with three bars that are as light as possible [26]. Stress, deflection, and buckling are three optimization constraints that are applied to two optimization variables in this test. It is put together as depicted in (10). According to [26], this problem has a very limited search space. Fig. 10 depicts the three-bar truss' overall structure.

Minimize:

$$f(X) = l \cdot \left(x_2 + 2\sqrt{2}\, x_1 \right). \tag{10}$$

by considering:

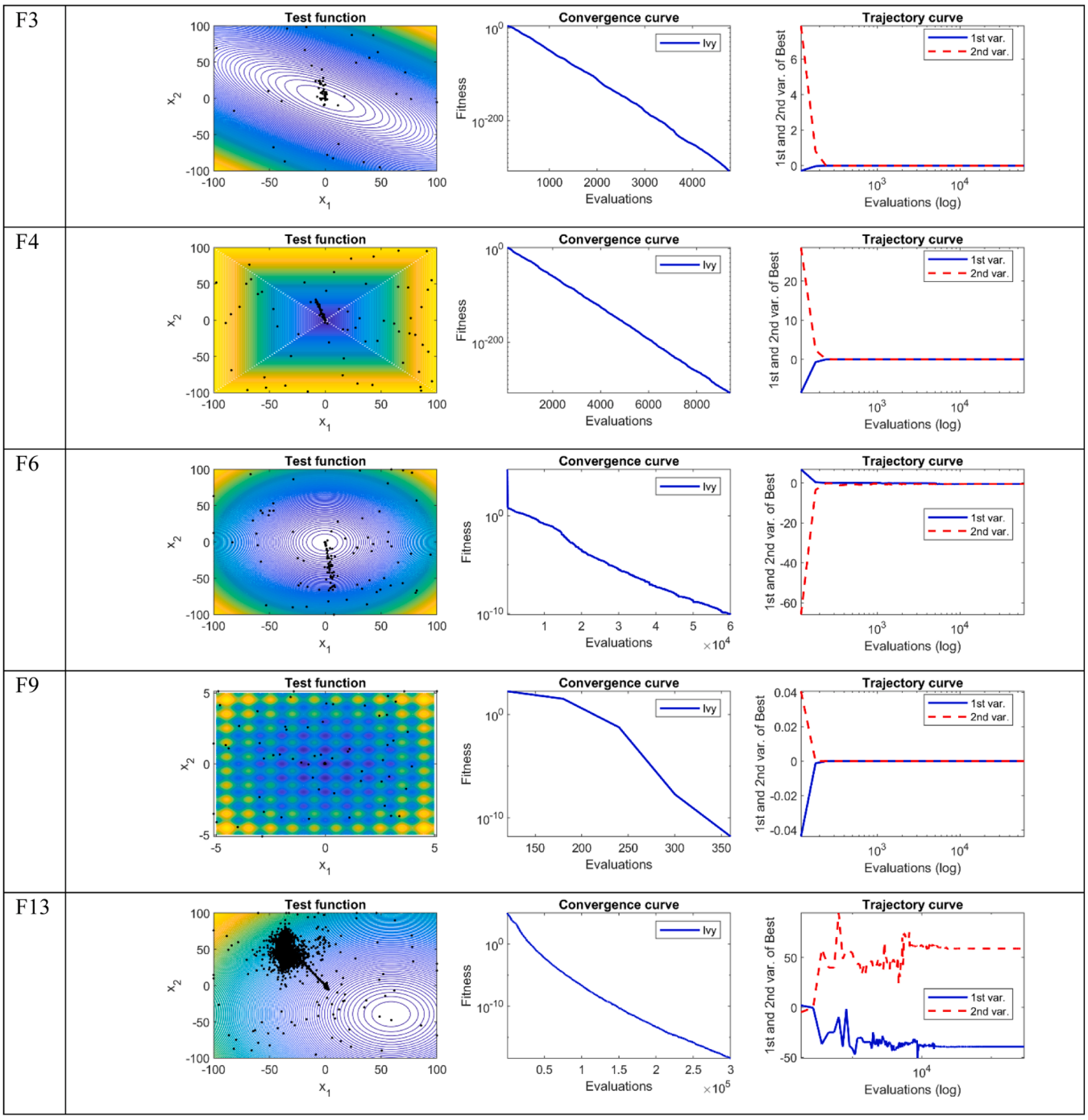


Fig. 5. Simulation results for the studied functions.

$$g_1(X) = P\cdot\left(x_2 + \sqrt{2}\,x_1\right)\left(2x_1x_2 + \sqrt{2}\,x_1^2\right)^{-1} - \sigma \le 0, \tag{11}$$

$$\begin{aligned} g_2(X) &= P\cdot\left(2x_1x_2 + \sqrt{2}x_1^2\right)^{-1}x_2 - \sigma \le 0, \\ g_3(X) &= P\cdot\left(x_1 + \sqrt{2}x_2\right)^{-1} - \sigma \le 0, \end{aligned} \tag{12}$$

where $0 \le x_1, x_2 \le 1$ and with values of constants: $l = 100$ cm, $\sigma = 2\,\frac{\text{kN}}{\text{cm}^2}$, $P = 2\,\frac{\text{kN}}{\text{cm}^2}$.

The IVYA findings are now included in Table 13, updated with data from recent studies. The mean of results are shown in Fig. 11, along with a convergence map for the number of function evaluations NFEs ∈ [1, 40000]. Firefly algorithm (FA) [177], memory-based grey wolf optimizer (mGWO) [178], Levy-flight FA (LF-FA) [177], sailfish optimizer (SFO) [26], society and civilization algorithm (SCA) [179], LSHADE with an improved ε constraint-handling method (iLSHADEε) [107], DE with Q-learning (DE-QL) [181], the ensemble of constraint handling techniques based on the voting-mechanism (VMCH) [182], an improved unified DE (IUDE) [107], and a matrix adaptation evolution strategy (εMAgES) [107] are a few of the algorithms selected for comparison. The comparison with the algorithms mentioned above demonstrates that the IVYA outperforms the others, offers highly competitive, and obtained similarly excellent results as SFO, SBO,

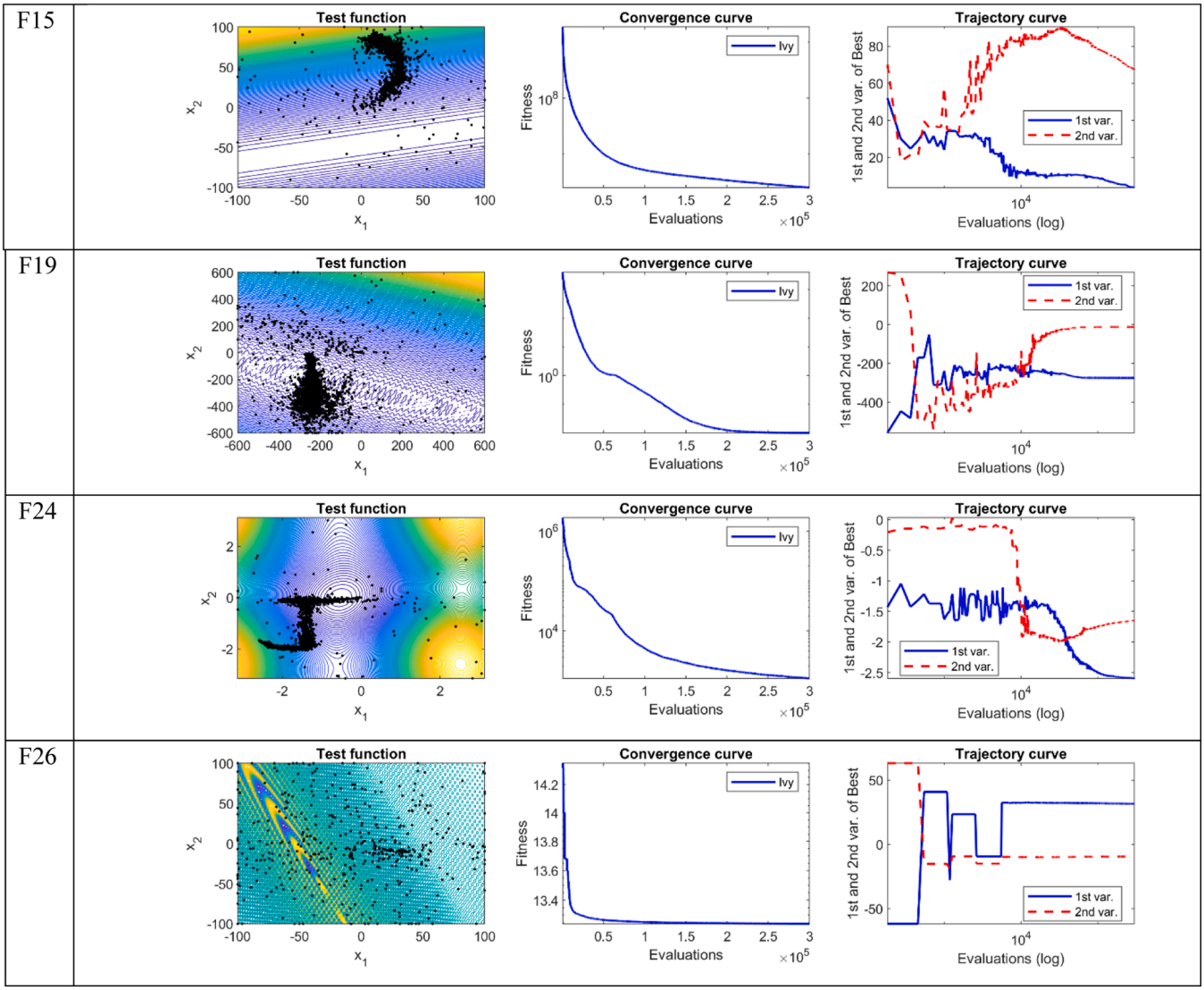


Fig. 5. (*continued*).

iLSHADEε, DE-QL, VMCH, IUDE, and εMAgES. Table 14 summarizes the values of all variables and constraints of this problem and the best solution values found by IVYA.

4.2. Design of a cantilever beam

Fig. 12 illustrates a cantilever beam constructed using five square hollow blocks. Weight reduction is the goal [183]. A cantilever's side length is represented by one of the five optimization variables in this problem, which also has an optimization constraint [184]. The following equation represents the cantilever weight optimization [185]:

Minimize:

$$f(X) = 0.0624 \cdot \sum_{j=1}^{5} x_j \tag{13}$$

by considering:

$$g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0, \tag{14}$$

where

$0.01 \leq x_j \leq 100$ for $j = 1, 2, 3, 4, 5$.

Since a solution to this problem has already appeared in several previous articles, we have selected the following optimizers for our comparison: WOA [186], grasshopper optimization algorithm (GOA) [184], SCA [186], hybrid self-adaptive SCA (m-SCA) [187], FA [177], moth-flame optimization (MFO) [33], and an adaptive logarithmic spiral-Levy FA (AD-IFA) [60]. Table 15 provides outcomes obtained by IVYA and their comparison with the aforementioned optimizers. Be aware that the IVYA performed much better for this problem than other optimizers while finding the lowest weight of the cantilever beam and obtaining outcomes that were quite competitive with the other strategies.

The convergence chart in Fig. 13 displays the mean solution value derived by the IVYA. Table 16 exhibits the variables, constraints, and the best solution values discovered by IVYA for this particular problem.

4.3. Welded beam design

The practical design of a welded beam is the problem frequently used to compare various optimization techniques [104]. As depicted in Fig. 14, the problem is to identify the ideal parameter to reduce manufacturing costs [188]. The control variables include the bar height t (x_3), the weld thickness h (x_1), the thickness of bar b (x_4), and the length of clamped bar l (x_2). The following limitations apply to the reduction

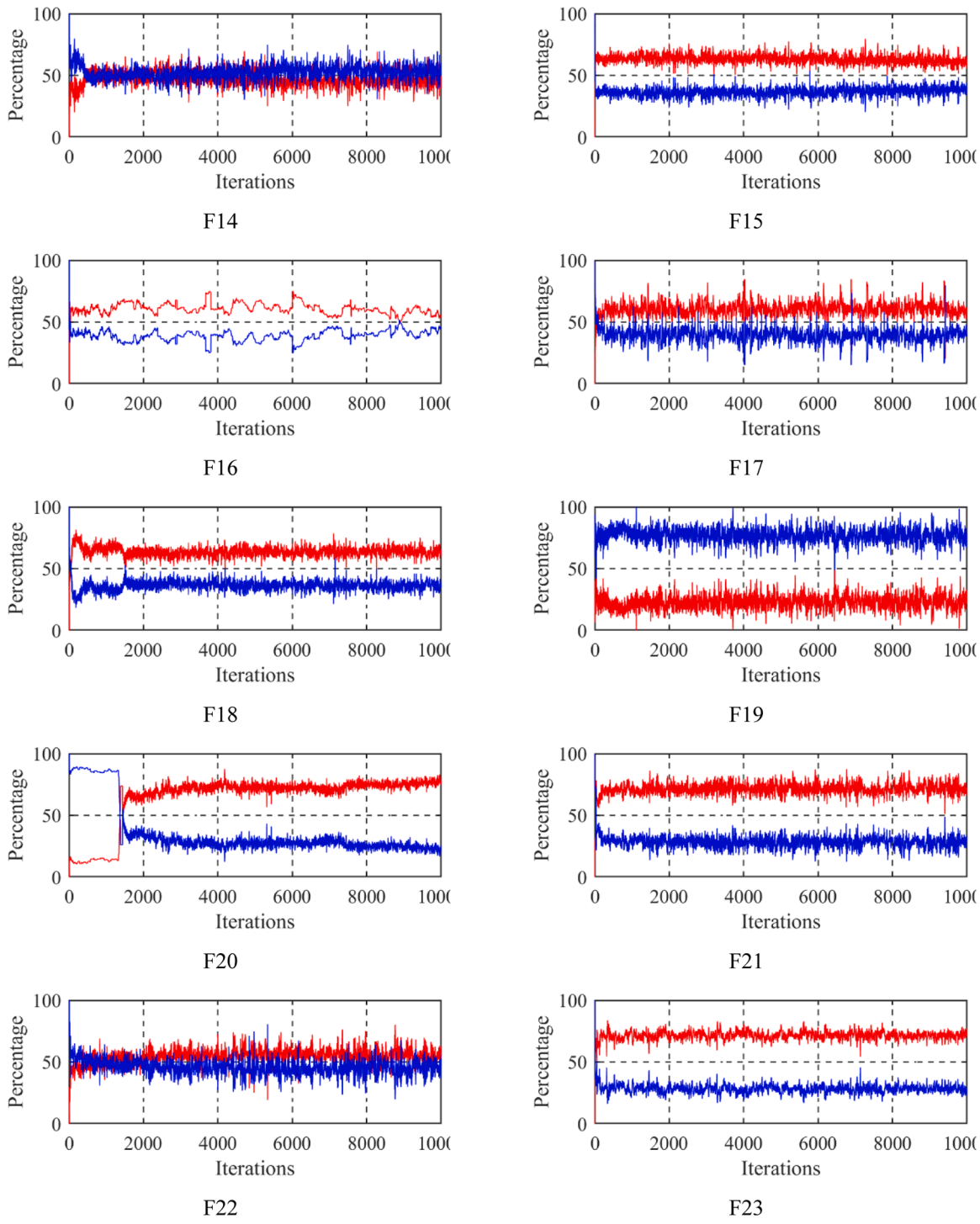


Fig. 6. Exploration-exploration of IVYA.

process: the deflection of the beam δ, the buckling load on the bar P_c, the bending stress in the beam σ, and the shear stress τ. A detailed description of the model is shown below [131,189]:

Minimize:

$$f(X) = 1.10471x_2x_1^2 + 0.04811x_3x_4(14 + x_2) \tag{15}$$

by considering:

$$\begin{aligned} g_1(X) &= \tau(x) - \tau_{\max}, \\ g_2(X) &= -\sigma_{\max} + \sigma(x), \\ g_3(X) &= -x_4 + x_1 \le 0, \end{aligned} \tag{16}$$

$$g_4(X) = 0.04811x_3x_4(14 + x_2) + 0.10471x_1^2 - 5 \le 0, \tag{17}$$

$$\begin{aligned} g_5(X) &= -x_1 + 0.125 \le 0, \\ g_6(X) &= \delta(x) - \delta_{\max} \le 0, \\ g_7(X) &= -P_c(x) + P \le 0, \end{aligned} \tag{18}$$

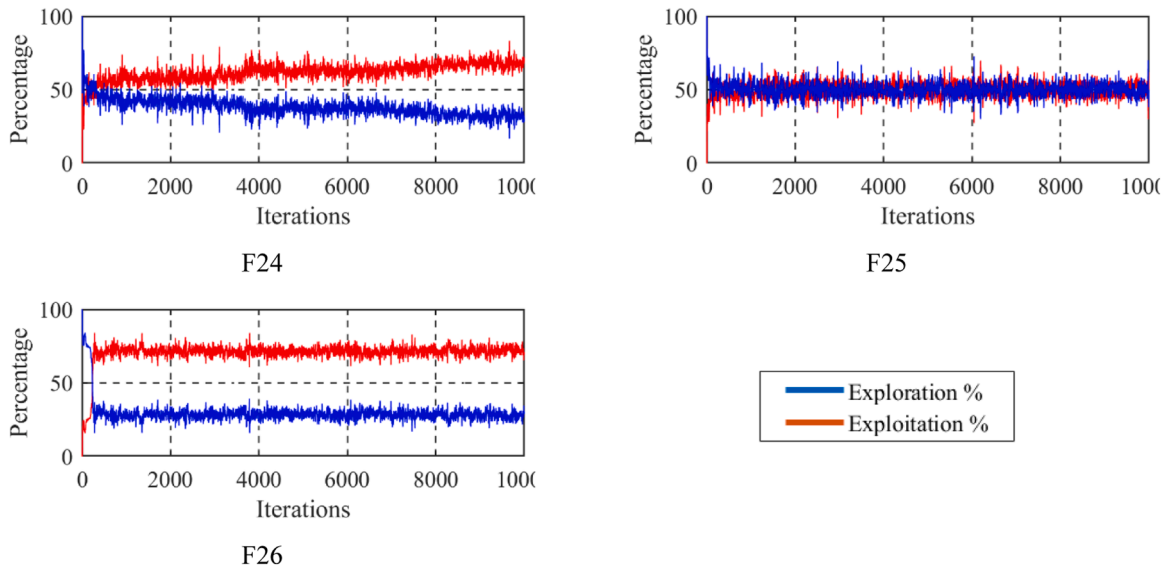


Fig. 6. (*continued*).

