



Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications

Weiguo Zhao^a, Liying Wang^{a,*}, Zhenxing Zhang^{b,*}, Honggang Fan^c, Jiajie Zhang^a, Seyedali Mirjalili^{d,e}, Nima Khodadadi^f, Qingjiao Cao^a

^a School of Water Conservancy and Hydropower, Hebei University of Engineering, Handan, Hebei, 056038, China

^b Prairie Research Institute, University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA

^c State Key Laboratory of Hydroscience and Engineering, Department of Energy and Power Engineering, Tsinghua University, Beijing 100084, China

^d Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Fortitude Valley, Brisbane, 4006, QLD, Australia

^e YFL (Yonsei Frontier Lab), Yonsei University, Seoul, Korea

^f Department of Civil and Architectural Engineering, University of Miami, Coral Gables, FL, USA

ARTICLE INFO

Keywords:

Engineering design
Hydropower station sluice
Optimization
Metaheuristics
Swarm intelligence

ABSTRACT

An original swarm-based, bio-inspired metaheuristic algorithm, named electric eel foraging optimization (EEFO) is developed and tested in this work. EEFO draws inspiration from the intelligent group foraging behaviors exhibited by electric eels in nature. The algorithm mathematically models four key foraging behaviors: interaction, resting, hunting, and migration, to provide both exploration and exploitation during the optimization process. In addition, an energy factor is developed to manage the transition from global search to local search and the balance between exploration and exploitation in the search space. EEFO reveals various foraging patterns based on the foraging characteristics of electric eels. In this study, such dynamic patterns and behaviors are mathematically imitated to design an effective global optimizer. The effectiveness of EEFO is verified through a comparison with 12 other algorithms using the 23 test functions, Congress on Evolutionary Computation 2011 (CEC2011) test suite, and Congress on Evolutionary Computation 2017 (CEC2017) test suite. The experimental results reveal that the EEFO algorithm outperforms the other algorithms for 87% of the 23 test functions and 59% of the CEC2011 test suite, as well as for 66%, 52% and 45% of the CEC2017 test suite with 10, 30, and 50 dimensions, respectively. The applicability of EEFO is comprehensively tested with 10 engineering problems and the application of hydropower station sluice opening control under accident tripping conditions. The results demonstrate the superiority and promising prospects of EEFO when solving a wide range of challenging real-world problems. Overall, the proposed algorithm showcases exceptional performance in terms of exploitation, exploration, the ability to balance exploitation and exploration, and avoiding local optima. EEFO exhibits remarkable competitiveness, particularly in optimization problems that involve unimodal characteristics and numerous constraints and variables. The source code of EEFO is publicly available at <https://ww2.mathworks.cn/matlabcentral/fileexchange/153461-electric-eel-foraging-optimization-eefo>.

1. Introduction

In our daily work and life, optimization-oriented problems are ubiquitous. Finding effective and efficient methods to solve optimization problems is gradually becoming an important research topic. Optimization refers to, for a given problem, finding the optimal solution or an acceptable approximate solution among many solutions under certain conditions. With the rapid development of new technologies, many optimization problems are becoming increasingly prevalent and

complex in a wide variety of engineering fields, ranging from image processing (Melman & Evsutin, 2023; Hu et al., 2022a), artificial intelligence (Diop et al., 2020; Tikhamarine et al., 2020), hydrologic and hydraulic modeling (Hosseiny, 2022; Brunetti et al., 2022), production scheduling (Foroughi et al., 2019), automatic control (Dash et al., 2015), aerospace (Wang et al., 2002), biomedicine (Mahalakshmi et al., 2020), production design (Ali et al., 2023), vehicle routing (Kuo et al., 2023), pattern recognition (Too & Abdullah, 2020), mechanical engineering (Abualigah et al., 2021), and fault diagnosis (Yang et al., 2020; Wang

* Corresponding authors.

E-mail addresses: zhaowei@hebeu.edu.cn (W. Zhao), wangliying@hebeu.edu.cn (L. Wang), zhang538@illinois.edu (Z. Zhang).

et al., 2022b) and many more. Optimization can significantly improve problem-solving efficiency, reduce related computational load, and save financial resources. Optimization methods can be generally classified as: mathematical methods and metaheuristic methods.

Mathematical methods refer to finding the optimal solution by each successive iteration according to an established mathematical model with the initial condition. Mathematical methods include the Nelder and Mead algorithm (Wang & Shoup, 2011), gradient descent method (Burges et al., 2005), Hooke and Jeeve algorithm (Altinoz & Yilmaz, 2019), Lagrange multiplier method (Everett, 1963), Newton's method (Galántai, 2000), etc. When problems are simple and the dimensionalities of the solution space are small, traditional mathematical methods can effectively find the optimal solution. However, many large-scale, nonlinear, and multimodal complex optimization problems exist in practical applications (Houssein et al., 2023; Meng et al., 2021; Youssi et al., 2020; Singh et al., 2021; Yadav & Kumar, 2020). Mathematical methods are generally dependent on gradient information of given problems and are sensitive to the initial points (Agushaka & Ezugwu, 2021). For such complex issues, however, it is challenging to find the optimal solution, and it is easy to fall into local optima. Therefore, there are great limitations in tackling complex optimization problems using mathematical methods.

Metaheuristic methods refer to a problem-independent algorithmic framework with heuristics inspired by natural phenomena, biological behaviors, or even mathematics (Abdel-Basset et al., 2018). Metaheuristic methods, as an ideal alternative to mathematical methods, have the merits of randomness, ease of implementation, and black box consideration, and these advantages make up for the shortcomings of the mathematical algorithms. Metaheuristics have recently received significant attention in literature and are frequently employed to tackle various complex engineering problems.