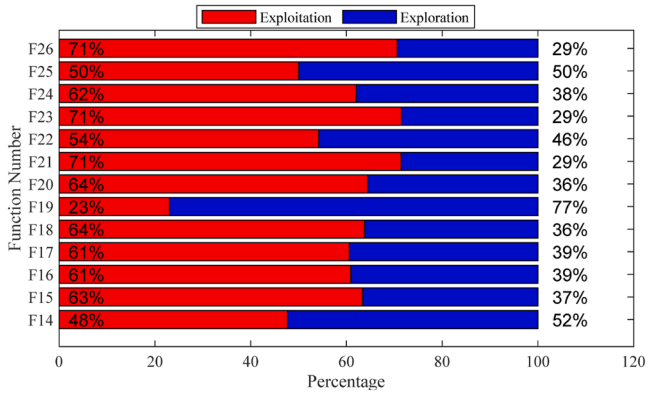


Fig. 7. Mean values for exploration and exploitation of test functions F14 to F26 by IVYA.

where

$$\tau(x)=\sqrt{(\tau')^2+2\tau'\tau''\frac{x_2}{2R}+(\tau'')^2},$$
$$\tau'=\frac{P}{\sqrt{2}x_1x_2},$$
$$\tau''=\frac{MR}{J}, \tag{19}$$

$$M=P\cdot\left(L+\frac{x_2}{2}\right),$$
$$R=\sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2}, \tag{20}$$
$$\delta(x)=\frac{4P\ L^3}{E\ x_3^3x_4},$$

$$J=2\sqrt{2}x_1x_2\left(\frac{x_2^2}{12}+\left(\frac{x_1+x_3}{2}\right)^2\right),$$
$$\sigma(x)=\frac{6P\ L}{x_4x_3^2}, \tag{21}$$
$$P_c(x)=\frac{4.013E\ x_4^3x_3}{6L^2}\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

with

$0.1\le x_2,x_3\le 10$, $0.1\le x_1,x_4\le 2$ and with values of constants:

$\sigma_{max}=30{,}000, E=30{\cdot}10^6$ psi, $\tau_{\max}=13{,}000$ psi, $L=14$ in, $P=6{,}000$ lb, $\delta_{max}=0.25$ in, and $G=12{\cdot}10^6$ psi.

In the design of welded beam, the IVYA is compared with GWO [163], hybrid real-parameter GA (HSA-GA) [190], RAER [191], AD-IFA [177], human mental search (HMS) [192], SFO [26], filter SA algorithm (FSA) [193], unified PSO (UPSO) [194], bacterial foraging method (BFOA) [195], improved PSO (IPSO) [196] algorithms, and Gaussian backbone-based spherical evolutionary with cross-search (CGSE) [197]. Keep in mind that Table 17 shows the results of all calculations. The IVYA's welded beam design had an ideal cost of 1.724867. Table 17

Table 9
Settings of parameters of all used optimizers.

Algorithm	Parameter = Value
AO [169]	$G_1\in[-1,1]$; $G_2=[2,0]$, $\delta=0.1$; $\alpha=0.1$; $\omega=0.005$; $r_1=10$; $u=0.00565$; $Npop=60$
AOA [170]	$a=5$; $\mu=0.5$, $Npop=60$
PSO [171]	$w_{initial}=1$, $w_{damp}=0.99$, $w_{Iter+1}=w_{damp}{\cdot}w_{Iter}$, $c_1=1.5$, $c_2=2.0$, $Npop=60$, $Npop=60$
BBO [172]	$hmp=1$, $stSize=1$, $elit=2$, $mir=1$, $mer=1$, $mt\ge 0.005$, $Npop=60$
SCA [9]	$A=2$, $Npop=60$
RSA [173]	$\beta=0.1$, $\alpha=0.005$, $Npop=60$
Rao-1 [174]	$Npop=60$
JAYA [23]	$Npop=60$
GWO [163]	$a=2$ to 0, $Npop=60$
WOA [1,175]	$A=2$ to 0, $b=1$, $l\in[-1,1]$, $Npop=60$

Table 10
Summary of indicators Mean, Winner, and Std. results for the classic test functions.

Function	Rao-1 Mean Winner Std.	JAYA Mean Winner Std.	AO Mean Winner Std.	PSO Mean Winner Std.	AOA Mean Winner Std.	BBO Mean Winner Std.	GWO Mean Winner Std.	RSA Mean Winner Std.	SCA Mean Winner Std.	WOA Mean Winner Std.	IVYA Mean Std.
F1	2.85E-02 − 8.20E-03	1.53E-01 − 5.78E-02	1.50E-294 − 0	1.17E + 01 − 4.53	9.79E-204 − 0	5.57E-01 − 0.0141	1.54E-74 − 3.40E-74	0 = 0	1.75E-03 − 6.89E-03	6.81E-175 − 0	0 0
F2	8.24E-01 − 2.35	2.80E-01 − 6.27E-02	1.21E-150 − 2.99E-150	6.76E-01 − 2.14E-01	0 = 0	2.23E-01 − 2.47E-02	4.25E-43 − 5.38E-43	0 = 0	5.28E-06 − 1.49E-05	5.31E-110 − 1.60E-109	0 0
F3	7.55E + 03 − 2.87E + 03	3.36E + 04 − 5.26E + 03	2.73E-220 − 0	603.0 − 155.0	7.60E-04 − 3.40E-03	72.0 − 25.1	9.29E-23 − 1.66E-22	0 = 0	2.44E + 03 − 2.86E + 03	7.34E + 03 − 5.08E + 03	0 0
F4	2.29 − 0.583	10.9 − 3.04	5.19E-117 − 2.32E-116	5.27 − 0.711	9.84E-03 − 1.75E-02	0.689 − 6.61E-02	1.33E-18 − 1.45E-18	0 = 0	9.37 − 7.17	23.6 − 29.5	0 0
F5	69.3 − 60.1	109.0 − 80.1	4.73E-04 + 9.12E-04	355.0 − 145.0	27.9 − 0.628	85.9 − 50.4	26.5 − 0.984	5.8 + 2.01	46.9 − 60.4	26.4 − 0.283	25.0 0.75
F6	2.87E-02 − 9.05E-03	5.54E + 00 − 7.73E-01	6.71E-06 + 1.78E-05	9.53E + 00 − 3.62E + 00	2.33E + 00 − 2.71E-01	5.13E-01 − 9.82E-02	2.87E-01 − 2.32E-01	7.14E + 00 − 3.44E-01	4.15E + 00 − 4.18E-01	2.06E-03 + 9.89E-04	2.53E-02 2.71E-02
F7	6.24E-02 − 2.17E-02	7.90E-02 − 2.26E-02	3.05E-05 − 2.99E-05	1.77E-02 − 5.48E-03	1.67E-05 − 1.67E-05	3.45E-03 − 8.02E-04	4.18E-04 − 1.82E-04	3.15E-05 − 2.70E-05	1.70E-02 − 1.63E-02	5.29E-04 − 6.25E-04	1.46E-05 1.23E-05
F8	−5.37E + 03 + 6.02E + 02	−5.41E + 03 + 6.44E + 02	−8.62E + 03 + 3.39E + 03	−6.70E + 03 + 6.54E + 02	−6.46E + 03 + 3.52E + 02	−8.71E + 03 + 6.32E + 02	−6.31E + 03 + 8.68E + 02	−5.46E + 03 + 7.16E + 01	−4.17E + 03 + 3.19E + 02	-1.17E + 04 + 1.43E + 03	−3.6E + 03 6.6E + 02
F9	2.39E + 02 − 2.37E + 01	2.33E + 02 − 1.71E + 01	0 = 0	4.30E + 01 − 1.49E + 01	0 = 0	3.34E + 01 − 7.35E + 00	0 = 0	0 = 0	1.14E + 01 − 1.79E + 01	0 = 0	0 0
F10	3.07E + 00 − 6.96E + 00	1.66E + 00 − 3.61E + 00	8.88E-16 = 0	1.8046 − 6.11E-01	8.88E-16 = 0	2.16E-01 − 1.25E-04	1.28E-14 − 2.89E-15	8.88E-16 = 0	1.01E + 01 − 1.03E + 01	3.91E-15 − 2.89E-15	8.9E-16 0
F11	3.49E-01 − 1.96E-01	6.18E-01 − 1.70E-01	0 = 0	1.10E + 00 − 6.51E-02	6.01E-02 − 5.25E-02	4.94E-01 − 9.09E-02	5.32E-04 − 2.38E-03	0 = 0	1.42E-01 − 1.99E-01	7.76E-03 − 2.45E-02	0 0
F12	2.41E + 00 − 2.48E + 00	5.62E + 00 − 2.02E + 00	4.29E-07 + 1.10E-06	6.69E-02 − 1.01E-01	2.90E-01 − 4.76E-02	1.14E-03 − 3.39E-04	1.91E-02 − 1.01E-02	1.52E + 00 − 2.12E-01	7.60E-01 − 4.40E-01	4.40E-04 + 6.65E-04	1.11E-03 1.57E-03
F13	1.18E-27 + 6.60E-28	4.67E + 03 − 1.37E + 03	2.32E + 00 − 6.73E-01	5.34E-02 − 5.75E-02	5.20E + 04 − 7.74E + 03	1.78E-02 − 5.59E-03	1.03E + 03 − 1.42E + 03	5.62E + 04 − 1.18E + 04	1.12E + 04 − 1.66E + 03	4.12E-01 − 2.31E-01	1.31E-23 5.89E-23
F14	2.06E + 00 − 2.28E + 00	1.70E + 04 − 3.99E + 03	2.17E + 03 − 1.05E + 03	2.09E + 02 − 4.51E + 01	3.96E + 04 − 1.38E + 04	9.73E + 00 − 2.32E + 00	1.03E + 04 − 3.23E + 03	5.67E + 04 − 1.74E + 04	1.82E + 04 − 2.11E + 03	5.33E + 04 − 6.82E + 03	1.85E-02 2.59E-02
F15	2.83E + 07 − 9.99E + 06	9.00E + 07 − 2.64E + 07	1.20E + 07 − 3.73E + 06	8.82E + 06 − 3.00E + 06	6.74E + 08 − 2.48E + 08	2.54E + 06 − 6.59E + 05	1.54E + 07 − 1.11E + 07	5.06E + 08 − 1.82E + 08	1.41E + 08 − 5.13E + 07	2.05E + 07 − 9.52E + 06	1.5E + 06 3.8E + 05
F16	3.85E + 02 + 4.72E + 02	2.52E + 04 − 4.00E + 03	1.30E + 04 − 3.13E + 03	5.42E + 02 + 1.38E + 02	4.85E + 04 − 1.52E + 04	4.96E + 01 + 1.84E + 01	1.17E + 04 − 3.99E + 03	5.72E + 04 − 1.62E + 04	2.38E + 04 − 3.98E + 03	1.36E + 05 − 3.75E + 04	1.0E + 04 4.7E + 03
F17	1.96E + 03 + 1.54E + 03	6.59E + 03 − 1.40E + 03	1.08E + 04 − 2.06E + 03	3.48E + 03 − 5.07E + 02	3.21E + 04 − 4.63E + 03	3.93E + 03 − 6.86E + 02	4.05E + 03 − 2.99E + 03	3.14E + 04 − 6.45E + 03	1.57E + 04 − 2.09E + 03	1.70E + 04 − 3.88E + 03	2.6E + 03 2.6E + 03
F18	6.93E + 01 + 1.12E + 02	1.71E + 08 − 6.96E + 07	2.70E + 03 − 4.09E + 03	1.74E + 03 − 2.36E + 03	1.90E + 10 − 3.12E + 09	2.01E + 03 − 2.98E + 03	3.48E + 07 − 2.74E + 07	2.04E + 10 − 6.18E + 09	1.10E + 09 − 4.02E + 08	1.19E + 04 − 7.63E + 03	8.2E + 02 4.1E + 02
F19	8.41E-02 −	3.59E + 02 −	2.57E + 01 −	1.33E + 00 −	1.34E + 03 −	8.92E-01 −	5.77E + 01 −	1.84E + 03 −	5.59E + 02 −	1.91E + 00 −	2.39E-02

(*continued on next page*)

Table 10 (*continued*)

Function	Rao-1 Mean Winner Std.	JAYA Mean Winner Std.	AO Mean Winner Std.	PSO Mean Winner Std.	AOA Mean Winner Std.	BBO Mean Winner Std.	GWO Mean Winner Std.	RSA Mean Winner Std.	SCA Mean Winner Std.	WOA Mean Winner Std.	IVYA Mean Std.
	1.71E-01	4.57E + 01	7.93E + 00	1.93E-01	2.52E + 02	8.81E-02	4.72E + 01	4.22E + 02	9.90E + 01	6.66E-01	9.29E-03
F20	2.10E + 01	2.09E + 01	2.10E + 01	2.09E + 01	2.10E + 01	2.07E + 01	2.09E + 01	2.09E + 01	2.09E + 01	2.07E + 01	2.0E + 01
	−	−	−	−	−	−	−	−	−	−	
	4.49E-02	6.27E-02	5.41E-02	8.27E-02	4.93E-02	7.78E-02	4.41E-02	5.49E-02	4.81E-02	4.95E-02	3.5E-02
F21	1.84E + 02	2.23E + 02	1.37E + 02	9.31E + 01	3.17E + 02	3.62E + 01	8.49E + 01	3.22E + 02	2.42E + 02	2.12E + 02	1.2E + 02
	−	−	−	−	−	+	+	−	−	−	
	2.61E + 01	1.24E + 01	2.00E + 01	1.76E + 01	2.40E + 01	9.95E + 00	2.29E + 01	1.87E + 01	1.81E + 01	3.19E + 01	3.1E + 01
F22	2.25E + 02	2.75E + 02	2.30E + 02	1.42E + 02	5.15E + 02	7.31E + 01	1.17E + 02	5.14E + 02	3.30E + 02	4.55E + 02	1.0E + 02
	−	−	−	−	−	+	−	−	−	−	
	1.97E + 01	1.55E + 01	5.47E + 01	4.53E + 01	4.25E + 01	1.71E + 01	5.87E + 01	3.15E + 01	2.39E + 01	1.45E + 02	7.9E + 01
F23	3.93E + 01	3.90E + 01	3.83E + 01	1.75E + 01	4.07E + 01	2.48E + 01	1.62E + 01	3.96E + 01	3.94E + 01	3.67E + 01	1.6E + 01
	−	−	−	−	−	−	−	−	−	−	
	8.56E-01	1.35E + 00	4.32E + 00	2.80E + 00	1.92E + 00	4.62E + 00	1.77E + 00	1.08E + 00	9.91E-01	2.60E + 00	4.3E + 00
F24	8.35E + 04	3.60E + 05	9.22E + 04	2.52E + 04	9.06E + 05	1.49E + 04	8.61E + 04	7.39E + 05	4.56E + 05	1.04E + 05	2.0E + 04
	−	−	−	−	−	−	−	−	−	−	
	7.25E + 04	1.28E + 05	2.94E + 04	8.56E + 03	1.35E + 05	1.31E + 04	2.94E + 04	1.13E + 05	9.88E + 04	7.21E + 04	7.9E + 03
F25	1.61E + 01	3.77E + 01	1.85E + 01	5.95E + 00	6.08E + 01	2.20E + 00	6.02E + 00	1.51E + 02	3.19E + 01	2.17E + 01	5.9E + 00
	−	−	−	−	−	−	+	−	−	−	
	1.69E + 00	4.86E + 00	2.52E + 00	1.93E + 00	1.08E + 01	2.92E-01	4.04E + 00	4.80E + 01	6.02E + 00	6.31E + 00	1.6E + 00
F26	1.34E + 01	1.35E + 01	1.27E + 01	1.23E + 01	1.31E + 01	1.28E + 01	1.18E + 01	1.33E + 01	1.35E + 01	1.32E + 01	1.2E + 01
	−	−	−	−	−	−	+	−	−	−	
	1.72E-01	1.63E-01	3.61E-01	6.34E-01	2.49E-01	4.36E-01	6.81E-01	1.73E-01	2.19E-01	3.79E-01	6.3E-01
+ / − / =	7/**19**/0	2/**24**/0	5/**17**/4	8/**18**/0	1/**22**/3	8/**18**/0	8/**23**/1	2/**17**/7	1/**25**/0	2/**23**/1	—

Table 11
Results of the Friedman test, which were conducted on all compared algorithms across 26 benchmark functions.

Function	Rao-1	JAYA	AO	PSO	AOA	BBO	GWO	RSA	SCA	WOA	IVYA
F1	8	9	3	11	4	10	6	1.5	7	5	1.5
F2	11	9	4	10	2	8	6	2	7	5	2
F3	10	11	3	7	5	6	4	1.5	8	9	1.5
F4	7	10	3	8	5	6	4	1.5	9	11	1.5
F5	8	10	1	11	6	9	5	2	7	4	3
F6	4	9	1	11	7	6	5	10	8	2	3
F7	10	11	3	9	2	7	5	4	8	6	1
F8	9	8	3	4	5	2	6	7	10	1	11
F9	11	10	3.5	9	3.5	8	3.5	3.5	7	3.5	3.5
F10	10	8	2.5	9	2.5	7	6	2.5	11	5	2.5
F11	8	10	2	11	6	9	4	2	7	5	2
F12	10	11	1	6	7	4	5	9	8	2	3
F13	1	8	6	4	10	3	7	11	9	5	2
F14	2	7	5	4	9	3	6	11	8	10	1
F15	7	8	4	3	11	2	5	10	9	6	1
F16	2	8	6	3	9	1	5	10	7	11	4
F17	1	6	7	3	11	4	5	10	8	9	2
F18	1	8	5	3	10	4	7	11	9	6	2
F19	2	8	6	4	10	3	7	11	9	5	1
F20	10	6	10	6	10	2.5	6	6	6	2.5	1
F21	6	8	5	3	10	1	2	11	9	7	4
F22	5	7	6	4	11	1	3	10	8	9	2
F23	8	7	6	3	11	4	2	10	9	5	1
F24	4	8	6	3	11	1	5	10	9	7	2
F25	5	9	6	3	10	1	4	11	8	7	2
F26	9	10.5	4	3	6	5	1	8	10.5	7	2
MFr	6.5000	8.6346	4.3077	5.9615	7.4615	4.5192	4.7885	7.1731	8.2885	5.9615	**2.4038**
Final rank	7	11	2	5.5	9	3	4	8	10	5.5	**1**

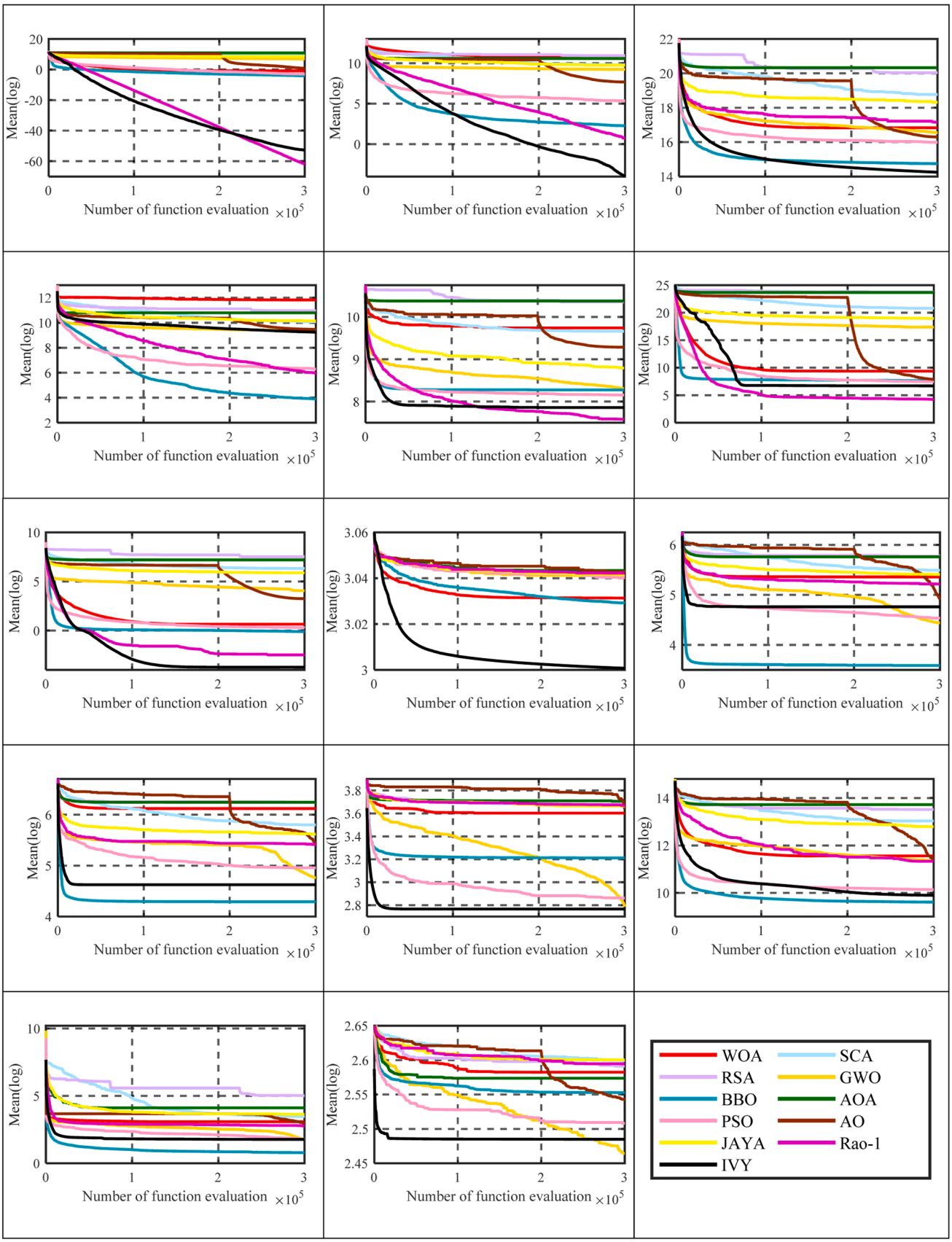


Fig. 8. Convergence analysis of optimization algorithms applied to functions F13 through F26 for NFEs $\in [1,\ 3\times 10^5]$.

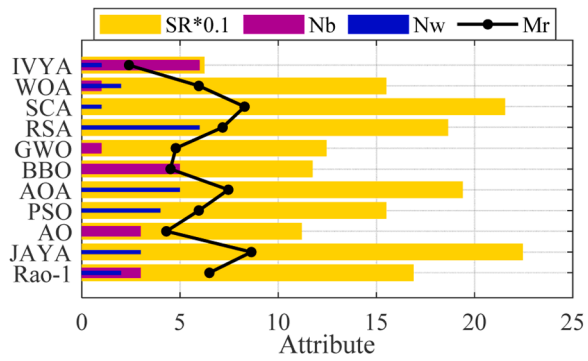


Fig. 9. The statistical analysis of IVYA and other algorithms on 26 benchmark functions.

shows that the IVYA produces better results compared to the other algorithms we considered. The lowest cost and accompanying ideal variable value, as determined by the algorithm mentioned above, are shown in Table 18. A convergence graph of the mean solutions for the problem of constructing welded beams found by IVYA is shown in Fig. 15.

4.4. Design of the gear train problem

Optimizing gear trains minimizes the transmission ratio value while adhering to specified constraints. This issue is a well-known engineering design challenge. As illustrated in Fig. 16, this problem has four key parameters: n_A (denoted as x_1), n_B (denoted as x_2), n_D (denoted as x_3), and n_F (denoted as x_4),

The mathematical model has the following form [194].

Minimize

$$f(X) = \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4}\right)^2, \qquad (22)$$

$$12 \le x_1, x_2, x_3, x_4 \le 60.$$

Fig. 17 depicts the convergence tendency of IVYA. The experimental outcomes for IVYA and several other well-known swarm intelligence optimizers are shown in Table 19. The experiment uses CSA [25], GOA [198], UPSO [194], memory-based GWO (mGWO) [178], and escaping bird search (BES) [198] as comparative algorithms. According to experimental data, the IVYA performs somewhat worse than the CSA method for the Mean, Worst, and Std. indices. Still, it is superior to the GOA and BES optimizers for designing the gear train. The IVYA continues to outperform all comparison algorithms. Table 20 contains the solution to the problem's best-case scenarios.

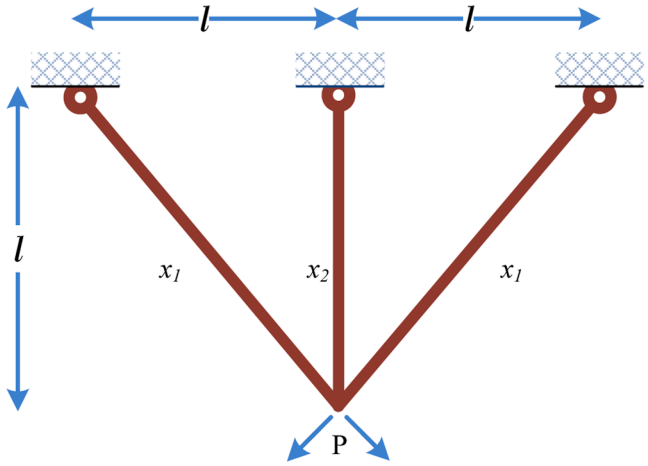


Fig. 10. The schema of a three-bar truss.

Table 13
The list of optimal outcomes that were achieved for the three-bar truss problem.

Optimizers	Best	Mean	Worst	Std.
LF-FA [177]	282.84	283.20	N.A.	1.10
SCA [179]	**263.89584**	**263.90336**	263.96976	1.3E − 02
mGWO [178]	263.8961	N.A.	N.A.	N.A.
SFO [26]	263.89592	N.A.	N.A.	N.A.
FA [177]	282.84	287.84	N.A.	4.94
iLSHADEε [107]	**2.64E + 02**	**2.64E + 02**	2.65E + 02	4.47E-01
DE-QL [181]	**2.638958E + 02**	**2.638958E + 02**	**2.638958E + 02**	1.640928E−14
VMCH [182]	**2.64E + 02**	**2.64E + 02**	**2.64E + 02**	1.16E-14
IUDE [107]	**2.64E + 02**	**2.64E + 02**	**2.64E + 02**	**0.00E + 00**
εMAgES [107]	**2.64E + 02**	**2.64E + 02**	**2.64E + 02**	**0.00E + 00**
IVYA	**263.89584**	263.90581	**263.9460**	0.0126

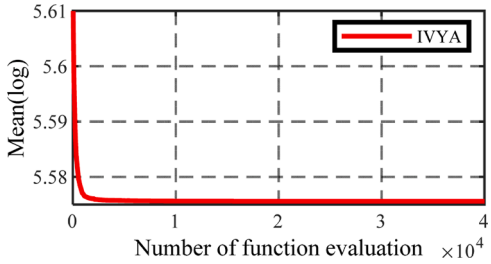


Fig. 11. The three-bar truss problem's convergence plot obtained via the IVYA (NFEs $\in [1, 40000]$).

Table 12
The Wilcoxon test results between IVYA and other algorithms on 26 benchmark functions.

i	j	MoNR	MoPR	SNR	SPR	$F(i) < F(j)$	$F(j) < F(i)$	p-value	0.95 Confidence interval	
IVYA	Rao−1	12.52	17.60	263.00	88.00	21.00	5.00	0.03	−64.04	−0.05
	JAYA	13.32	18.00	333.00	18.00	25.00	1.00	0.00	−16,099.99	−42.14
	AO	12.84	11.20	244.00	56.00	19.00	5.00	0.01	−1501.16	−0.35
	PSO	12.52	21.00	288.00	63.00	23.00	3.00	0.00	−322.50	−0.76
	AOA	12.77	9.50	281.00	19.00	22.00	2.00	0.00	−26,207.50	−2.30
	BBO	11.50	20.17	230.00	121.00	20.00	6.00	0.17	−30.69	13.20
	GWO	12.61	15.83	277.50	47.50	22.00	3.00	0.00	−5149.99	−0.06
	RSA	11.12	7.00	189.00	21.00	17.00	3.00	0.00	−359,500.50	−73.00
	SCA	13.32	18.00	333.00	18.00	25.00	1.00	0.00	−9100.37	−13.00
	WOA	13.36	10.33	294.00	31.00	22.00	3.00	0.00	−26,649.99	−1.15

Here i, j are algorithm index, MoNR is the mean of all negative ranks, MoPR is the mean of all positive ranks, SNR is the sum of all negative ranks, SPR is the sum of all positive ranks, $F(\cdot)$ is the value of an objective function.

Table 14
The best outcomes for the three-bar truss problem.

Variables	IVYA
x_1	0.7887031
x_2	0.4081692
$g_1(X)$	−1.44037E-09
$g_2(X)$	−1.4642
$g_3(X)$	−0.5358
f_{Best}	**263.89584**

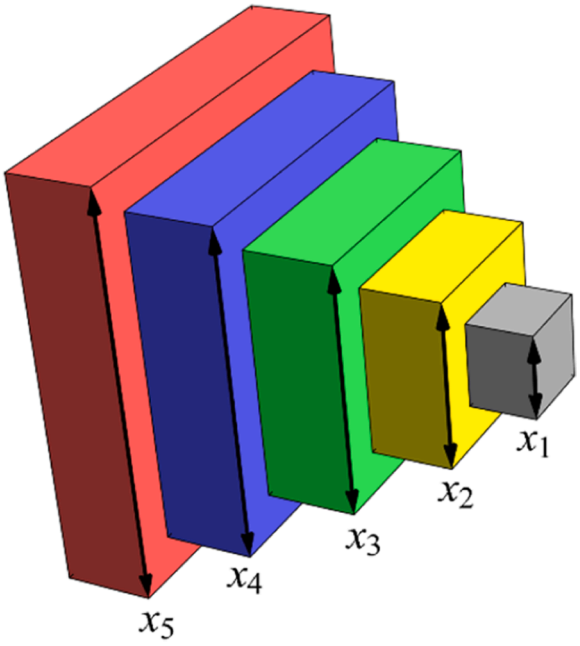


Fig. 12. The schema of the cantilever beam.

4.5. Pressure vessel optimization problem

Fig. 18 depicts this problem: a hemispherical head covers the cylindrical container's open end [199]. This problem aims to minimize the total cost of cylindrical pressure vessels, considering factors like welding, material selection, and forming processes. This problem has four control parameters, including the thickness of the shell T_s (denoted as x_1), the thickness of the head T_h (denoted as x_2), the length of the cylindrical shell L (denoted as x_4), and the inner radius R (denoted as x_3). The model is shown below [200]:

Minimize:

Table 15
The list of optimal outcomes that were achieved in the cantilever beam problem.

Optimizers	Best	Mean	Worst	Std.
SCA [186]	1.356647	1.382846	1.426982	0.017955
m-SCA [187]	1.33999	N.A.	N.A.	N.A.
GOA [184]	1.33996	N.A.	N.A.	N.A.
WOA [186]	1.347944	1.436524	1.664954	0.076076
MFO [33]	**1.33998**	N.A.	N.A.	N.A.
FA [177]	3.59	8.72	N.A.	1.93
AD-IFA [177]	**1.34**	1.54	N.A.	0.44
IVYA	**1.33996**	**1.33998**	**1.34003**	**1.86E-05**

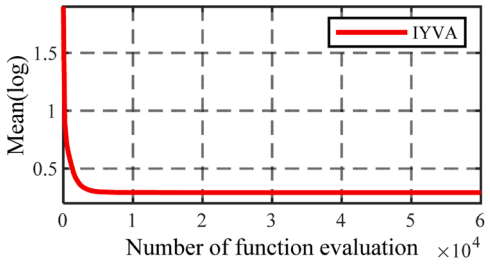


Fig. 13. The convergence graph for the cantilever beam problem obtained via the IVYA (NFEs $\in [1, 60000]$).

Table 16
The optimal outcomes of the cantilever beam design.

Variables	IVYA
x_1	6.015945
x_2	5.30941
x_3	4.49431
x_4	3.50119
x_5	2.15280
$g(X)$	−6.7998E-09
f_{Best}	1.33996

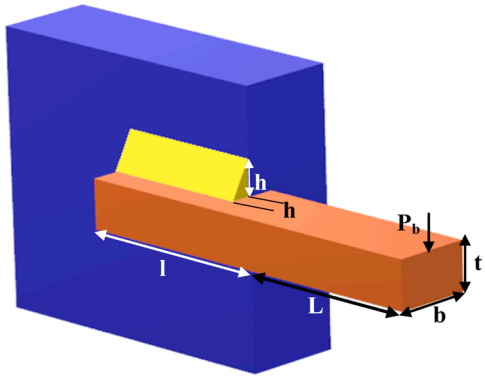


Fig. 14. The schema of the welded beam.

Table 17
The best solutions for the design of the welded beam.

Optimizers	Best	Mean	Worst	Std.
IPSO [196]	2.3810	2.3819	N.A.	5.23E-03
FSA [193]	2.3811	2.4041	2.4889	N.A.
UPSO [194]	1.92199	2.83721	N.A.	6.83E-01
RAER [191]	2.3816	N.A.	**2.38297**	**3.4E-04**
AD-IFA [177]	1.81	2.40	N.A.	0.50
HMS [192]	1.7255	N.A.	N.A.	N.A.
SFO [26]	1.73231	N.A.	N.A.	N.A.
GWO [163]	1.72624	N.A.	N.A.	N.A.
HSA-GA [190]	2.2500	2.26	2.28	7.80E-03
BFOA [195]	2.3868	2.4040	N.A.	1.60E-02
CGSE [197]	1.727201	N.A.	N.A.	N.A.
IVYA	**1.724867**	**1.944475**	2.474548	0.129

Table 18
The optimal outcomes for the welded beam design.