With the rapid development of technology and the emergence of new application needs, optimization algorithms are increasingly being applied to new fields such as smart cities (He et al., 2022), smart water management (Yan et al., 2023), unmanned driving (Hu et al., 2023a) and the Internet of Things (Fiza et al., 2023). In addition, new key technologies such as parallel computing and adaptive computing, have gradually emerged. The emergence of these new technologies greatly improves the efficiency and accuracy of various algorithms and expands the application scope of optimization algorithms. The application of optimization algorithms in other fields is constantly increasing, while also facing many challenges and difficulties. For example, many modern engineering problems often exhibit issues such as nonseparable, non-convex, and large search spaces. For these types of problems, most optimizers may encounter performance degradation or low convergence (Hu et al., 2023b). Therefore, it is critical to continue to explore and research algorithms that improve these applications and find effective optimization technology through practice.

While there are numerous existing optimizers, developing a new optimizer remains essential. According to the no free lunch (NFL) theory (Wolpert & Macready, 1997), no specific optimizer can effectively solve every optimization problem. This is due to three main reasons. First, the stochastic nature of metaheuristic algorithms can result in discrepancies between the found optimal solution and the true solution for a given problem. Especially for problems with unknown solutions, it is difficult to determine whether the existing algorithms are optimal. Therefore, it is feasible to develop new and more efficient optimization algorithms to better address these problems and further enhance the solution accuracy and algorithm efficiency. Second, many practical problems exhibit peculiar characteristics that can be likened to the search behaviors of a particular optimization algorithm, and this search behavior originates from its inspiration leading to the emergence of the algorithm (Salawudeen et al., 2021). Therefore, this algorithm may handle this type of problem very well but it is less powerful when used for different problems. For example, in ant colony optimization (ACO), the path selection of the path planning problem is related to the process of ants searching

for food, so ACO simulates the foraging behavior of ants. When searching for food, ants release pheromones on the ground and choose the next direction according to the concentration of pheromones. When other ants follow and repeat this process, the pheromone accumulation becomes thicker, attracting more ants. In this way, ants can find the optimal path to reach the food source. This real-world foraging strategy of ants becomes the inspiration of ACO. Therefore, ACO is suitable for solving problems related to path planning, such as combinatorial optimization, shop scheduling, and allocation problems. However, ACO is not very good at solving other problems, such as continuous optimization problems and high-dimensional optimization problems. Different metaheuristic algorithms are derived from varying inspirations, which show distinct search behaviors; this makes each algorithm suitable for only certain types of problems (Zhong et al., 2022), and this may result in existing optimization algorithms being unable to effectively solve some newly emerging or highly complex problems. Additionally, a new optimization algorithm can provide unique values beyond what other algorithms offer and complement existing ones. Meanwhile, a new optimization algorithm presents an opportunity for knowledge sharing to tackle challenging real-world problems. Typically, a new optimization algorithm consists of specific strategies or operators that can be easily integrated into existing methods to enhance their optimization performance (Seyyedabbasi, 2022). Therefore, new optimization algorithms are still warranted to try varying search strategies for certain problems and are beneficial for the optimization community. These factors serve as the primary motivation behind the current study.

In the literature, there are three primary approaches to the development of metaheuristic algorithms: proposing new algorithms, combining existing algorithms, and developing hyper heuristics. Developing new optimization algorithms and hybridizing existing ones are not contradictory, but complementary. On the one hand, new optimization algorithms can not only compensate for the shortcomings of existing algorithms in certain areas or specific problems, but also provide better choices for solving complex practical problems. Many new optimization algorithms use strategies or operators with novel search characteristics. These components provide a broad channel for improving existing algorithms to enhance their optimization performance. For example, cuckoo search (CS) (Yang & Deb, 2014) is a relatively new algorithm that proposed a Levy flight strategy with good exploration characteristics. This strategy has been widely incorporated into many existing algorithms to improve their ability to avoid local optima. In addition, many new algorithms are combined with existing algorithms to develop hybrid algorithms, which can take advantage of each algorithm to improve optimization performance. Hyper heuristics is a powerful optimization framework, and a hyper heuristic is identified as a heuristic to choose or generate heuristics (Burke et al., 2010). Hyper heuristics provide a high-level strategy (HLS) by manipulating or managing a set of low-level heuristics (LLH) to generate new heuristics. Metaheuristics fit many problems and are therefore general (Blocho, 2020). Hyper heuristics can improve solution performance and adaptability by combining and adjusting multiple metaheuristics. Thus, metaheuristics are an essential component of hyper heuristics, and developing new metaheuristics provides a broader range of optimization techniques, allowing hyper heuristics to select from a more diverse set of options. This can potentially improve the performance of hyper heuristics by the incorporation of more effective and efficient metaheuristics. In addition, new metaheuristics often introduce innovative strategies and concepts. Hyper heuristics can benefit from these advancements by incorporating them into their decision-making process, thereby enhancing their adaptability to different problem domains. Additionally, the introduction of new metaheuristics can inspire the development of novel hyper heuristics. Researchers may draw inspiration from the new techniques and adapt them to develop more advanced hyper heuristics. It is evident that the development of new metaheuristics plays a catalytic role in the advancement of hyper heuristics. Hence, developing new metaheuristics inspired by different animals or

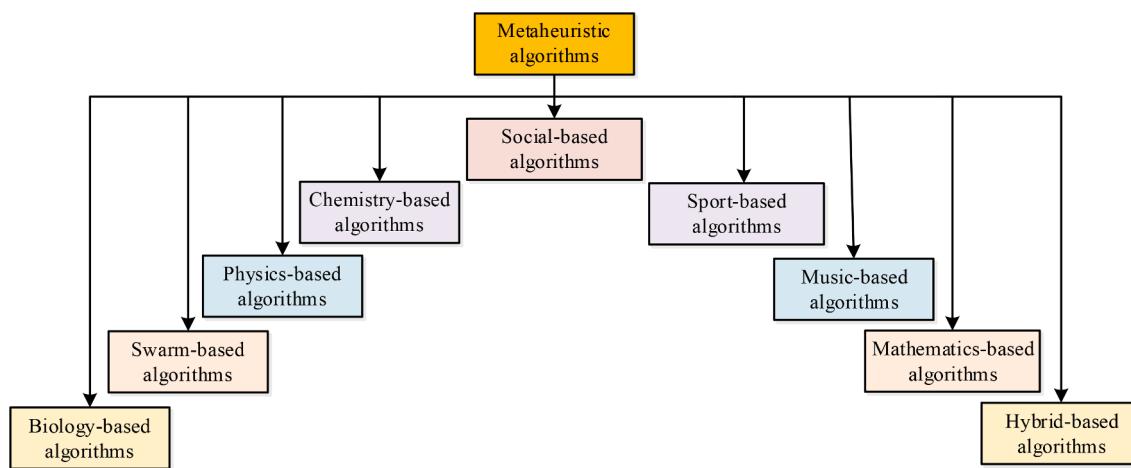


Fig. 1. A classification of metaheuristic algorithms.