Variables	IVYA
x_1	0.2057288821
x_2	3.4704816304
x_3	9.0366881947
x_4	0.2057298829
$g_1(X)$	−0.0044
$g_2(X)$	−0.4623
$g_3(X)$	−1E-06
$g_4(X)$	−3.4329
$g_5(X)$	−0.08073
$g_2(X)$	−0.2355
$g_3(X)$	−0.04933
f_{Best}	1.724867

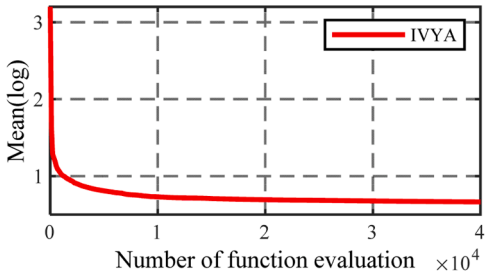


Fig. 15. The convergence plot for the problem obtained via the IVYA (NFEs $\in [1,\ 40000]$).

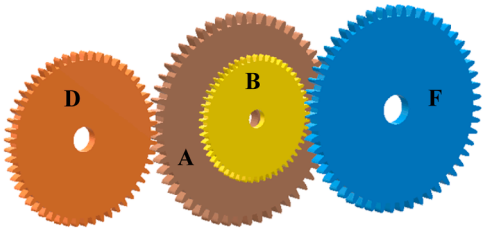


Fig. 16. The schema of the gear train.

$$f(X) = 19.84x_1^2x_3 + 3.1661x_1^2x_4 + 1.7781x_2x_3^2 + 0.6224x_1x_3x_4 \tag{23}$$

by considering:

$$\begin{aligned} g_1(X) &= -x_1 + 0.0193x_3 \le 0, \\ g_2(X) &= -x_2 + 0.00954x_3 \le 0, \end{aligned} \tag{24}$$

$$\begin{aligned} g_3(X) &= 1296000 - \pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 \le 0, \\ g_4(X) &= x_4 - 240 \le 0, \end{aligned} \tag{25}$$

where

$$10 \le x_i \le 200,\ i = 3,\ 4,\ 0 \le x_i \le 100,\ i = 1,\ 2.$$

In Table 21, the Ivy algorithm undergoes comparison with RFO [42], deer hunting optimization algorithm (DHOA) [201], UPSO [194], co-evolutionary PSO (CPSO) [202], HAIS-GA [203], NHAIS-GA [204], multiscale medalist learning algorithm (MMLA) [205], DE with

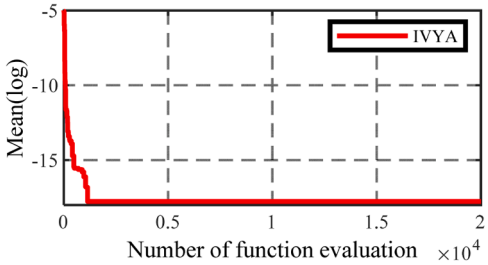


Fig. 17. The convergence plot for the gear train issue obtained via the IVYA (NFEs $\in [1, 20000]$).

Table 19
The list of optimal outcomes for the gear train design.

Optimizers	Best	Mean	Worst	Std.
GOA [198]	2.308E-11	2.096E-08	1.110E-07	2.596E-08
mGWO [178]	**2.7009E-12**	N.A.	N.A.	N.A.
UPSO [194]	**2.700857E-12**	3.80562E−8	N.A.	1.09E-7
BES [198]	1.166E-10	1.590E-03	4.563E-02	8.327E-03
CSA [25]	**2.700857E-12**	**2.059327E-9**	3.184738E-8	**5.0598E-9**
IVYA	**2.700857E-12**	7.92108E-09	**2.96991E-08**	1.816E-08

Table 20
The optimal outcomes for the construction of the gear train.

Variables	IVYA
x_1	43
x_2	19
x_3	16
x_4	49
f_{Best}	2.700857E-12

Q-learning (DE-QL) [181], the ensemble of constraint handling techniques based on the voting-mechanism (VMCH) [182], and EO [104] for this particular problem. Table 22 also includes the IVYA's minimum cost and accompanying ideal variable values. EO and IVYA yielded 6059.7143 as the ideal pressure vessel design cost. Fig. 19 displays the convergence trend of the mean of the findings obtained using the IVYA.

4.6. Speed reducer optimal design

The optimization challenge represented by the speed reducer design in Fig. 20 aims to lower the speed reducer's weight [206,207]. The control parameters are the face width b (denoted as x_1), the number of teeth on the pinion z (denoted as x_3), the module of teeth m (denoted as x_2), the length of the second shaft between bearings l_2 (denoted as x_5), the length of the first shaft between bearings l_1 (denoted as x_4), the second shaft's diameter d_2 (denoted as x_7), and the first shaft's diameter d_1 (denoted as x_6). The model is shown below [179,208]:

Minimize:

$$\begin{aligned} f(X) = {} & 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.5079x_1\left(x_6^2 + x_7^2\right) \\ & + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \end{aligned} \tag{26}$$

Subject to:

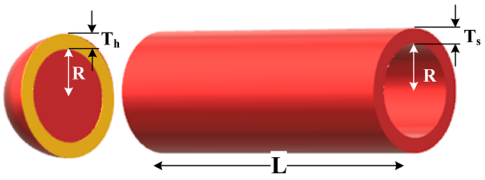


Fig. 18. The schema of the pressure vessel.

Table 21
The optimal solutions for the design of the pressure vessel.

Optimizers	Best	Mean	Worst	Std.
HAIS-GA [203]	6832.584	7187.314	8012.615	276
DHOA [201]	6103.842	N.A.	N.A.	N.A.
EO [104]	6059.7143	6668.114	7544.4925	566.24
CPSO [202]	6061.0777	6147.1332	6363.8041	86.4545
RFO [42]	6113.3195	N.A.	N.A.	N.A.
NHAIS-GA [204]	6061.1229	6743.0848	7368.0602	457.99
UPSO [194]	6154.70	8016.37	9387.77	745.869
MMLA [205]	6059.71433	6347.730290	6891.504402	188.3245
DE-QL [181]	8.077438E + 03	1.085138E + 04	1.355810E + 04	1.459232E + 03
VMCH [182]	6.07E + 03	**6.06E + 03**	**6.37E + 03**	**6.23E + 01**
IVYA	**6059.7143**	6286.8871	7544.4925	4.35E + 02

Table 22
The optimal results for the pressure vessel problem.

Variables	IVYA
x_1	0.8125
x_2	0.4375
x_3	42.09845
x_4	176.6366
$g_1(X)$	−1.13E−10
$g_2(X)$	−0.03588
$g_3(X)$	−2.789E-05
$g_4(X)$	−63.36336
f_{Best}	6059.7143

$$g_1(X) = -1 + \frac{27}{x_1 x_2^2 x_3} \leq 0, \quad g_2(X) = -1 + \frac{397.5}{x_1 x_2^2 x_3} \leq 0, \tag{27}$$

$$g_3(X) = -1 + \frac{1.93 x_4^3}{x_2 x_3 x_6^4} \leq 0, \quad g_4(X) = -1 + \frac{1.93 x_5^3}{x_2 x_3 x_7^4} \leq 0, \tag{28}$$

$$g_5(X) = \frac{1}{110 x_6^3} \sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9 \cdot 10^6} - 1 \leq 0, \tag{29}$$

$$g_6(X) = \frac{1}{85 x_7^3} \sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5 \cdot 10^6} - 1 \leq 0, \tag{30}$$

$$g_7(X) = -1 + \frac{x_2 x_3}{40} \leq 0, \quad g_8(X) = -1 + \frac{5 x_2}{x_1} \leq 0, \quad g_9(X) = -1 + \frac{x_1}{12 x_2} \leq 0, \tag{31}$$

$$g_{10}(X) = -1 + \frac{1.5 x_6 + 1.9}{x_4} \leq 0, \quad g_{11}(X) = -1 + \frac{1.1 x_7 + 1.9}{x_5} \leq 0, \tag{32}$$

with

$$\begin{aligned} 2.6 \leq x_1 \leq 3.6, \\ 0.7 \leq x_2 \leq 0.8, \\ 17 \leq x_3 \leq 28, \\ 7.3 \leq x_4 \leq 8.3, \\ 7.8 \leq x_5 \leq 8.3, \\ 2.9 \leq x_6 \leq 3.9, \\ 5.0 \leq x_7 \leq 5.5. \end{aligned}$$

This problem is discussed by numerous optimization works such as anti-coronavirus optimization algorithm (ACVO) [209], MFO [210], modified DE (MDE) [211], chaotic and neighborhood search-based ABC (CNSABC) [212], LFD [206], SCA [210], GWO [207], ABC [213], WOA [207], mine blast algorithm (MBA) [30], ant lion optimizer (ALO) [207], harris hawks-sine cosine method (HHO-SCA) [214], spotted hyena algorithm (SHO) [215], accelerated PSO (APSO) [216], salp swarm algorithm (SSA) [207], multi-verse optimizer (MVO) [210], AO [169], RSA [173], AOA [170], POA [217], stock exchange trading optimization algorithm (SETO) [218], socio evolution & learning optimization algorithm (SELO) [218], combinatorial the evolutionary optimizer and the adaptive constraint-handling method (HEAA) [219], iLSHADEε [107], εMAgES [107], IUDE [107], VMCH [182], DE-QL [181], L-SHADE with a new adaptive Levy flight mutation operator (COLSHADE) [220],

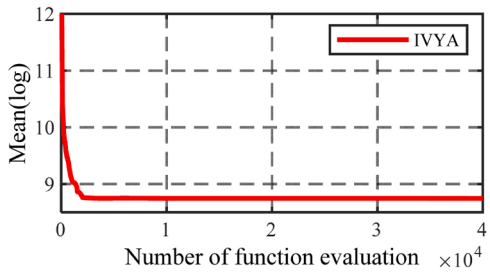


Fig. 19. The convergence plot for the pressure vessel problem obtained via IVYA (NFEs ∈ [1, 40000]).

enhanced multi-operator DE (EnMODE) [221], and an improved εMAgES with restarts (BP-εMAg-ES) [222]. Table 23 displays the experimental outcomes. According to the table, the IVYA outperformed other methods in minimizing the speed reducer's weight based on the Mean, Best, and Worst indices. Although the proposed IVYA, compared to the modern and modified algorithms iLSHADEε, εMAgES, IUDE, VMCH, DE-QL, COLSHADE, EnMODE, and BP-εMAg-ES, has good and satisfactory comparison results because the proposed Ivy method is a basic method and it can be improved in the near future. In addition, Table 24 indicates the ideal variable values derived by the IVYA. Fig. 21 depicts the convergence tendency of the mean of solutions obtained using the IVYA.

4.7. Design of the tension/compression spring issue

According to three limitations [198,223], the objective of this problem is to minimize the weight of the spring illustrated in Fig. 22. The following three control parameters (i.e., problem variables), the wire's diameter d (denoted as x_1), the number of active coils P (denoted as x_3), and the mean coil's diameter D (denoted as x_2) need to be tuned [224].

Minimize:

$$f(X) = x_1^2 x_2 (2 + x_3) \tag{33}$$

Subject to:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0, \quad g_2(X) = -1 + \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} \leq 0, \tag{34}$$

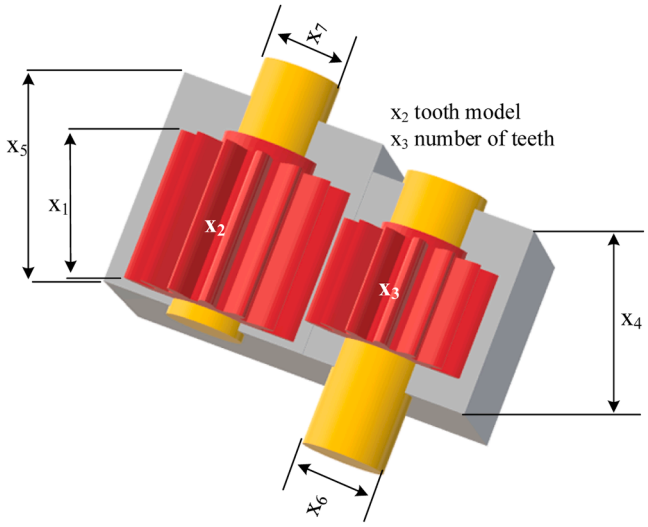


Fig. 20. The schema of the speed reducer.

Table 23
The optimal solutions for the design of the speed reducer.

Optimizers	Best	Mean	Worst	Std.
MFO [210]	3009.5717	3021.2565	3054.5248	11.0235
ACVO [209]	2994.4718	N.A.	N.A.	N.A.
MDE [211]	2996.356689	2996.367220	N.A.	8.2E-03
CNSABC	**2994.47107**	2995.36298	2997.28675	0.9864
LFD [206]	3007.7820	N.A.	N.A.	N.A.
SCA [210]	3030.5633	3065.9172	3104.7791	18.0742
AOA [170]	2997.9157	N.A.	N.A.	N.A.
ABC [213]	2997.058412	2997.058412	N.A.	**0.00E + 00**
SELO [218]	2999.2274	N.A.	N.A.	N.A.
MBA [30]	2994.482453	2996.769019	2999.652444	1.56
ALO [207]	2996.521745	3005.644280	3014.379001	4.7422
hHHO-SCA [214]	3029.873076	N.A.	N.A.	N.A.
SHO [215]	2998.5507	2999.64	3003.889	1.93E + 00
APSO [216]	3187.630486	3822.640624	4443.017639	366.146
RSA [173]	2996.5157	N.A.	N.A.	N.A.
SSA [207]	2996.021720	3005.574377	3015.662612	4.63871
AO [169]	3007.7328	N.A.	N.A.	N.A.
MVO [210]	3002.9281	3028.8411	3060.9582	13.0186
POA [217]	2996.3482	2999.88	3001.491	1.782335
SETO [218]	2994.4991	N.A.	N.A.	N.A.
GWO [207]	2995.704435	3001.556162	3009.944297	4.1218
WOA [207]	2996.604340	3003.042915	3233.598124	4.0888E + 01
HEAA [219]	2994.499107	2994.613368	2994.752311	7.0E-02
iLSHADEε [107]	**2.99E + 03**	**2.99E + 03**	**2.99E + 03**	0.00E + 00
DE-QL [181]	**2.994424E + 03**	**2.994424E + 03**	**2.994424E + 03**	4.641246E−13
VMCH [182]	2.99E + 03	2.99E + 03	3.00E + 03	1.04E + 00
COLSHADE [220]	**2.9944E + 03**	**2.9944E + 03**	**2.9944E + 03**	4.5475E−13
IUDE [107]	**2.99E + 03**	**2.99E + 03**	**2.99E + 03**	0.00E + 00
εMAgES [107]	**2.99E + 03**	**2.99E + 03**	**2.99E + 03**	0.00E + 00
BP-εMAg-ES [222]	**2.9944E + 03**	**2.9944E + 03**	**2.9944E + 03**	4.6413E−13
EnMODE [221]	**2.9944E + 03**	**2.9944E + 03**	**2.9944E + 03**	4.6412E−13
IVYA	**2994.47107**	**2994.47118**	**2994.47612**	6.645E-04

Table 24
The optimal solutions achieved for the speed reducer issue.

Variables	IVYA
x_1	3.5
x_2	0.70
x_3	17.0
x_4	7.30
x_5	7.71531992
x_6	3.35021467
x_7	5.28665447
$g_1(X)$	−0.073915
$g_2(X)$	−0.1979985
$g_3(X)$	−0.499172
$g_4(X)$	−0.90464
$g_5(X)$	−3.49549E-09
$g_6(X)$	−2.8469E-09
$g_7(X)$	−0.7025
$g_8(X)$	0.0
$g_9(X)$	−0.58333
$g_10(X)$	−0.051326
$g_11(X)$	−3.8884E-10
f_{Best}	**2994.47107**

$$g_3(X) = -\frac{140.45x_1}{x_2^2 x_3} + 1 \le 0,$$
$$g_4(X) = -1 + \frac{x_1 + x_2}{1.5} \le 0, \quad (35)$$

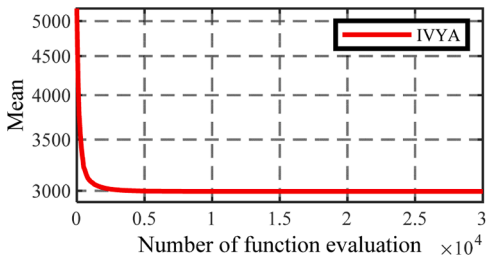


Fig. 21. The convergence plot for the problem obtained via IVYA (NFEs ∈ [1, 30000]).

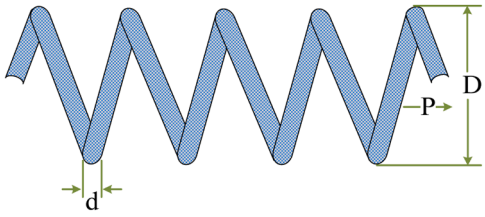


Fig. 22. The schema of the tension/compression spring.

with

$2 \le x_3 \le 15,\ 0.25 \le x_2 \le 1.3,\ 0.05 \le x_1 \le 2$.

IVYA is compared with the following algorithms: dynamic differential annealed optimization (DDAO) [225], mGWO [178], the weighted mean of vectors (INFO) optimizer [226], Gaussian quantum-behaved PSO (G-QPSO) [199], MFO [210], GCHHO [183], which integrates the Gaussian mutation and cuckoo search (CS) strategy to the Harris hawks optimizer (HHO), pathfinder algorithm (PFA) [227], EA without using penalty functions (1 + λ)-ES [228], Coot algorithm [229], MVO [210], hybrid GA (HGA) [230], and UPSO [194] as shown in Table 25. Moreover, Table 26 displays the lowest cost and matches the ideal variable values that the IVYA found. The IVYA found the minimum spring weight to be 0.0126653. The convergence graph for this problem using the Ivy approach is shown in Fig. 23 as a final step.

4.8. I-beam vertical deflection optimal design

This optimal design challenge aims to achieve the lowest possible vertical deflection with outstanding algorithmic performance, as shown in Fig. 24. This problem involves four parameters: the width b, two thicknesses t_w, t_f, and the height h. The following is the problem's theoretical structure [24]:

Minimize:

Table 25
The optimal solution to the tension/compression spring issue.

Optimizers	Best	Mean	Worst	Std.
DDAO [225]	0.0129065	0.0151829	0.0173199	1.26E-03
mGWO [178]	0.012668	N.A.	N.A.	N.A.
UPSO [194]	0.01312	0.02294	N.A.	7.2E-03
G-QPSO [199]	**0.012665**	0.013524	0.017759	1.268E-03
MFO [210]	0.012753902	0.014023657	0.017236590	0.001390
GCHHO [183]	**0.0126653**	N.A.	N.A.	N.A.
PFA [227]	**0.0126653**	N.A.	N.A.	N.A.
HGA [230]	0.012668	0.013481	0.016155	N.A.
Coot [229]	**0.0126653**	N.A.	N.A.	N.A.
MVO [210]	0.012816930	0.014464372	0.017839737	0.001622
(1 + λ)-ES [228]	0.012689	**0.013165**	N.A.	3.9E-04
INFO [226]	0.012666	N.A.	N.A.	N.A.
IVYA	**0.0126653**	0.0134604	**0.0159532**	3.62E-04

Table 26
The best solution to the issue tension/compression spring problem.

Variables	IVYA
x_1	0.051719438
x_2	0.357448964
x_3	11.246226277
$g_1(X)$	−1.2089*E*-07
$g_2(X)$	−1.003*E*-08
$g_3(X)$	−4.0552
$g_4(X)$	−0.72722
f_{Best}	0.0126653

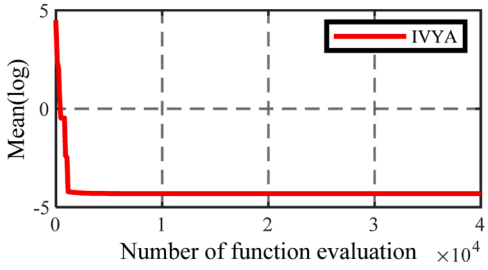


Fig. 23. The convergence plot for the tension/compression spring problem obtained via IVYA (NFEs ∈ [1, 40000]).

$$f(h,b,t_w,t_f) = \frac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^3}{6} + 2bt_f\left(\frac{h-t_f}{2}\right)^2} \tag{36}$$

Subject to:

$$g_1 = t_w(h-2t_f) + 2bt_f \le 300, \tag{37}$$

$$g_2 = \frac{18h\cdot 10^4}{2bt_f\left(4t_f^2 + 3h(h-2t_f)\right) + t_w(h-2t_f)^3} + \frac{15b\cdot 10^3}{2t_fb^3 + (h-2t_f)t_w^3} \tag{38}$$

with

$$0.9 \le t_f \le 5,$$
$$0.9 \le t_w \le 5,$$
$$10 \le h \le 80,$$
$$10 \le b \le 50.$$

IVYA is compared with some top optimization techniques for such situations, including CS [231], symbiotic organisms search (SOS) [185], CPA [24], and AOS [185]. The findings are presented in Table 27. Table 27 reveals the IVYA's minimal vertical deflection was 0.0130741, which is the best optimization outcome for this kind of issue. Fig. 25 shows the IVYA convergence curve for this problem. The top Ivy algorithm-derived strategies for creating I-beams with slight vertical deflection are also shown in Table 28.

4.9. Piston lever optimal design

The primary objective of this problem, depicted in Fig. 26, is to optimize the oil amount for piston components*H*, *B*, *X*, and *D* while the

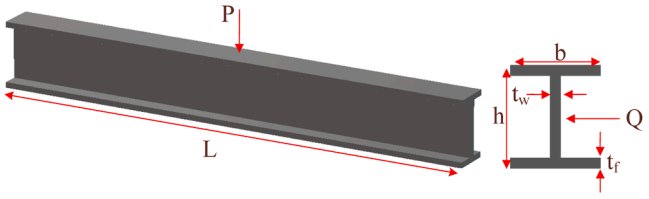


Fig. 24. The schema of the design of the I-beam.

Table 27
The optimal solutions for the I-beam vertical deflection problem.

Optimizers	Best	Mean	Worst	Std.
CS [231]	0.0130747	0.0132165	0.01353646	0.0001345
SOS [185]	**0.0130741**	**0.0130884**	N.A.	4.0E-05
CPA [24]	0.0132240	N.A.	N.A.	N.A.
AOS [185]	**0.0130741**	0.0131788	0.013814	1.555E-04
IVYA	**0.0130741**	0.0131269	**0.013369**	**3.13E-05**

piston lever angle increases from 0° to 45°. The problem's formulation is provided at [232]:

Minimize:

$$f(z) = f(H,B,X,D) = \frac{1}{4}\pi D^2(L_2 - L_1) \tag{39}$$

Subject to

$$g_1(z) = Q\cdot L\cos\theta - R\cdot F \le 0 \text{ at } \theta = 45^\circ, \tag{40}$$

$$\begin{aligned} g_2(z) &= Q\cdot(L-X) - M_{\max} \le 0,\\ g_3(z) &= 1.2(L_2 - L_1) - L_1 \le 0,\\ g_4(z) &= 0.5D - B \le 0, \end{aligned} \tag{41}$$

where

$$R = |H(B - X\cos\theta) - X(X\sin\theta + H)|\cdot\left(\sqrt{H^2 + (X-B)^2}\right)^{-1}, \qquad F = \frac{\pi PD^2}{4}, \tag{42}$$

$$L_1 = \sqrt{H^2 + (X-B)^2}, \qquad L_2 = \sqrt{(B - X\cos\theta)^2 + (X\sin\theta + H)^2}, \tag{43}$$

with

$$0.05 \le H,\ B,\ D \le 500,\ 0.05 \le X \le 120,$$

and with values of constants: $Q = 10,000$ lbs, $P = 1,500$ psi, $L = 240$ in, and $M_{max} = 1.8\cdot 10^6$ lbs·in.

These restrictions on inequality consider the force equilibrium, geometrical requirements, the smallest piston stroke, and the maximum lever bending moment. Table 29 shows the internal search algorithm (ISA) [233], hybrid PSO (HPSO), Q-learning (HPSO-Q) [232], DE [232], PSO [232], and the proposed IVYA. After 5000 evaluations, the IVYA produces results much superior to those of other approaches. The top results the IVYA came up with are depicted in Table 30. The convergence graph for this problem using the Ivy approach is demonstrated in Fig. 27 as a final step.

4.10. Design of the corrugated bulkhead

Tankers frequently use corrugated bulkhead designs, as shown in Fig. 28, to aid in efficiently washing cargo tanks at facilities [234]. This

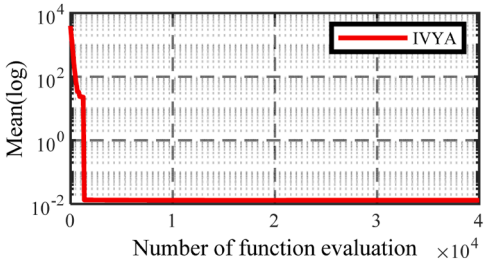


Fig. 25. The convergence plot for the issue of the vertical deflection of the I-beam obtained via IVYA (NFEs ∈ [1, 40000]).

Table 28
The optimal solution to the I-beam vertical deflection problem.

Variables	IVYA
h	80.0
b	50.0
t_w	0.90
t_f	2.321792260
$g_1(X)$	$-6.80E{-}08$
$g_2(X)$	-1.570228
f_{Best}	**0.0130741**

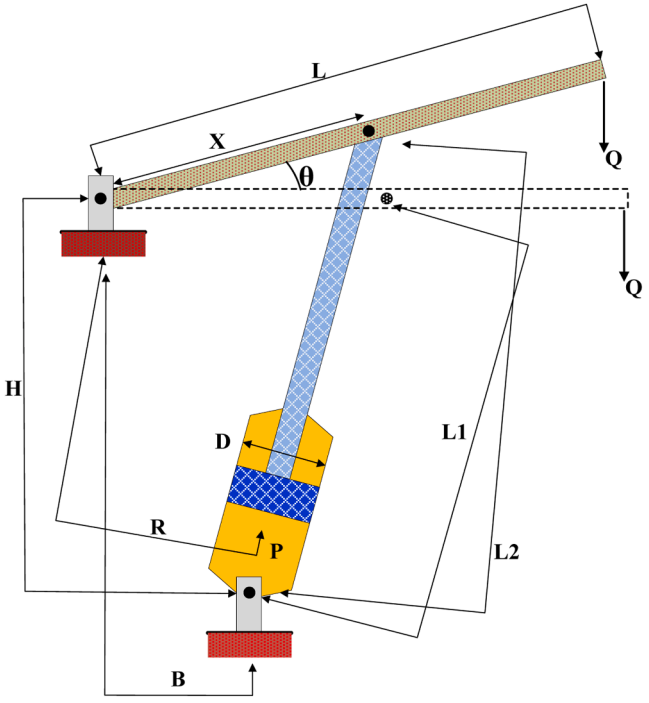


Fig. 26. The schema of the piston lever.

Table 29
The optimal solutions to the piston lever problem.

Optimizers	Best	Mean	Worst	Std.
ISA [233]	**8.4**	226.5	610.6	111.2
HPSO-Q [232]	129	151	**168**	**13.4**
DE [232]	159	187	199	14.2
PSO [232]	122	166	294	51.7
IVYA	**8.41269832**	**145.6892**	286.2716	125.37

issue serves as an illustration of the corrugated bulkheads for a tanker's minimum-weight construction. The objective function's plate, the thickness t, the length l, the depth h, and the width b, are its four control

Table 30
The optimal results for the piston lever problem.

Variables	IVYA
H	0.05
B	2.04151359
X	120
D	4.08302718
$g_1(z)$	$-1.373E$-04
$g_2(z)$	$-600{,}000$
$g_3(z)$	$-1.172E+02$
$g_4(z)$	0.00
f_{Best}	**8.41269832410**

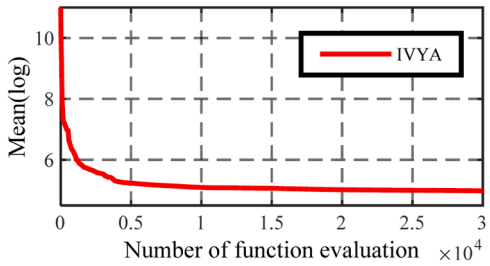


Fig. 27. Convergence plots for the piston lever problem obtained via IVYA (NFEs $\in [1,\ 30000]$).

variables. The mathematical formula for the optimization problem for the corrugated bulkheads for a tanker's minimum-weight design is as follows [177].

Minimize:

$$f(z) = f(b,h,l,t) = \frac{5.885t\cdot(b+l)}{b+\sqrt{l^2-h^2}}. \tag{44}$$

Subject to

$$g_1(z) = t\cdot h\cdot\left(0.4b+\frac{l}{6}\right) - 8.94\left(b+\sqrt{l^2-h^2}\right) \geq 0, \tag{45}$$

$$g_2(z) = -2.2\left(8.94\left(b+\sqrt{l^2-h^2}\right)\right)^{\frac{4}{3}} + \left(0.2b+\frac{l}{12}\right)\left(th^2\right) \geq 0, \tag{46}$$

$$\begin{aligned} g_3(z) &= t - 0.15 - 0.0156b \geq 0, \\ g_4(z) &= t - 0.15 - 0.0156l \geq 0, \end{aligned} \tag{47}$$

$$\begin{aligned} g_5(z) &= -1.05 + t \geq 0, \\ g_6(z) &= -h + l \geq 0, \end{aligned} \tag{48}$$

where

$$0 < b,h,l < 100,\ 0 < t < 5.$$

Table 31 provides the optimal results as determined through FA [177], LF-FA [177], AD-IFA [177], LS-LF-FA [177], FPSA [235], and IVYA, as well as the comparison of minimum-weight results. The optimal solutions computed through 30 independent runs are used to quantify accuracy performance. Value 6.84295801 was obtained as the optimum weight in this investigation using AD-IFA [177] and the suggested Ivy method. The top results the Ivy algorithm came up with have been given in Table 32. Fig. 29 provides IVYA's convergence graph for the ideal corrugated bulkhead design. The IVYA is notably distinguished by its rapid convergence to the global minimum.

4.11. Design of the tubular column

The aim of this optimization issue is to produce a tubular column, as shown in Fig. 30, with a tubular part that must support a compressive force $P = 2500$ kgf at the minimum possible cost [236]. The column's

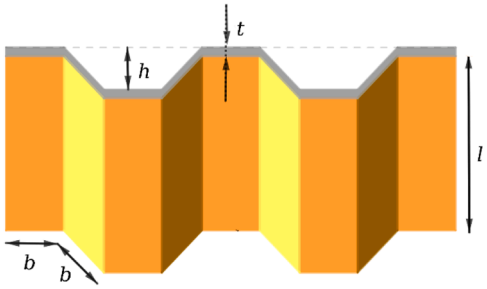


Fig. 28. The schema of the corrugated bulkhead.

Table 31
The optimal solutions for the optimal design of the corrugated bulkhead.

Optimizers	Best	Mean	Worst	Std.
FA [177]	7.21	10.23	N.A.	1.95
LF-FA [177]	6.95	8.83	N.A.	1.26
AD-IFA [177]	**6.84**	7.21	N.A.	0.58
LS-LF-FA [177]	6.86	7.44	N.A.	0.67
FPSA [235]	7.008391	N.A.	N.A.	N.A.
IVYA	**6.84295801**	**6.9245807**	**7.9378206**	**0.186**

component material must adhere to the following requirements: yield stress $\sigma_y = 500$ kgf·cm^{-2}, density $\rho = 0.0025$ kgf·cm^{-3}, the length of the column $L = 250$cm, and modulus of elasticity $E = 8.5 \cdot 10^5$ kgf·cm^{-2}. The mean diameter of the column d and the thickness of the column t are the two control variables. The objective function has the following form when a column's construction and material costs are included in the model and subject to six constraints [236].

Minimize:

$$f(z) = f(d,t) = 9.8d{\cdot}t + 2d \tag{49}$$

Subject to

$$\begin{aligned} g_1(z) &= \frac{P}{\pi d{\cdot}t{\cdot}\sigma_y} - 1 \le 0, \\ g_2(z) &= \frac{8P}{\pi^3 E{\cdot}d{\cdot}t{\cdot}\left(d^2+t^2\right)} - 1 \le 0, \\ g_3(z) &= \frac{2}{d} - 1 \le 0, \end{aligned} \tag{50}$$

$$\begin{aligned} g_4(z) &= \frac{d}{14} - 1 \le 0, \\ g_5(z) &= \frac{1}{5t} - 1 \le 0, \\ g_6(z) &= \frac{5t}{4} - 1 \le 0, \end{aligned} \tag{51}$$

where $2 < d < 14$ and $0.2 < t < 0.8$.

We chose the six most recently developed methods for experimental comparison in the design of the tubular column. The optimal solutions of the experiment have been displayed in Table 33. The ideal design was found using the suggested IVYA with a minimal design cost of $26.499497. The IVYA, however, was closely followed by the other methods used for comparison. The IVYA's top solutions are in Table 34, and convergence graphs for that solution have been given in Fig. 31.

4.12. Design of the car side impact problem

The primary objective is to optimize the car structural elements to bolster passenger safety during side-impact collisions, enhance vehicle efficiency, and minimize overall weight. The aim is to optimize eleven design parameters, including the B-pillar inner's thickness, the cross members, the floor side inner, its reinforcement, the door beam, its reinforcement along the beltline, the barrier height, its hitting position, the floor side inner, the roof rail, and its material to minimize the car's weight. We used the same mathematical model that was used in [207] with ten constraints, as follows.

Table 32
The optimal outcomes for the optimal design of the corrugated bulkhead.

Variables	IVYA
b	57.69230769
h	34.14762035
l	57.69230769
t	1.05
$g_1(z)$	$-2.40694E+02$
$g_2(z)$	$-2.32688E$-06
$g_3(z)$	$-3.60001E$-11
$g_4(z)$	$-3.60001E$-11
$g_5(z)$	0.0
$g_6(z)$	-23.54469
f_{Best}	**6.84295801**

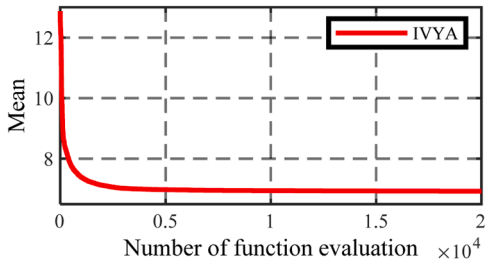


Fig. 29. The convergence plot for the corrugated bulkhead design issue obtained via IVYA (NFEs $\in [1, 20000]$).