natural phenomena is a worthwhile pursuit that allows researchers to explore new perspectives and potentially discover novel and effective optimization techniques. By drawing inspiration from nature, problem-solving strategies that may not have been previously explored can be uncovered. However, it is important to note that reinventing the wheel should be done with a clear purpose and careful consideration. Researchers should ensure that their new algorithms offer significant improvements over existing ones and address limitations in solving practical problems. Therefore, it is necessary to develop new effective algorithms, but this is also a challenge.

There are two studies in the literature that have stimulated the migration behaviors of eels to reach their survival environment (Yao-sheng et al., 2014) and the prey position and communication behaviors of electric fish (Yilmaz & Sen, 2020), but our optimizer in this study focus primarily on modeling the social intelligent foraging behaviors of eels with high efficiency. Therefore, a new optimizer mimicking the socially intelligent foraging behaviors of electric eels, named electric eel foraging optimization (EEFO), is proposed and developed in this study. The EEFO is tested using an overall set of 23 benchmark functions, the CEC2017 test suite, the CEC2011 test suite, and ten engineering problems and it is applied to sluice opening control of a hydropower station under accident tripping conditions.

The primary originality and contribution of this study are summarized below, which also provide an explanation for the optimization performance of EEFO.

- A new optimizer named electric eel foraging optimization (EEFO) is designed and four unique foraging behaviors of electric eels are analyzed thoroughly and expressed mathematically in detail.
- There are four foraging behaviors in EEFO, including interacting, resting, hunting, and migrating. The interacting behavior is mimicked for better exploration and the resting, hunting, and migrating behaviors are simulated for greater exploitation. The energy factor used in EEFO improves the balance between exploration and exploitation.
- The effectiveness of EEFO is verified over 23 benchmark problems, the CEC2017 test suite, the CEC2011 test suite, and 10 engineering problems.
- A wide range of evaluation methods, including convergence analysis, the Wilcoxon test, the Friedman test, and the Nemenyi post-hoc test, are used to assess the effectiveness of EEFO and compare EEFO with 12 well-established metaheuristic algorithms.
- The practicability of EEFO is examined on 10 classical engineering cases and the application of sluice opening control of hydropower stations under accident tripping conditions.

The remainder of this study is organized as follows. The literature review is presented in Section 2. Section 3 describes the inspiration and the mathematical model of EEFO. In Section 4, three sets of experiments are carried out over the 23 benchmark problems, CEC2011 test suite and CEC2017 test suite. In addition, the efficiency of EEFO in tackling 10 engineering applications is tested. The practicality of EEFO in engineering practices using the application of sluice opening control of hydropower stations under accident tripping conditions is presented in Section 5. Conclusions and future directions are included in Section 6.

2. Literature review

Metaheuristics can be organized into nine main types according to their inspirations (Akyol & Alatas, 2017; Alatas & Bingol, 2020): biology-based algorithms (BAs), swarm-based algorithms (SAs), physics-based algorithms (PAs), chemistry-based algorithms (YAs), social-based algorithms (LAs), sport-based algorithms (TAs), music-based algorithms (CAs), mathematics-based algorithms (MAs), and hybrid-based algorithms (HAs). Fig. 1 depicts the classification of metaheuristic algorithms.

BAs are the earliest developed metaheuristic algorithms, which are inspired by natural selection, natural inheritance, and other evolutionary mechanisms related to biology (Back, 1996). One of the most popular BAs is genetic algorithm (GA) (Holland, 1992) which stimulates Darwin's natural selection in biology. Meanwhile, GA is also one of the oldest metaheuristics we have known. In GA, a group of individuals are randomly initiated in the search space and are evolved by several evolutionary operators such as selection, mutation, and recombination in the specific iteration process. At the end of the iterations, the best individual thus far is considered the optimal solution. Differential evolution (DE) (Price, 1996) is a classical BA that shares some evolutionary operators with GA. The significant difference between DE and GA is that DE focuses on the mutation operator while GA emphasizes the crossover operator (Pant et al., 2009). In addition, some other BAs are included in Table 1.

SAs are one of the fastest developing metaheuristic algorithms and originate from the group behaviors of biological populations in nature, ranging from animals, plants, and microorganisms. Some classic SAs have profound effects on the optimization field. ACO (Dorigo & Stützle, 2019) is derived from the colony-foraging behavior of ants. Ants are able to leave behind substances called pheromones in their paths as they forage, sense the strength of these substances and direct ants toward the food. Particle swarm optimization (PSO) (Kennedy & Eberhart, 1995) models the social behaviors of birds or fish. Artificial bee colony (ABC) (Karaboga & Akay, 2009) mimics individual bees' division of labor cooperation when they search for nectar in the natural environment.

Table 1

A brief review of biology-based and swarm-based algorithms.

Categories	Algorithms	Inspiration	Year of proposal
Biology-based algorithms (BAs)	Evolution strategy (ES) (Beyer & Schwefel, 2002) Genetic programming (GP) (Angeline, 1994) Grammatical Evolution (GE) (Ryan et al., 1998) Selfish gene algorithm (SGA) (Coro et al., 1998) Artificial immune system (AIS) (Castro & Timmis, 2003) Biogeography-based optimization (BBO) (Simon, 2008) Wildebeests herd optimization (WHO) (Motevali et al., 2019)	Biological evolution Biological evolution Biological evolutionary process Selfish gene theory Theoretical immunology in human Geographical distribution of biological organisms Wildebeest herding behavior	1973 1992 1998 1998 2003 2008 2019
Swarm-based algorithms (SAs)	Shuffled frog leaping algorithm (SFLA) (Eusuff & Lansey, 2003) Firefly algorithm (FA) (Geem et al., 2001) Wind driven optimization (WDO) (Bayraktar et al., 2013) Grey wolf optimizer (GWO) (Mirjalili, et al., 2014) Moth-flame optimization (MFO) (Mirjalili, 2015) Salp swarm algorithm (SSA) (Mirjalili et al., 2017) Monarch butterfly optimization (MBO) (Wang et al., 2019) Butterfly optimization algorithm (BOA) (Arora & Singh, 2019) Harris hawks optimizer (HHO) (Heidari et al., 2019) Tunicate search algorithm (TSA) (Kaur et al., 2020) Manta ray foraging optimization (MRFO) (Zhao et al., 2020) Capuchin search algorithm (CapSA) (Braik et al., 2021) Horse herd optimization Algorithm (HOA) (MiarNaeimi et al., 2021) Artificial hummingbird algorithm (AHA) (Zhao et al., 2022b) Dwarf mongoose optimization (DMO) (Agushaka et al., 2022a) Sea-horse optimizer (SHO) (Zhao et al., 2022a) Artificial rabbits optimization (ARO) (Wang et al., 2022c) Starling murmuration optimizer (SMO) (Zamani et al., 2022) White shark optimizer (WSO) (Braik et al., 2022) Gazelle optimization algorithm (GOA) (Agushaka et al., 2022b) Honey badger algorithm (HBA) (Hashim et al., 2022) Coati optimization algorithm (COA) (Dehghani et al., 2023)	Foraging behavior of the frog population Flashing behavior of fireflies Atmospheric motion in nature Foraging behaviors of grey wolves Navigation method of moths in nature Navigating and foraging of salp swarm Migration behavior of monarch butterflies Foraging behavior and information sharing of butterflies Collaborative action and hunting style of the Harris hawk Jet propulsion and swarm behaviors of tunicates Intelligent foraging behaviors of manta rays Dynamic behavior of capuchin monkeys Horses' herding behavior Flight skills and intelligent foraging strategies of hummingbirds in nature Foraging and social behavior of dwarf mongoose Movement, hunting, and breeding behavior of sea horses Survival strategies of rabbits in nature Behavior of starlings during stunning murmuration Scholastic behavior of white sharks Gazelles' survival ability in their predator-dominated environment Two foraging behaviors of honey badgers Natural behaviors of coatis	2003 2010 2013 2014 2015 2017 2019 2019 2019 2020 2020 2020 2021 2021 2022 2022 2022 2022 2022 2022 2022 2023

Whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016) mimics the predatory behaviors of whales, including encircling prey, bubble-net attacking and searching for prey. CS models the obligate brood parasitism flight characteristic of cuckoos. This category has enormous various widely known algorithms that are listed in Table 1.

PAs are the main branch of metaheuristic algorithms and are motivated by physical models. These physical models generally consist of various physical laws, processes, phenomena, concepts, and motions, which involve mechanics, heat, electromagnetism, optics, and atomic physics. Gravitational search algorithm (GSA) (Rashedi et al., 2009) is a popular PA that is inspired by the law of gravitation. In GSA, a population of agents moves toward each other according to the gravitational law, and a heavier agent has a better ability to attract other agents. Atom search optimization (ASO) (Zhao et al., 2019) is another typical PA; it simulates the natural motion of atoms using the forces among them, in which the atom interaction occurs due to the forces generated from the Lennard-Jones potential and bond-length potential, which lead to constraint forces. Other common PAs are presented in Table 2.

YAs are generally based on the principles of chemical reaction, which mainly include chemical kinetics and thermodynamics as well as principles of chemical reaction. Chemical reaction optimization (CRO) is also a representative YA method (Lam & Li, 2012) and is based on the phenomenon that molecular collision leads the reaction to the low and stable direction of the potential energy surface in the process of the chemical reaction. According to the law of energy conservation, four basic collision reactions are built. Other YAs are listed in Table 2.

As newly developed metaheuristic algorithms, LAs are receiving increasing attention in the literature. LAs are derived from two main factors: social behaviors and nonphysical activities in humans. Imperalist competitive algorithm (ICA) (Atashpaz-Gargari & Lucas, 2007) originates from the social behaviors of humans that mimic colonial

assimilation mechanisms and imperial competition mechanisms. Teaching-learning-based optimization (TLBO) (Rao et al., 2011) originates from teacher guidance and mutual learning among students. The optimization process is divided into: the teaching stage and the learning stage. The teaching phase means learning from the teacher, and the learning phase means learning through interaction between students. Other examples of LAs are provided in Table 2.

TAs are derived from sports events and fitness programs related to human. League championship algorithm (LCA) (Kashan, 2009) is a classic TA that is inspired by the competition of sport teams in a sport league. In LCA, an artificial league involves weeks of competition among individual sport teams. Each week, teams play in pairs with outcomes based on their win/loss and fitness value determined by a specific formation. During the recovery period, teams adjust their formation/playing style as a new solution for the next week's contest. The championship continues for multiple seasons following the league schedule until a stopping condition is met. Other TAs are shown in Table 3. (See Table 4).

CAs are a new type of metaheuristic algorithm inspired by music and melody in the nature. Harmony search (HS) is a well-known representative of CAs) (Geem et al., 2001), and it resembles music improvisation, in which musicians search for an ideal state of harmony by improvising the pitch of their instruments. Some representatives of CAs are shown in Table 3.