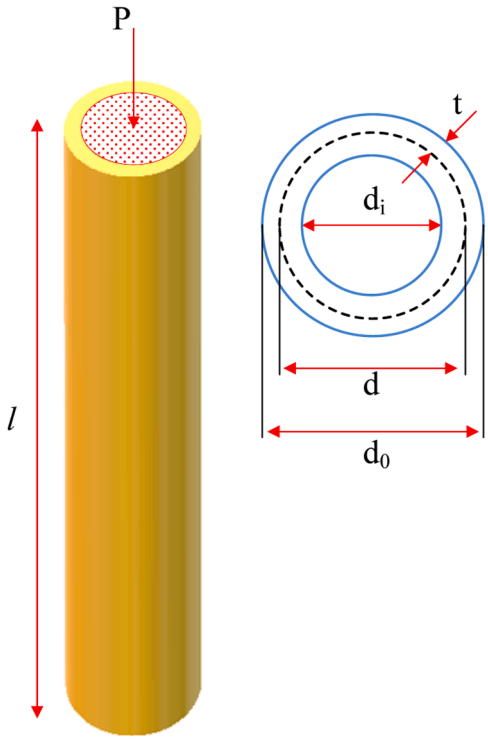


Fig. 30. The schema of the tubular column.

Table 33
The optimal solutions to the tubular column issue.

Optimizers	Best	Mean	Worst	Std.
LS-LF-FA [177]	26.50	26.58	N.A.	0.19
AD-IFA [177]	26.50	26.54	N.A.	0.07
DBB-BC [236]	26.49954	26.50117	N.A.	2.0E-03
LF-FA [177]	26.50	27.46	N.A.	1.36
ISA [233]	26.531	26.531	26.532	0.00017
FA [177]	26.52	28.74	N.A.	2.08
IVYA	**26.499497**	**26.499617**	**26.502314**	**1.45E-04**

Minimize:

$$f(X) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7 \tag{52}$$

Subject to

Table 34
The optimal results for the tubular column issue.

Variables	IVYA
d	5.45115622
t	0.29196548
$g_1(z)$	−7.16807E-09
$g_2(z)$	−2.00896E-09
$g_3(z)$	−0.6331054
$g_4(z)$	−0.610632
$g_5(z)$	−0.3149875
$g_6(z)$	−0.6350432
f_{Best}	**26.499497**

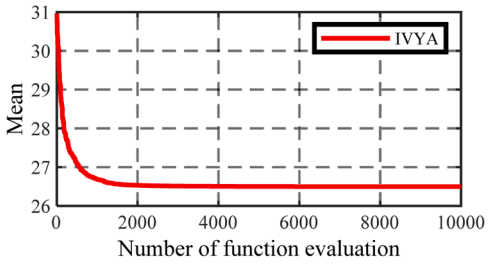


Fig. 31. The convergence plot for the tubular column problem obtained via IVYA (NFEs $\in [1, 10000]$).

$$g_1(X) = 1.16 - 0.3717x_2x_4 + 0.01343x_6x_{10} - 0.484x_3x_9 - 0.00931x_2x_{10} \leq 1, \tag{53}$$

$$\begin{aligned} g_2(X) = {} & 0.261 + 0.00001575x_{10}x_{11} + 0.00139x_8x_{11} + 0.08045x_6x_9 \\ & + 0.0008757x_5x_{10} + 0.0144x_3x_5 - 0.0159x_1x_2 - 0.019x_2x_7 \\ & - 0.188x_1x_8 \\ \leq {} & 0.32, \end{aligned} \tag{54}$$

$$\begin{aligned} g_3(X) = {} & 0.214 + 0.00817x_5 + 0.00121x_8x_{11} + 0.0007715x_5x_{10} + 0.121x_3x_9 \\ & + 0.03099x_2x_6 + 0.0208x_3x_8 - 0.0005354x_6x_{10} - 0.00364x_5x_6 \\ & - 0.018x_2x_7 - 0.0704x_1x_9 - 0.131x_1x_8 \\ \leq {} & 0.32, \end{aligned} \tag{55}$$

$$\begin{aligned} g_4(X) = {} & 0.74 + 0.227x_2^2 + 0.001232x_3x_{10} - 0.166x_7x_9 - 0.163x_3x_8 - 0.61x_2 \\ \leq {} & 0.32, \end{aligned} \tag{56}$$

$$\begin{aligned} g_5(X) = {} & 28.98 + 0.32x_9x_{10} + 6.63x_6x_9 + 3.818x_3 + 0.0207x_5x_{10} - 7.7x_7x_8 \\ & - 4.2x_1x_2 \\ \leq {} & 32, \end{aligned} \tag{57}$$

$$\begin{aligned} g_6(X) = {} & 33.86 + 22x_8x_9 + 0.1792x_{10} + 2.95x_3 - 9.98x_7x_8 - 0.0215x_5x_{10} \\ & - 11x_2x_8 - 5.057x_1x_2 \\ \leq {} & 32, \end{aligned} \tag{58}$$

$$g_7(X) = 46.36 + 0.1107x_3x_{10} - 12.9x_1x_8 - 9.9x_2 \leq 32, \tag{59}$$

$$\begin{aligned} g_8(X) = {} & 4.72 + 0.000191x_{11}^2 + 0.009325x_6x_{10} - 0.0122x_4x_{10} - 0.19x_2x_3 \\ & - 0.5x_4 \\ \leq {} & 4, \end{aligned} \tag{60}$$

$$\begin{aligned} g_9(X) = {} & 10.58 + 0.028x_6x_{10} + 0.02054x_3x_{10} - 0.0198x_4x_{10} - 1.95x_2x_8 \\ & - 0.674x_1x_2 \\ \leq {} & 9.9, \end{aligned} \tag{61}$$

$$\begin{aligned} g_{10}(X) = {} & 16.45 + 0.0432x_9x_{10} - 0.000786x_{11}^2 - 0.843x_5x_6 - 0.489x_3x_7 \\ & - 0.0556x_9x_{11} \\ \leq {} & 15.7, \end{aligned} \tag{62}$$

where $0.5 \leq x_i \leq 1.5$ for $i = 1, \ldots, 7, x_8, x_9 \in (0.192,\ 0.345)$, and $-30 \leq x_{10}, x_{11} \leq 30$.

The design of an automobile side collision is often used to test the performance of various strategies. The statistical findings produced by the different optimizers for the car side impact optimal modeling are summarized in Table 35. The Ivy algorithm has obtained its solutions through 40,000 NFEs in this example. According to the solutions given in Table 35, the Ivy technique outperforms all the developed methods in this study. Furthermore, the Ivy method produces a superior optimal design compared to the developed methods in this study. Table 36 displays the best results of the IVYA. In addition, the convergence graph produced by IVYA for this problem is shown in Fig. 32.

5. Conclusion and future works

This paper introduced the metaheuristic algorithm called the Ivy algorithm (IVYA). The Ivy plant's ability to grow, move, and intertwine intelligently upwards and toward the sun motivated this new optimization algorithm. The IVYA's capability as an optimizer, as well as its performance and mathematical modeling, were shown. The IVYA underwent evaluation and optimization on diverse test functions, including unimodal, multimodal, classical variations, and shifted-rotated real-parameter benchmark functions, compared against several established optimization methods. The results showcased the IVYA's exceptional performance in exploiting and exploring solution spaces, effectively steering clear of local optima and outperforming established metaheuristics. The tension/compression spring, pressure vessel, welded beam, cantilever beam, three-bar truss, gear train, speed reducer, I-beam vertical deflection, piston lever, corrugated bulkhead, tubular column, and car side impact optimal designs are just a few examples of practical engineering problems to which the IVYA was applied. The simulation results showed that, compared to many contemporary and sophisticated well-known meta-heuristics in recent studies, the IVYA is reasonably robust and competitive for optimal design challenges. Therefore, the IVYA can be viewed as a cutting-edge and reliable algorithm for upcoming research and optimization applications.

However, increasing the dimensions of an optimization problem can reduce the algorithm's convergence speed, which is a possible limitation of the IVYA method. However, this limitation can be mitigated considerably by introducing a term that reduces the effect of random coefficients as the number of iterations increases. Such an approach can help to avoid convergence issues associated with higher-dimensional optimization problems. The potential integration of the IVYA into diverse cooperative co-evolution strategies could elevate its efficiency across multiple categories, optimizing its performance further. In the

Table 35
The optimal solutions to the car side impact optimal design issue.

Optimizers	Best	Mean	Worst	Std.
DE [231]	22.84298	23.22828	24.12606	0.34451
WOA [207]	23.042162	24.814486	27.360814	9.6570E-01
ER-WCA [207]	22.843265	**23.069925**	24.455313	3.5021E-01
GA [231]	22.85653	23.51585	26.240578	0.66555
CSS [237]	23.007336	23.523265	24.863563	0.562345
IVYA	**22.842970**	23.205854	**24.093470**	**0.335850**

Table 36
The optimal results for the car side impact optimal design issue.

Variables	IVYA
x_1	0.5
x_2	1.11645931
x_3	0.50000001
x_4	1.3020416
x_5	0.5
x_6	1.49999991
x_7	0.50000048
x_8	0.345
x_9	0.345
x_10	−19.54476638
x_11	0.01319926
$g_1(X)$	−0.6543979
$g_2(X)$	−0.07423476
$g_3(X)$	−0.064913
$g_4(X)$	−0.046881
$g_5(X)$	−3.7128215
$g_6(X)$	−6.1202479
$g_7(X)$	−9.769E-09
$g_8(X)$	−1.86E−10
$g_9(X)$	−0.965077
$g_10(X)$	−0.2960486
f_{Best}	**22.8429703**

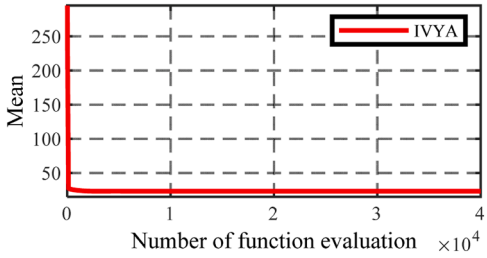


Fig. 32. The Convergence plot for the car side impact optimal design issue obtained via IVYA (NFEs ∈ [1, 40000]).

future, there's potential for creating a binary version of the IVYA, offering an interesting avenue for development. In addition, it is suggested that future researchers consider the incorporation of additional methodologies further to enhance the efficacy and versatility of the IVYA. Specifically, the FDC method presents a promising avenue for effectively addressing constrained optimization problems.

Furthermore, integrating the NSM approach into the update mechanism of IVYA may offer valuable insights for enhancing its performance in multi-objective optimization scenarios. These proposed enhancements hold the potential to contribute to the development of a more robust and adaptable algorithm, facilitating its application across diverse optimization tasks in the future. Also, for future studies, researchers may consider applying modified and enhanced versions of modern algorithms like iLSHADEε, DE-QL, VMCH, COLSHADE, IUDE, εMAgES, BP-εMAg-ES, and EnMODE to the proposed IVYA introduced in this paper. This approach could improve performance and versatility in various application domains. Subsequently, researchers can explore and evaluate the diverse applications of these hybrid methods.

CRediT authorship contribution statement

Mojtaba Ghasemi: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mohsen Zare:** Writing – review & editing, Project administration, Methodology. **Pavel Trojovský:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization. **Ravipudi Venkata Rao:** Supervision, Project administration, Methodology. **Eva Trojovská:** Writing – review & editing, Visualization, Conceptualization. **Venkatachalam Kandasamy:** Visualization, Writing – review & editing.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Declaration of Generative AI and AI-assisted technologies in the writing process

The authors declare that they did not use AI-assisted technologies to create this article.

Acknowledgments

This work was partly based upon work supported by the Faculty of Science, University of Hradec Králové, Czech Republic, under Grant of Excellence no: 2210/2024-2025.

References

[1] M.E. Çimen, Y. Yalçin, A novel hybrid firefly–whale optimization algorithm and its application to optimization of MPC parameters, Soft. Comput. 26 (2022) 1845–1872.
[2] A. Bahreininejad, H. Taib, The novel combination lock algorithm for improving the performance of metaheuristic optimizers, Adv. Eng. Softw. 172 (2022) 103177.
[3] D.H. Winfield, Function and functional optimization by interpolation in data tables, Harvard University, 1970.
[4] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308–313.
[5] M. Ghasemi, I.F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, L. Li, A novel and effective optimization algorithm for global optimization and its engineering applications: turbulent flow of water-based optimization (TFWO), Eng. Appl. Artif. Intell. 92 (2020) 103666.
[6] M.A. Akbari, M. Zare, R. Azizipanah-Abarghooee, S. Mirjalili, M. Deriche, The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems, Sci. Rep. 12 (2022) 1–20.
[7] H.T. Kahraman, M. Katı, S. Aras, D.A. Taşci, Development of the natural survivor method (NSM) for designing an updating mechanism in metaheuristic search algorithms, Eng. Appl. Artif. Intell. 122 (2023) 106121.
[8] M. Ghasemi, M.-A. Akbari, C. Jun, S.M. Bateni, M. Zare, A. Zahedi, et al., Circulatory system based optimization (CSBO): an expert multilevel biologically inspired meta-heuristic algorithm, Eng. Appl. Comput. Fluid Mech. 16 (2022) 1483–1525.
[9] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[10] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft. Comput. 23 (2019) 715–734.
[11] N. Chopra, M.M. Ansari, Golden jackal optimization: a novel nature-inspired optimizer for engineering applications, Expert. Syst. Appl. 198 (2022) 116924.
[12] M. Ghasemi, A. Rahimnejad, R. Hemmati, E. Akbari, S.A. Gadsden, Wild geese algorithm: a novel algorithm for large scale optimization based on the natural life and death of wild geese, Array 11 (2021) 100074.
[13] Y. Jiang, Q. Wu, S. Zhu, L. Zhang, Orca predation algorithm: a novel bio-inspired algorithm for global optimization problems, Expert. Syst. Appl. 188 (2022) 116026.
[14] N. Eslami, S. Yazdani, M. Mirzaei, E. Hadavandi, Aphid-ant mutualism: a novel nature-inspired metaheuristic algorithm for solving optimization problems, Math. Comput. Simul. (2022).
[15] M.H. Sulaiman, Z. Mustaffa, M.M. Saari, H. Daniyal, Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (2020) 103330.
[16] H. Jia, X. Peng, C. Lang, Remora optimization algorithm, Expert. Syst. Appl. 185 (2021) 115665.
[17] J.-S. Pan, L.-G. Zhang, R.-B. Wang, V. Snášel, S-C. Chu, Gannet optimization algorithm: a new metaheuristic algorithm for solving engineering optimization problems, Math. Comput. Simul. 202 (2022) 343–373.