As a new, important part of metaheuristic algorithms, MAs have advanced rapidly in the optimization field. MAs originate from certain mathematical functions, rules, formulas, and theories. Two representatives in this category are sine–cosine algorithm (SCA) (Mirjalili, 2016) and arithmetic optimization algorithm (AOA) (Abualigah et al., 2021). SCA stimulates the fluctuation and periodicity of sine and cosine functions in mathematics. AOA is based on the distribution behaviors of four

Table 2

A brief review of physics-based, chemistry-based and social-based algorithms.

Categories	Algorithms	Inspiration	Year of proposal
Physics-based algorithms (PAs)	Big bang-big crunch (BBBC) (Erol & Eksin, 2006) Central force optimization (CFO) (Formato, 2009) Charged system search (CSS) (Kaveh & Talatahari, 2010) Water cycle algorithm (WCA) (Eskandar et al., 2012) Black hole algorithm (BHA) (Hatamlou, 2013) Kinetic gas molecules optimization (KGMO) (Moein & Logeswaran, 2014) Colliding bodies optimization (CBO) (Kaveh & Mahdavi, 2014) Vortex search algorithm (VSA) (Dogan & Ölmez, 2015) Optics-inspired optimization (OIO) (Kashan, 2015) Water evaporation optimization (WEO) (Kaveh & Bakhshpoori, 2016) Thermal exchange optimization (TEO) (Kaveh & Dadras, 2017) Flow regime algorithm (FRA) (Tahani & Babayan, 2019) Equilibrium optimizer (EO) (Faramarzi et al., 2020b) Lichtenberg algorithm (LA) (Pereira et al., 2021) Energy valley optimizer (EVO) (Azizi et al., 2021) Special relativity search (SRS) (Goodarzimehr et al., 2023) Young double-slit experiment optimizer (YDSE) (Abdel-Basset et al., 2023)	Theories of evolution of universe Particle kinematics in a gravitational field Coulomb law from electrostatics and the Newtonian laws of mechanics Water cycle process in river and sea Black hole in general relativity and cosmology. Law of thermodynamics and heat transfer. Momentum and energy conservation laws Vortex pattern generate by the vortical flow Optical characteristics of convex and concave mirrors Phenomenon of water evaporation on solid surface Process of heat exchange in a thermodynamic system Fluid mechanics and flow regimes Mass balance equation in physics Lichtenberg figures patterns Advanced physics principles and particle decay Interaction of particles in an electromagnetic field Young's double-slit experiment	2006 2007 2010 2012 2013 2014 2014 2015 2015 2016 2017 2019 2020 2021 2023 2023 2023
Chemistry-based algorithms (YAs)	Artificial chemical reaction optimization algorithm (ACROA) (Alatas, 2011)	Types and occurring of chemical reactions.	2011
Social-based algorithms (LAS)	parliamentary optimization algorithm (POA) (Borji & Hamidi, 2009) Brain storm optimization (BSO) (Shi, 2011) Society and civilization algorithm (SCA) (Ray & Liew, 2003) Human mental search (HMS) (Mousavirad & Ebrahimpour-Komleh, 2017) human behavior-based optimization (HBBO) (Ahmadi, 2017) Poor and rich optimization (PRO) (Moosavi & Bardsiri, 2019) Dynamic optimization algorithm (DOA) (Wagan & Shaikh, 2020) Heap-based optimizer (HBO) (Askari et al., 2020a) Human urbanization algorithm (HUA) (Ghasemian et al., 2020) Student psychology-based optimization (SPBO) (Das et al., 2020) Political optimizer (PO) (Askari et al., 2020b) War strategy optimization (WSO) (Ayyarao et al., 2022) Sewing training-based optimization (STBO) (Dehghani et al., 2022) Skill optimization algorithm (SOA) (Givi & Hubalovska, 2023)	Competitive and cooperative behaviors of parliamentary parties Brainstorming process in humans Intra and intersociety interactions within a formal society Exploration strategies of the bid space Behavior of the human beings Behaviors of the poor and the rich to change their economic situation Social behavior in human dynasties Corporate rank hierarchy Human behavior for urbanization and improving life situations Psychology of students in the class Multi-phased process of politics Strategic movement of army troops in the war Teaching the process of sewing to beginner tailors Human efforts to obtain and enhance skills	2009 2011 2003 2017 2019 2020 2020 2020 2020 2022 2022 2023

Table 3

A brief review of sport-based, music-based and mathematics-based algorithms.

Categories	Algorithms	Inspiration	Year of proposal
Sport-based algorithms (TAs)	Soccer league competition (SLC) (Moosavian & Roodsari, 2014) Tug of war optimization (TWO) (Kaveh & Zolghadr, 2016) World cup optimization (WCO) (Razmjoo et al., 2016) Athletic run-based optimization (ARBO) (Barik & Das, 2021)	Soccer leagues the competitions soccer leagues Game of tug of war The world's FIFA competitions Determining the top-performing athlete in a competition	2014 2016 2016 2021
Music-based algorithms (CAs)	Melody search algorithm (MSA) (Ashrafi & Dariane, 2011) Method of musical composition (MMC) (Mora-Gutiérrez et al., 2014)	Performance processes of the group improvisation A dynamic system to compose music	2011 2014
Mathematics-based algorithms (MAs)	Base optimization algorithm (BOA) (Salem, 2012) Golden sine algorithm (GSA) (Tanyildizi & Demir 2017) Lévy flight distribution (LFD) (Houssein et al., 2020) Circle search algorithm (CSA) (Qais et al., 2022)	Basic arithmetic operators Characteristics of sine function Lévy flight random walk Geometrical features of circles	2012 2017 2020 2022

arithmetic operators in mathematics including multiplication, division, subtraction, and addition. Although MAs are not yet competitive compared with other categories in metaheuristic algorithms, this class appears promising. Previous works in this category are presented in Table 3.

HAs are a type of optimization algorithm that combine two or more different types of optimization methods to achieve improved performance and results. By utilizing the advantages of each method while

avoiding their limitations, it can effectively solve intricate problems that are challenging to address with a single optimization method. WOAGWO (Mohammed & Rashid, 2020) is a typical HA in which the hunting strategy of GWO is integrated into the WOA exploitation process, which can compensate for the shortcomings of both. Other HAs are hybrid ACO-ABC algorithm (HACO-ABC) (Comert & Yazgan, 2023), hybrid algorithm based on advanced CS and adaptive Gaussian quantum behaved PSO (AGQPSO) (Kumar et al., 2021), hybrid ant colony and

Table 4
Parameter settings of EEFO and other competing algorithms.

Algorithms	Parameter settings
LFD	$\text{Threshold} = 2$, $\text{CSV} = 0.5$, $\beta = 1.5$, $\alpha_1 = 10$, $\alpha_2 = 0.00005$, $\alpha_3 = 0.005$, $\delta_1 = 0.9$, $\delta_2 = 0.1$
AOA	Control parameter $\sigma = 0.499$ and sensitive parameter $v = 0.5$
WOA	Convergence parameter (a) linearly decreases from 2 to 0
SCA	Parameter (a) = 2
HHO	E_0 = a random value in (-1, 1)
BOA	Sensory modality (c) = 0.01, power exponent (a) increases from 0.1 to 0.3, switch probability (p) = 0.8.
WDO	$RT = 3$, gravitational constant = 0.2, Coriolis effect = 0.4, maximum allowable speed = 0.3
CMA-ES	Expected initial distance to optimum per coordinate $\text{Coordist} = 5$
MFO	a linearly reduces from -1 to -2, $b = 1$
GSA	Gravitational constant = 100 and decreasing coefficient = 20
ASO	Depth weight = 50, multiplier weight = 0.2

ABC (AC-ABC) ([Shunmugapriya & Kanmani, 2017](#)), and many others beyond these examples.

Most algorithms mentioned above share the same characteristics: exploration and exploitation ([Lynn & Suganthan, 2015](#); [Alba & Dorronsoro, 2005](#)). Exploration is how an algorithm searches the entire decision space extensively and globally. Exploitation occurs when an algorithm searches intensively in a local decision space, often around existing solutions, to find better solutions. Exploration makes it easy to search the variable space and produce solutions far away from the current solutions, improving the diversity of the set of candidate solutions. Exploration is valuable to dramatically reduce the chance of being trapped by local optima. However, the convergence rate of exploration is slow, and the precision of global solutions is significantly reduced. Exploitation makes algorithms frequently probe the neighborhood of the current solutions in local areas to find better solutions, increasing convergence and greatly enhancing solution accuracy. However, this search cannot avoid being trapped in local optima. Over-exploration and under-exploitation in an algorithm may slow down convergence and reduce the accuracy of solutions. Under-exploration and over-exploitation may accelerate convergence with the risk of being trapped in local optima. Therefore, to mitigate local optima stagnation and immature convergence, striking a balance between exploitative and exploratory searches is crucial for a successful optimizer ([Wang & Li, 2019](#)).

3. Electric eel foraging optimization (EEFO)

This section describes the inspiration and idea behind EEFO, the mathematical model, procedure, and complexity of EEFO.

3.1. Inspiration

Electric eels are remarkable predators in the animal world. Electric eels are from South America in the family Gymnotidae ([Wikipedia, 2022](#)) and are known for their striking discharge ability in freshwater fishes. Adult eels can release a voltage of 300–800 V to stun their prey and eat it. Therefore, electric eels are known as “high voltage wires” in water. To generate electricity, eels have three pairs of electric organs containing thousands of electrical generating cells called electrocytes. These electrocytes in the body can store power similar to tiny batteries ([National Geographic, 2022](#)). Fig. 2 shows the physical structure and electricity producing organs of electric eels. Eels often generate approximately 10 V to navigate and locate prey due to their poor eyesight. Eels use the feedback from these electrical signals to efficiently track and precisely locate fast-moving prey ([Catania, 2015a](#)). In addition, the high electric charge is used for defense against an enemy as a weapon and low electric discharge is used for communication with each other. Eels can also detect and interpret discharge information from other eels ([Nationalzoo, 2022](#)). This electric charge is a highly advanced method of communication and defense that eels master. When eels find prey, they emit more substantial electric charge rapidly and stun the prey. This ability is also a highly effective foraging strategy in the biological world. New studies suggest that electric eels are swarm-based creatures. Like mammals, eels employ social predation for hunting. This is because groups of eels can coordinate actions, including resting, interacting, migrating, and hunting, to seek prey and conduct social predation ([Bastos et al., 2021](#)). When conducting group hunting, eels tend to cluster together, swim in circles, and herd the groups of fish into a “prey ball” before jointly launching a predatory high-voltage attack on the prey ball ([Bastos et al., 2021](#)). This group hunting behavior increases the opportunity to obtain more prey, especially when there is an abundance of fish. Inspired by these optimized behaviors, EEFO is designed.