[18] F. MiarNaeimi, G. Azizyan, M. Rashki, Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems, Knowl.-Based Syst. 213 (2021) 106711.
[19] C. Li, G. Chen, G. Liang, F. Luo, J. Zhao, Z.Y. Dong, Integrated optimization algorithm: a metaheuristic approach for complicated optimization, Inf. Sci. (N.Y.) 586 (2022) 424–449.
[20] A.-A.A. Mohamed, S.A. Hassan, A.M. Hemeida, S. Alkhalaf, M.M.M. Mahmoud, Eldin AMB. parasitism–predation algorithm (PPA): a novel approach for feature selection, Ain Shams. Eng. J. 11 (2020) 293–308.
[21] G. Dhiman, V. Kumar, Emperor penguin optimizer: a bio-inspired algorithm for engineering problems, Knowl.-Based Syst. 159 (2018) 20–50.
[22] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Dwarf mongoose optimization algorithm, Comput. Methods Appl. Mech. Eng. 391 (2022) 114570.
[23] R.R. Jaya, A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, Int. J. Ind. Eng. Comput. 7 (2016) 19–34.
[24] J. Tu, H. Chen, M. Wang, A.H. Gandomi, The colony predation algorithm, J. Bionic. Eng. 18 (2021) 674–710.
[25] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[26] S. Shadravan, H.R. Naji, V.K. Bardsiri, The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, Eng. Appl. Artif. Intell. 80 (2019) 20–34.
[27] W. Zhao, L. Wang, Z. Zhang, Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm, Neural. Comput. Appl. 32 (2020) 9383–9425.
[28] D. Zaldivar, B. Morales, A. Rodríguez, A. Valdivia-G, E. Cuevas, M. Pérez-Cisneros, A novel bio-inspired optimization model based on yellow saddle goatfish behavior, Biosystems 174 (2018) 1–21.
[29] J.M. Abdullah, T. Ahmed, Fitness dependent optimizer: inspired by the bee swarming reproductive process, IEEe Access. 7 (2019) 43473–43486.
[30] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems, Appl. Soft. Comput. 13 (2013) 2592–2612.
[31] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Trans. Cybern. 45 (2014) 191–204.
[32] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, MHS'95 Proc. Sixth Int. Symp. Micro. Mach. Hum. Sci. (1995), https://doi.org/10.1109/mhs.1995.494215.
[33] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[34] N.A. Kallioras, N.D. Lagaros, D.N. Avtzis, Pity beetle algorithm–A new metaheuristic inspired by the behavior of bark beetles, Adv. Eng. Softw. 121 (2018) 147–166.
[35] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, Futur. Gener. Comput. Syst. 97 (2019) 849–872.
[36] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: squirrel search algorithm, Swarm. Evol. Comput. 44 (2019) 148–175.
[37] M. Ghasemi, E. Akbari, A. Rahimnejad, S.E. Razavi, S. Ghavidel, L. Li, Phasor particle swarm optimization: a simple and efficient variant of PSO, Soft. Comput. 23 (2019), https://doi.org/10.1007/s00500-018-3536-8.
[38] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: a new method for stochastic optimization, Futur. Gener. Comput. Syst. 111 (2020) 300–323.
[39] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine predators algorithm: a nature-inspired metaheuristic, Expert. Syst. Appl. 152 (2020) 113377.
[40] S. Kaur, L.K. Awasthi, A.L. Sangal, G. Dhiman, Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.
[41] A.R. Moazzeni, E. Khamehchi, Rain optimization algorithm (ROA): a new metaheuristic method for drilling optimization solutions, J. Pet. Sci. Eng. 195 (2020) 107512.
[42] D. Połap, M. Woźniak, Red fox optimization algorithm, Expert. Syst. Appl. 166 (2021) 114107.
[43] H. Zamani, M.H. Nadimi-Shahraki, A.H. Gandomi, Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization, Comput. Methods Appl. Mech. Eng. 392 (2022) 114616.
[44] K. Zervoudakis, S. Tsafarakis, A global optimizer inspired from the survival strategies of flying foxes, Eng. Comput. (2022) 1–34.
[45] C. Zhong, G. Li, Z. Meng, Beluga whale optimization: a novel nature-inspired metaheuristic algorithm, Knowl.-Based Syst. 251 (2022) 109215.
[46] I. Naruei, F. Keynia, Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems, Eng. Comput. (2021) 1–32.
[47] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, Honey badger algorithm: new metaheuristic algorithm for solving optimization problems, Math. Comput. Simul. 192 (2022) 84–110.
[48] J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, Syst. Sci. Control Eng. 8 (2020) 22–34.
[49] M.C. Catalbas, A. Gulten, Pufferfish optimization algorithm: a bioinspired optimizer, Handb. Intell. Comput. Optim. Sustain. Dev. (2022) 461–485.
[50] B. Mohammad Hasani Zade, N. Mansouri, PPO: a new nature-inspired metaheuristic algorithm based on predation for optimization, Soft. Comput. 26 (2022) 1331–1402.
[51] S. Zhao, T. Zhang, S. Ma, M. Chen, Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications, Eng. Appl. Artif. Intell. 114 (2022) 105075.
[52] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, W. Zhao, Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 114 (2022) 105082.
[53] M. Braik, A. Hammouri, J. Atwan, M.A. Al-Betar, M.A. Awadallah, White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems, Knowl.-Based Syst. 243 (2022) 108457.
[54] H.T. Sadeeq, A.M. Abdulazeez, Giant trevally optimizer (GTO): a novel metaheuristic algorithm for global optimization and challenging engineering problems, IEEE Access. 10 (2022) 121615–121640.
[55] M. Abdel-Basset, R. Mohamed, M. Jameel, M. Abouhawwash, Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, Knowl.-Based Syst. 262 (2023) 110248.
[56] G. Hu, Y. Guo, G. Wei, L. Abualigah, G. Khan, Shark optimizer: a novel nature-inspired algorithm for engineering optimization, Adv. Eng. Inform. 58 (2023) 102210.
[57] B. Ahmadi, J.S. Giraldo, G. Hoogsteen, Dynamic hunting leadership optimization: algorithm and applications, J. Comput. Sci. 69 (2023) 102010.
[58] J.-S. Chou, C-Y. Liu, Pilgrimage walk optimization: folk culture-inspired algorithm for identification of bridge deterioration, Autom. Constr. 155 (2023) 105055.
[59] A.H. Rabie, N.A. Mansour, A.I. Saleh, Leopard seal optimization (LSO): a natural inspired meta-heuristic algorithm, Commun. Nonlinear. Sci. Numer. Simul. (2023) 107338.
[60] K. Golalipour, S.A. Nowdeh, E. Akbari, S.S. Hamidi, D. Ghasemi, A.Y. Abdelaziz, et al., Snow avalanches algorithm (SAA): a new optimization algorithm for engineering applications, Alexandria Eng. J. 83 (2023) 257–285.
[61] M.V. Anaraki, S. Farzin, Humboldt squid optimization algorithm (HSOA): a novel nature-inspired technique for solving optimization problems, IEEE Access. 11 (2023) 122069–122115.
[62] S. Zhao, T. Zhang, S. Ma, M. Wang, Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems, Appl. Intell. 53 (2023) 11833–11860.
[63] A. Majumder, Termite alate optimization algorithm: a swarm-based nature inspired algorithm for optimization problems, Evol. Intell. 16 (2023) 997–1017.
[64] S.K. De, The goat search algorithms, Artif. Intell. Rev. 56 (2023) 8265–8301.
[65] M. Abdel-Basset, R. Mohamed, M. Abouhawwash, Crested porcupine optimizer: a new nature-inspired metaheuristic, Knowl.-Based Syst. (2023) 111257.
[66] S. Zhao, T. Zhang, L. Cai, R. Yang, Triangulation topology aggregation optimizer: a novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications, Expert. Syst. Appl. 238 (2024) 121744.
[67] M.H. Amiri, N. Mehrabi Hashjin, M. Montazeri, S. Mirjalili, N. Khodadadi, Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm, Sci. Rep. 14 (2024) 5032.
[68] W. Zhao, L. Wang, Z. Zhang, H. Fan, J. Zhang, S. Mirjalili, et al., Electric eel foraging optimization: a new bio-inspired optimizer for engineering applications, Expert. Syst. Appl. 238 (2024) 122200.
[69] G. Hu, B. Du, K. Chen, G. Wei, Super eagle optimization algorithm based three-dimensional ball security corridor planning method for fixed-wing UAVs, Adv. Eng. Inform. 59 (2024) 102354.
[70] B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovsk\`y, F. S. Gharehchopogh, E.-S.M. El-kenawy, et al., Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning, Cluster. Comput. (2024) 1–49.
[71] J. Lian, G. Hui, L. Ma, T. Zhu, X. Wu, A.A. Heidari, et al., Parrot optimizer: algorithm and applications to medical problems, Comput. Biol. Med. (2024) 108064.
[72] M. Zhang, G. Wen, Duck swarm algorithm: theory, numerical optimization, and applications, Cluster. Comput. (2024) 1–29.
[73] S. Gajawada, POSTDOC: the human optimization, in: Second Int. Conf. Inf. Technol. Converg. Serv. (ITCSE 2013), December, 2013, pp. 20–21.
[74] Q. Askari, I. Younas, M. Saeed, Political optimizer: a novel socio-inspired meta-heuristic for global optimization, Knowl.-Based Syst. 195 (2020) 105709.
[75] M. Mirrashid, H. Naderpour, Transit search: an optimization algorithm based on exoplanet exploration, Results Control Optim. 7 (2022) 100127.
[76] R. Moghdani, K. Salimifard, Volleyball premier league algorithm, Appl. Soft. Comput. 64 (2018) 161–185.
[77] M. Elsisi, Future search algorithm for optimization, Evol. Intell. 12 (2019) 21–31.
[78] O. Montiel, O. Castillo, P. Melin, A.R. D\'\iaz, R. Sepúlveda, Human evolutionary model: a new approach to optimization, Inf. Sci. (N.Y.) 177 (2007) 2075–2098.
[79] Y. Shi, Brain storm optimization algorithm, in: Adv. Swarm Intell. Second Int. Conf. ICSI 2011, Chongqing, China, June 12-15, 2011, Proceedings, Part I 2, 2011, pp. 303–309.
[80] S. Satapathy, A. Naik, Social group optimization (SGO): a new population evolutionary optimization technique, Complex Intell. Syst. 2 (2016) 173–203.
[81] L. Wang, L. An, J. Pi, M. Fei, P.M. Pardalos, A diverse human learning optimization algorithm, J. Glob. Optim. 67 (2017) 283–323.
[82] W. Zhao, L. Wang, Z. Zhang, Supply-demand-based optimization: a novel economics-inspired algorithm for global optimization, IEEe Access. 7 (2019) 73182–73206.
[83] S.H.S. Moosavi, V.K. Bardsiri, Poor and rich optimization algorithm: a new human-based and multi populations algorithm, Eng. Appl. Artif. Intell. 86 (2019) 165–181.

[84] A. Kaveh, M. Khanzadi, M.R. Moghaddam, Billiards-inspired optimization algorithm; a new meta-heuristic method, Structures 27 (2020) 1722–1739.
[85] A.I. Wagan, M.M. Shaikh, A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem, Appl. Soft. Comput. 90 (2020) 106176.
[86] H. Ghasemian, F. Ghasemian, H. Vahdat-Nejad, Human urbanization algorithm: a novel metaheuristic approach, Math. Comput. Simul. 178 (2020) 1–15.
[87] S.Q. Salih, A.A. Alsewari, A new algorithm for normal and large-scale optimization problems: nomadic people optimizer, Neural. Comput. Appl. 32 (2020) 10359–10386.
[88] E.F. Veysari, A new optimization algorithm inspired by the quest for the evolution of human society: human felicity algorithm, Expert Syst. Appl. 193 (2022) 116468.
[89] T.S.L.V. Ayyarao, N.S.S. RamaKrishna, R.M. Elavarasan, N. Polumahanthi, M. Rambabu, G. Saini, et al., War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization, IEEe Access. 10 (2022) 25073–25105.
[90] S. Thapliyal, N. Kumar, Numeric crunch algorithm: a new metaheuristic algorithm for solving global and engineering optimization problems, Soft. Comput. 27 (2023) 16611–16657.
[91] J. Lian, G. Hui, Human evolutionary optimization algorithm, Expert Syst. Appl. 241 (2024) 122638.
[92] A. Taheri, K. RahimiZadeh, A. Beheshti, J. Baumbach, R.V. Rao, S. Mirjalili, et al., Partial reinforcement optimizer: an evolutionary optimization algorithm, Expert. Syst. Appl. (2023) 122070.
[93] H.T. Kahraman, S. Aras, E. Gedikli, Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms, Knowl.-Based Syst. 190 (2020) 105169.
[94] B. Ozkaya, H.T. Kahraman, S. Duman, U. Guvenc, Fitness-distance-constraint (FDC) based guide selection method for constrained optimization problems, Appl. Soft. Comput. 144 (2023) 110479.
[95] H.T. Kahraman, H. Bakir, S. Duman, M. Kat\i, S. Aras, U. Guvenc, Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination, Appl. Intell. (2022) 1–36.
[96] S. Duman, H.T. Kahraman, M. Kati, Economical operation of modern power grids incorporating uncertainties of renewable energy sources and load demand using the adaptive fitness-distance balance-based stochastic fractal search algorithm, Eng. Appl. Artif. Intell. 117 (2023) 105501.
[97] H.T. Kahraman, M.H. Hassan, M. Kat\i, M. Tostado-Véliz, S. Duman, S. Kamel, Dynamic-fitness-distance-balance stochastic fractal search (dFDB-SFS algorithm): an effective metaheuristic for global optimization and accurate photovoltaic modeling, Soft. Comput. (2023) 1–28.
[98] S. Duman, H.T. Kahraman, Y. Sonmez, U. Guvenc, M. Kati, S. Aras, A powerful meta-heuristic search algorithm for solving global optimization and real-world solar photovoltaic parameter estimation problems, Eng. Appl. Artif. Intell. 111 (2022) 104763.
[99] S. Duman, H.T. Kahraman, B. Korkmaz, H. Bakir, U. Guvenc, C. Yilmaz, Improved phasor particle swarm optimization with fitness distance balance for optimal power flow problem of hybrid AC/DC power grids, Int. Conf. Artif. Intell. Appl. Math. Eng. (2021) 307–336.
[100] V.K. Patel, V.J. Savsani, Heat transfer search (HTS): a novel optimization algorithm, Inf. Sci. (N.Y.) 324 (2015) 217–246.
[101] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[102] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: a novel physics-based algorithm, Futur. Gener. Comput. Syst. 101 (2019) 646–667.
[103] M. Ghasemi, S. Ghavidel, J. Aghaei, E. Akbari, L. Li, CFA optimizer: a new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems, Int. Trans. Electr. Energy Syst. 28 (2018) e2536, https://doi.org/10.1002/etep.2536.
[104] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190.
[105] I. Ahmadianfar, O. Bozorg-Haddad, X. Chu, Gradient-based optimizer: a new metaheuristic optimization algorithm, Inf. Sci. (N.Y.) 540 (2020) 131–159.
[106] V. Punnathanam, P. Kotecha, Yin-Yang-pair optimization: a novel lightweight optimization algorithm, Eng. Appl. Artif. Intell. 54 (2016) 62–79.
[107] A.F. Nematollahi, A. Rahiminejad, B. Vahidi, A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization, Appl. Soft. Comput. 59 (2017) 596–621.
[108] E.H. Houssein, M.R. Saad, F.A. Hashim, H. Shaban, M. Hassaballah, Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 94 (2020) 103731.
[109] J.L.J. Pereira, M.B. Francisco, C.A. Diniz, G.A. Oliver, S.S. Cunha Jr, G.F. Gomes, Lichtenberg algorithm: a novel hybrid physics-based meta-heuristic for global optimization, Expert. Syst. Appl. 170 (2021) 114522.
[110] M. Abdel-Basset, R. Mohamed, K.M. Sallam, R.K. Chakrabortty, Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm, Mathematics 10 (2022) 3466.
[111] M.H. Qais, H.M. Hasanien, R.A. Turky, S. Alghuwainem, M. Tostado-Véliz, F. Jurado, Circle search algorithm: a geometry-based metaheuristic optimization algorithm, Mathematics 10 (2022) 1626.
[112] Q. Pan, J. Tang, S. Lao, EDOA: an elastic deformation optimization algorithm, Appl. Intell. (2022) 1–20.
[113] A. Mahdavi-Meymand, M. Zounemat-Kermani, Homonuclear molecules optimization (HMO) meta-heuristic algorithm, Knowl.-Based Syst. 258 (2022) 110032.
[114] V. Goodarzimehr, S. Shojaee, S. Hamzehei-Javaran, S. Talatahari, Special relativity search: a novel metaheuristic method based on special relativity physics, Knowl.-Based Syst. (2022) 109484.
[115] V. Goodarzimehr, S. Talatahari, S. Shojaee, S. Hamzehei-Javaran, Special relativity search for applied mechanics and engineering, Comput. Methods Appl. Mech. Eng. 403 (2023) 115734.
[116] W. Zhao, L. Wang, Z. Zhang, S. Mirjalili, N. Khodadadi, Q. Ge, Quadratic interpolation optimization (QIO): a new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems, Comput. Methods Appl. Mech. Eng. 417 (2023) 116446.
[117] J. Bai, Y. Li, M. Zheng, S. Khatir, B. Benaisa, L. Abualigah, et al., A sinh cosh optimizer, Knowl.-Based Syst. (2023) 111081.
[118] M. Abdel-Basset, R. Mohamed, S.A.A. Azeem, M. Jameel, M. Abouhawwash, Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion, Knowl.-Based Syst. 268 (2023) 110454.
[119] M.H. Qais, H.M. Hasanien, S. Alghuwainem, K.H. Loo, Propagation search algorithm: a physics-based optimizer for engineering applications, Mathematics 11 (2023) 4224.
[120] Y. Gao, PID-based search algorithm: a novel metaheuristic algorithm based on PID algorithm, Expert. Syst. Appl. (2023) 120886.
[121] H. Su, D. Zhao, A.A. Heidari, L. Liu, X. Zhang, M. Mafarja, et al., RIME: a physics-based optimization, Neurocomputing. 532 (2023) 183–214.
[122] S. Thapliyal, N. Kumar, Hyperbolic sine optimizer: a new metaheuristic algorithm for high performance computing to address computationally intensive tasks, Cluster. Comput. (2024) 1–70.
[123] R. Sowmya, M. Premkumar, P. Jangir, Newton-Raphson-based optimizer: a new population-based metaheuristic algorithm for continuous optimization problems, Eng. Appl. Artif. Intell. 128 (2024) 107532.
[124] A. Radhakrishnan, G. Jeyakumar, Evolutionary algorithm for solving combinatorial optimization—a review, Innov. Comput. Sci. Eng. Proc. 8th ICICSE (2021) 539–545.
[125] D. Tang, S. Dong, Y. Jiang, H. Li, Y. Huang, ITGO: invasive tumor growth optimization algorithm, Appl. Soft. Comput. 36 (2015) 670–698.
[126] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1992) 66–73.
[127] A.H. Kashan, R. Tavakkoli-Moghaddam, M. Gen, Find-fix-finish-exploit-analyze (F3EA) meta-heuristic algorithm: an effective algorithm with new evolutionary operators for global optimization, Comput. Ind. Eng. 128 (2019) 192–218.
[128] F. Mart\'\inez-Álvarez, G. Asencio-Cortés, J.F. Torres, D. Gutiérrez-Avilés, L. Melgar-Garc\'\ia, R. Pérez-Chacón, et al., Coronavirus optimization algorithm: a bioinspired metaheuristic based on the COVID-19 propagation model, Big. Data 8 (2020) 308–322.
[129] M. Ćalasan, M. Micev, Ž. Djurovic, H.M.A. Mageed, Artificial ecosystem-based optimization for optimal tuning of robust PID controllers in AVR systems with limited value of excitation voltage, Int. J. Electr. Eng. Educ. (2020) 002072092094060, https://doi.org/10.1177/0020720920940605.
[130] A.M. Khalid, K.M. Hosny, S. Mirjalili, COVIDOA: a novel evolutionary optimization algorithm based on coronavirus disease replication lifecycle, Neural Comput. Appl. 34 (2022) 22465–22492.
[131] S. Kadkhoda Mohammadi, D. Nazarpour, M. Beiraghi, A novel metaheuristic algorithm inspired by COVID-19 for real-parameter optimization, Neural Comput. Appl. 35 (2023) 10147–10196.
[132] H.-L. Minh, T. Sang-To, G. Theraulaz, M.A. Wahab, T. Cuong-Le, Termite life cycle optimizer, Expert. Syst. Appl. 213 (2023) 119211.
[133] H. Jia, C. Lu, Guided learning strategy: a novel update mechanism for metaheuristic algorithms design and improvement, Knowl.-Based Syst. 286 (2024) 111402.
[134] Y. Gao, J. Zhang, Y. Wang, J. Wang, L. Qin, Love Evolution Algorithm: a stimulus–value–role theory-inspired evolutionary algorithm for global optimization, J. Supercomput. (2024) 1–62.
[135] M. Han, Z. Du, K.F. Yuen, H. Zhu, Y. Li, Q. Yuan, Walrus optimizer: a novel nature-inspired metaheuristic algorithm, Expert. Syst. Appl. 239 (2024) 122413.
[136] C. Y\ilmaz, E. Cengiz, H.T. Kahraman, A new evolutionary optimization algorithm with hybrid guidance mechanism for truck-multi drone delivery system, Expert. Syst. Appl. 245 (2024) 123115.
[137] H. Su, Z. Dong, Y. Liu, Y. Mu, S. Li, L. Xia, Symmetric projection optimizer: concise and efficient solving engineering problems using the fundamental wave of the Fourier series, Sci. Rep. 14 (2024) 6032.
[138] S. Akyol, B. Alatas, Plant intelligence based metaheuristic optimization algorithms, Artif. Intell. Rev. 47 (2017) 417–462.
[139] M. Ulucan, G. Yildirim, B. Alatas, K.E. Alyamac, A new intelligent sunflower optimization based explainable artificial intelligence approach for early-age concrete compressive strength classification and mixture design of RAC, Struct. Concr. 24 (2023) 7400–7418.
[140] B. McClintock, The significance of responses of the genome to challenge, Science (80-) 226 (1984) 792–801.
[141] A. Trewavas, Plant intelligence: mindless mastery, Nature 415 (2002) 841.
[142] L.S. dos Santos, V.H.S. dos Santos, F.R. Scarano, Plant intelligence: history and current trends, Theor. Exp. Plant Physiol. (2024) 1–11.
[143] T. Trewavas, Plant intelligence: an overview, Bioscience 66 (2016) 542–551.
[144] A. Cheraghalipour, M. Hajiaghaei-Keshteli, M.M. Paydar, Tree growth algorithm (TGA): a novel approach for solving optimization problems, Eng. Appl. Artif. Intell. 72 (2018) 393–414.