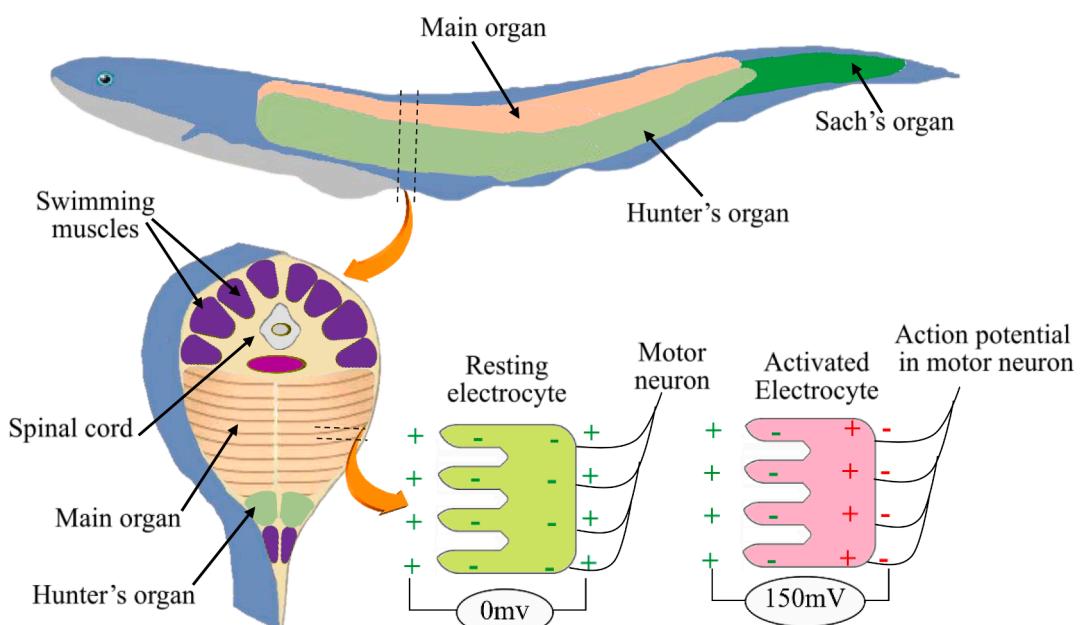


Fig. 2. Physical structure and electricity producing organs of electric eels ([Salama, 2019](#); [Nikki, 2016](#)).

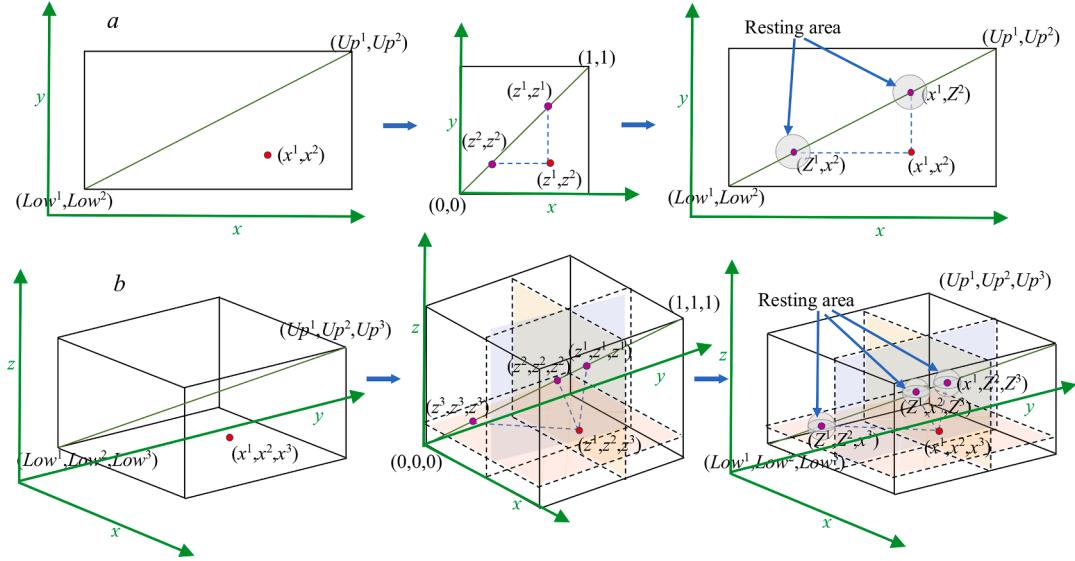


Fig. 3. Procedures to determine a resting area in (a) 2-D space and (b) 3-D space.

3.2. Mathematical model and algorithm

The exploitation and exploration phases of EEFO are modeled from the inspiration of electric eels' social predation, including interacting, resting, migrating, and hunting behaviors (Bastos et al., 2021). The mathematical models of the foraging behaviors are provided below.

3.2.1. Interacting

When eels encounter a school of fish, they interact by swimming and churning with one another. Then, the eels begin to swim in a giant electrified circle to trap numerous small fish at the circle center (Smithsonian, 2021; CARSON, 2021). In EEFO, each electric eel is a candidate solution and the best candidate solution obtained so far in each step is considered the intended prey. The interaction indicates that each eel cooperatively interacts with other individuals using the information of the eels' positions. This behavior can be regarded as the global exploration phase. Specifically, an electric eel can interact with any eel randomly chosen from the population by using the position information of all individuals in the population. Updating an eel's position involves comparing the disparity between a randomly selected eel and the population center.

Additionally, an electric eel can interact with other randomly selected eels within the population by utilizing regional information in the search space. The eel's position is updated by determining the difference between a randomly selected eel from the population and an eel generated randomly within the search space. The interaction between eels is marked by a churn, which denotes a random movement in various directions. The following model represents this churn.

$$C = n_1 \times B \quad (1)$$

$$n_1 \sim N(0, 1) \quad (2)$$

$$B = [b_1, b_2, \dots, b_k, \dots, b_d] \quad (3)$$

$$b(k) = \begin{cases} 1 & \text{if } k == g\{l \\ 0 & \text{else} \end{cases} \quad (4)$$

$$g = \text{randperm}(d) \quad (5)$$

$$l = 1, \dots, \lceil (\frac{T-t}{T} \times r_1 \times (d-2) + 2) \rceil \quad (6)$$

Where T is the number of the maximum iterations. The interacting

behavior can be defined as:

$$\begin{cases} \begin{cases} v_i(t+1) = x_j(t) + C \times (\bar{x}(t) - x_i(t)) & p_1 > 0.5 \\ v_i(t+1) = x_j(t) + C \times (x_r(t) - x_i(t)) & p_1 \leq 0.5 \end{cases} & \text{fit}(x_j(t)) < \text{fit}(x_i(t)) \\ \begin{cases} v_i(t+1) = x_i(t) + C \times (\bar{x}(t) - x_j(t)) & p_2 > 0.5 \\ v_i(t+1) = x_i(t) + C \times (x_r(t) - x_j(t)) & p_2 \leq 0.5 \end{cases} & \text{fit}(x_j(t)) \geq \text{fit}(x_i(t)) \end{cases} \quad (7)$$

$$\bar{x}(t) = \frac{1}{n} \sum_{i=1}^n x_i(t) \quad (8)$$

$$x_r = \text{Low} + r \times (\text{Up} - \text{Low}) \quad (9)$$

Where, p_1 and p_2 are random numbers within $(0, 1)$, $\text{fit}(x(i))$ is the fitness of the candidate position of the i th electric eel, x_j is the position of an eel chosen randomly from the current population and $j \neq i$, n is the size of the population, r_1 is a random number within $(0, 1)$, r is the random vector within $(0, 1)$, and, Low and Up are the lower and upper boundaries, respectively. According to Eq. (7), the interacting behavior enables electric eels to move toward different positions in the search space, which can contribute significantly to the exploration of EEFO in the entire search space.