[145] L. Cheng, X. Wu, Y. Wang, Artificial flora (AF) optimization algorithm, Appl. Sci. 8 (2018) 329.
[146] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1 (2006) 355–366.
[147] W. Cai, W. Yang, X. Chen, A global optimization algorithm based on plant growth theory: plant growth optimization, in: 2008 Int. Conf. Intell. Comput. Technol. Autom. 1, 2008, pp. 1194–1199.
[148] Z. Zhao, Z. Cui, J. Zeng, X. Yue, Artificial plant optimization algorithm for constrained optimization problems, in: 2011 Second Int. Conf. Innov. bio-inspired Comput. Appl., 2011, pp. 120–123.
[149] X-S. Yang, Flower pollination algorithm for global optimization, Int. Conf. Unconv. Comput. Nat. Comput. (2012) 240–249.
[150] X. Qi, Y. Zhu, H. Chen, D. Zhang, B. Niu, An idea based on plant root growth for numerical optimization, in: Intell. Comput. Theor. Technol. 9th Int. Conf. ICIC 2013, Nanning, China, July 28-31, 2013. Proc. 9, 2013, pp. 571–578.
[151] H. Zhang, Y. Zhu, H. Chen, Root growth model: a novel approach to numerical function optimization and simulation of plant root system, Soft. Comput. 18 (2014) 521–537.
[152] M.S. Kiran, TSA: tree-seed algorithm for continuous optimization, Expert. Syst. Appl. 42 (2015) 6686–6698.
[153] M. Sulaiman, A. Salhi, A seed-based plant propagation algorithm: the feeding station model, Sci. World J. 2015 (2015).
[154] G.F. Gomes, S.S. da Cunha, A.C. Ancelotti, A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates, Eng. Comput. 35 (2019) 619–626.
[155] K.M. Ong, P. Ong, C.K. Sia, A carnivorous plant algorithm for solving global optimization problems, Appl. Soft. Comput. 98 (2021) 106833.
[156] D. Chen, Y. Ge, Y. Wan, Y. Deng, Y. Chen, F. Zou, Poplar optimization algorithm: a new meta-heuristic optimization technique for numerical optimization and image segmentation, Expert. Syst. Appl. 200 (2022) 117118.
[157] Q. Zhang, H. Gao, Z.-H. Zhan, J. Li, H. Zhang, Growth optimizer: a powerful metaheuristic algorithm for solving continuous and discrete global optimization problems, Knowl.-Based Syst. 261 (2023) 110206.
[158] M. Kaveh, M.S. Mesgari, B. Saeidian, Orchard Algorithm (OA): a new meta-heuristic algorithm for solving discrete and continuous optimization problems, Math. Comput. Simul. 208 (2023) 95–135.
[159] R. Storn, K. Price, Differential evolution–a simple and efficient adaptive scheme for global optimization over continuous spaces: technical report TR-95-012, Int. Comput. Sci. Berkeley, Calif. (1995).
[160] M. Yazdani, F. Jolai, Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm, J. Comput. Des. Eng. 3 (2016) 24–36.
[161] D. Castagneri, M. Garbarino, P. Nola, Host preference and growth patterns of Ivy (Hedera helix L.) in a temperate alluvial forest, Plant Ecol. 214 (2013) 1–9.
[162] H. BAUER, U. BAUER, Photosynthesis in leaves of the juvenile and adult phase of Ivy (Hedera helix), Physiol. Plant 49 (1980) 366–372.
[163] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[164] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm. Evol. Comput. 1 (2011) 3–18.
[165] S. Gürgen, H.T. Kahraman, S. Aras, A. İsmail, A comprehensive performance analysis of meta-heuristic optimization techniques for effective organic rankine cycle design, Appl. Therm. Eng. 213 (2022) 118687.
[166] H.T. Öztürk, H.T. Kahraman, Meta-heuristic search algorithms in truss optimization: research on stability and complexity analyses, Appl. Soft. Comput. 145 (2023) 110573.
[167] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, On the exploration and exploitation in popular swarm-based metaheuristic algorithms, Neural Comput. Appl. 31 (2019) 7665–7683.
[168] M. Ghasemi, M. Zare, A. Zahedi, M.-A. Akbari, S. Mirjalili, L. Abualigah, Geyser inspired algorithm: a new geological-inspired meta-heuristic for real-parameter and constrained engineering optimization, J. Bionic. Eng. (2023) 1–35.
[169] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A.A. Al-Qaness, A. H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250.
[170] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Eng. 376 (2021) 113609.
[171] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: 1998 IEEE Int. Conf. Evol. Comput. proceedings. IEEE world Congr. Comput. Intell. (Cat. No. 98TH8360), 1998, pp. 69–73.
[172] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (2008) 702–713.
[173] L. Abualigah, M. Abd Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer, Expert. Syst. Appl. 191 (2022) 116158.
[174] R. Rao, Rao algorithms: three metaphor-less simple algorithms for solving optimization problems, Int. J. Ind. Eng. Comput. 11 (2020) 107–130.
[175] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[176] A. Kumar, G. Wu, M.Z. Ali, R. Mallipeddi, P.N. Suganthan, S. Das, A test-suite of non-convex constrained optimization problems from the real-world and some baseline results, Swarm. Evol. Comput. 56 (2020) 100693.
[177] J. Wu, Y.-G. Wang, K. Burrage, Y.-C. Tian, B. Lawson, Z. Ding, An improved firefly algorithm for global continuous optimization problems, Expert. Syst. Appl. 149 (2020) 113340.
[178] S. Gupta, K. Deep, A memory-based grey wolf optimizer for global optimization tasks, Appl. Soft. Comput. 93 (2020) 106367.
[179] T. Ray, K-M. Liew, Society and civilization: an optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (2003) 386–396.
[180] P.N. Suganthan, Guidelines for real-world single-objective constrained optimisation competition, in: PN-Suganthan/2020-RW-Constrained-Optimisation, 2020.
[181] D. Kizilay, M.F. Tasgetiren, H. Oztop, L. Kandiller, P.N. Suganthan, A differential evolution algorithm with q-learning for solving engineering design problems, in: 2020 IEEE Congr. Evol. Comput., 2020, pp. 1–8.
[182] X. Wen, G. Wu, M. Fan, R. Wang, P.N. Suganthan, Voting-mechanism based ensemble constraint handling technique for real-world single-objective constrained optimization, in: 2020 IEEE Congr. Evol. Comput., 2020, pp. 1–8.
[183] S. Song, P. Wang, A.A. Heidari, M. Wang, X. Zhao, H. Chen, et al., Dimension decided Harris hawks optimization with Gaussian mutation: balance analysis and diversity patterns, Knowl.-Based Syst. 215 (2021) 106425.
[184] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[185] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[186] Z. Liu, T. Nishi, Multipopulation ensemble particle swarm optimizer for engineering design problems, Math. Probl. Eng. (2020) 2020.
[187] S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert. Syst. Appl. 119 (2019) 210–230.
[188] X. Zhang, S. Wang, K. Zhao, Y. Wang, A salp swarm algorithm based on Harris Eagle foraging strategy, Math. Comput. Simul. 203 (2023) 858–877.
[189] M. Zare, M. Ghasemi, A. Zahedi, K. Golalipour, S.K. Mohammadi, S. Mirjalili, et al., A global best-guided firefly algorithm for engineering problems, J. Bionic. Eng. (2023) 1–30.
[190] S.-F. Hwang, R-S. He, A hybrid real-parameter genetic algorithm for function optimization, Adv. Eng. Inform. 20 (2006) 7–21.
[191] J. Zhang, C. Liang, Y. Huang, J. Wu, S. Yang, An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization, Appl. Math. Comput. 211 (2009) 392–416.
[192] S.J. Mousavirad, H. Ebrahimpour-Komleh, Human mental search: a new population-based metaheuristic optimization algorithm, Appl. Intell. 47 (2017) 850–887.
[193] A.-R. Hedar, M. Fukushima, Derivative-free filter simulated annealing method for constrained continuous global optimization, J. Glob. Optim. 35 (2006) 521–549.
[194] K.E. Parsopoulos, M.N. Vrahatis, Unified particle swarm optimization for solving constrained engineering optimization problems, in: Int. Conf. Nat. Comput., Springer, 2005, pp. 582–591.
[195] E. Mezura-Montes, B. Hernández-Ocana, Bacterial foraging for engineering design problems: preliminary results. Lab Nac Informática Av, LANIA AC)-Universidad Juárez Autónoma Tabasco México, 2008.
[196] S. He, E. Prempain, Q.H. Wu, An improved particle swarm optimizer for mechanical design optimization problems, Eng. Optim. 36 (2004) 585–605.
[197] Y. Li, D. Zhao, A.A. Heidari, S. Wang, H. Chen, Y. Zhang, Gaussian backbone-based spherical evolutionary algorithm with cross-search for engineering problems, J. Bionic. Eng. (2024) 1–37.
[198] M. Shahrouzi, A. Kaveh, An efficient derivative-free optimization algorithm inspired by avian life-saving manoeuvres, J. Comput. Sci. 57 (2022) 101483.
[199] L. dos Santos Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, Expert. Syst. Appl. 37 (2010) 1676–1683.
[200] S. Chun, Y.-T. Kim, T-H. Kim, A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems, Adv. Mech. Eng. 5 (2013) 130750.
[201] G. Brammya, S. Praveena, N.S. Ninu Preetha, R. Ramya, B.R. Rajakumar, D. Binu, Deer hunting optimization algorithm: a new nature-inspired meta-heuristic paradigm, Comput. J. (2019).
[202] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (2007) 89–99.
[203] C.A.C. Coello, N.C. Cortés, Hybridizing a genetic algorithm with an artificial immune system for global optimization, Eng. Optim. 36 (2004) 607–634.
[204] H.S. Bernardino, H.J.C. Barbosa, A.C.C. Lemonge, L.G. Fonseca, A new hybrid AIS-GA for constrained optimization problems in mechanical engineering, in: 2008 IEEE Congr. Evol. Comput. (IEEE world Congr. Comput. Intell., IEEE, 2008, pp. 1455–1462.
[205] S.-X. He, Y-T. Cui, Multiscale medalist learning algorithm and its application in engineering, Acta Mech. (2023) 1–27.
[206] H. Jia, K. Sun, W. Zhang, X. Leng, An enhanced chimp optimization algorithm for continuous optimization domains, Complex Intell. Syst. 8 (2022) 65–82.
[207] A.R. Yildiz, H. Abderazek, S. Mirjalili, A comparative study of recent non-traditional methods for mechanical design optimization, Arch. Comput. Methods Eng. 27 (2020) 1031–1048.
[208] W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications, Comput. Methods Appl. Mech. Eng. 388 (2022) 114194.
[209] H. Emami, Anti-coronavirus optimization algorithm, Soft. Comput. 26 (2022) 4991–5023.
[210] M.S. Braik, Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems, Expert. Syst. Appl. 174 (2021) 114685.

[211] E. Mezura-Montes, J. Velázquez-Reyes, C.A.C. Coello, Modified differential evolution for constrained optimization, in: 2006 IEEE Int. Conf. Evol. Comput., 2006, pp. 25–32.
[212] W. Xiao, G. Li, C. Liu, L. Tan, A novel chaotic and neighborhood search-based artificial bee colony algorithm for solving optimization problems, Sci. Rep. 13 (2023) 20496.
[213] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, J. Intell. Manuf. 23 (2012) 1001–1014.
[214] V.K. Kamboj, A. Nandi, A. Bhadoria, S. Sehgal, An intensify Harris Hawks optimizer for numerical and engineering optimization problems, Appl. Soft. Comput. 89 (2020) 106018.
[215] G. Dhiman, V. Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[216] X.-S. Yang, S. Deb, S. Fong, Accelerated particle swarm optimization and support vector machine for business optimization and applications, Int. Conf. Networked Digit. Technol. (2011) 53–66.
[217] P. Trojovsk\`y, M. Dehghani, Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications, Sensors 22 (2022) 855.
[218] H. Emami, Stock exchange trading optimization algorithm: a human-inspired method for global optimization, J. Supercomput. 78 (2022) 2125–2174.
[219] Y. Wang, Z. Cai, Y. Zhou, Z. Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique, Struct. Multidiscip. Optim. 37 (2009) 395–413.
[220] J. Gurrola-Ramos, A. Hernàndez-Aguirre, O. Dalmau-Cedeño, COLSHADE for real-world single-objective constrained optimization problems, in: 2020 IEEE Congr. Evol. Comput., 2020, pp. 1–8.
[221] K.M. Sallam, S.M. Elsayed, R.K. Chakrabortty, M.J. Ryan, Improved multi-operator differential evolution algorithm for solving unconstrained problems, in: 2020 IEEE Congr. Evol. Comput., 2020, pp. 1–8.
[222] M. Hellwig, H-G. Beyer, A modified matrix adaptation evolution strategy with restarts for constrained real-world problems, in: 2020 IEEE Congr. Evol. Comput., 2020, pp. 1–8.
[223] E.H. Houssein, M.A. Mahdy, M.J. Blondin, D. Shebl, W.M. Mohamed, Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems, Expert. Syst. Appl. 174 (2021) 114689.
[224] F. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (2007) 340–356.
[225] H.N. Ghafil, K. Jármai, Dynamic differential annealed optimization: new metaheuristic optimization algorithm for engineering applications, Appl. Soft. Comput. 93 (2020) 106392.
[226] I. Ahmadianfar, A.A. Heidari, S. Noshadian, H. Chen, A.H. Gandomi, INFO: an efficient optimization algorithm based on weighted mean of vectors, Expert. Syst. Appl. (2022) 116516.
[227] H. Yapici, N. Cetinkaya, A new meta-heuristic optimizer: pathfinder algorithm, Appl. Soft. Comput. 78 (2019) 545–568.
[228] E. Mezura-Montes, C.A.C. Coello, Useful infeasible solutions in engineering optimization with evolutionary algorithms, in: Mex. Int. Conf. Artif. Intell., Springer, 2005, pp. 652–662.
[229] I. Naruei, F. Keynia, A new optimization method based on COOT bird natural life model, Expert. Syst. Appl. 183 (2021) 115352.
[230] H.S. Bernardino, H.J.C. Barbosa, A.C.C. Lemonge, A hybrid genetic algorithm for constrained optimization problems in mechanical engineering, in: 2007 IEEE Congr. Evol. Comput., IEEE, 2007, pp. 646–653.
[231] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (2013) 17–35.
[232] P. Kim, J. Lee, An integrated method of particle swarm optimization and differential evolution, J. Mech. Sci. Technol. 23 (2009) 426–434.
[233] A.H. Gandomi, D.A. Roke, Engineering optimization using interior search algorithm, in: 2014 IEEE Symp. Swarm Intell., 2014, pp. 1–7.
[234] A. Yadav, N. Kumar, Artificial electric field algorithm for engineering optimization problems, Expert. Syst. Appl. 149 (2020) 113308.
[235] M. Abdel-Baset, I. Hezam, A hybrid flower pollination algorithm for engineering optimization problems, Int. J. Comput. Appl. 140 (2016).
[236] D. Prayogo, M.-Y. Cheng, Y.-W. Wu, A.A. Herdany, H. Prayogo, Differential big bang-big crunch algorithm for construction-engineering design optimization, Autom. Constr. 85 (2018) 290–304.
[237] S. Talatahari, M. Azizi, Tribe-charged system search for global optimization, Appl. Math. Model. 93 (2021) 115–133.