3.2.2. Resting

In EEFO, the resting area should be established before electric eels perform resting behavior. To enhance the search efficiency, a resting area is established in the region where any one dimension of the position vector of an eel is projected onto the main diagonal in the search space. To identify a resting area for an eel, both the search space and the position of the eel are normalized to a range of 0–1. A randomly chosen dimension of the position of the eel is projected onto the main diagonal of the normalized search space. The projected position is considered the center of the resting area of the eel. Fig. 3 shows the procedures used to determine the resting area of an eel in 2-D and 3-D spaces, respectively. The resting area can be described as:

$$\{X | X - Z(t)| \leq \alpha_0 \times |Z(t) - x_{\text{prey}}(t)|\} \quad (10)$$

$$\alpha_0 = 2 \cdot (e - e^{\frac{t}{T}}) \quad (11)$$

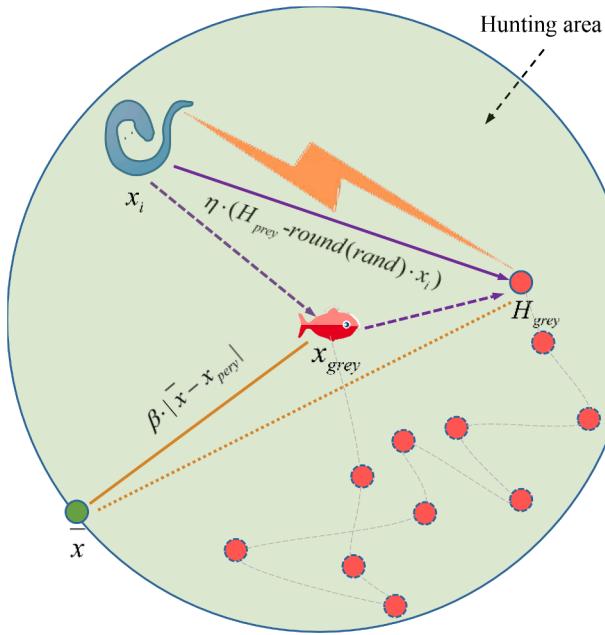


Fig. 4. Example of the curling hunting.

$$Z(t) = Low + z(t) \times (Up - Low) \quad (12)$$

$$z\{t = \frac{x_{rand(n)}^{rand(d)} \{t - Low^{rand(d)}}}{Up^{rand(d)} - Low^{rand(d)}} \quad (13)$$

Where x_{prey} is the position vector of the best solution obtained so far, α_0 is the initial scale of the resting area, the term $\alpha_0 \times |Z(t) - x_{prey}(t)|$ indicates the range of the resting area, $x_{rand(n)}^{rand(d)}$ is the random position dimension of a randomly chosen individual from the current population, z is the normalized number. So, the resting position of an eel is obtained within its resting area before performing resting behavior:

$$R_i(t+1) = Z(t) + \alpha \times |Z(t) - x_{prey}(t)| \quad (14)$$

$$\alpha = \alpha_0 \times \sin(2\pi r_2) \quad (15)$$

Where α is the scale of the resting area and r_2 is a random number within (0,1). The scale α enables the range of the resting area to decrease as the iterations proceed. This will enhance exploitation.

When the resting area is determined, eels will move to it to rest. That is, an eel updates its position toward the resting area with its resting position in its resting area. The resting behavior can be represented as:

$$v_i(t+1) = R_i(t+1) + n_2 \times (R_i(t+1) \text{round}(rand) \times x_i(t)) \quad (16)$$

$$n_2 \sim N(0, 1) \quad (17)$$

3.2.3. Hunting

When eels find prey, they do not simply swarm to hunt. Instead, they tend to cooperatively swim in the formation of a large circle and encircle the prey. Meanwhile, they constantly communicate and cooperate with their peers through low electric organ discharges. As the eels' interaction intensifies, the electrified circle decreases. Finally, eels drive the shoal of fish from the deeper end to the shallow end, where they are easy prey. Based on this behavior, the electrified circle becomes the hunting area, at this moment, the prey begins to run around in the hunting area, so the prey will suddenly and successively move from the current position to other positions in the hunting space because of being frightened. The hunting area can be defined as:

$$\{|X|X - x_{prey}(t)| \leq \beta_0 \times |\bar{x}(t) - x_{prey}(t)|\} \quad (18)$$

$$\beta_0 = 2 \times (e - e^{\frac{1}{2}}) \quad (19)$$

Where β_0 is the initial scale of the hunting area. From Eq. (18), an eel centers on the prey x_{prey} with its hunting range determined by the term $\beta_0 \times |\bar{x}(t) - x_{prey}(t)|$. So, a new position of the grey concerning its previous position within the hunting area can be generated as:

$$H_{prey}(t+1) = x_{prey}(t) + \beta \times |\bar{x}(t) - x_{prey}(t)| \quad (20)$$

$$\beta = \beta_0 \times \sin(2\pi r_3) \quad (21)$$

Where β is the scale of the hunting area and r_3 is a random number within (0,1). The scale β makes the range of the hunting area become smaller as time goes by. It is beneficial to exploitation.

After the hunting area is determined, an eel begins to perform prey in the hunting area. When hunting, the eel quickly locates the new position of the prey and curls to bring its head and tail together with the prey in between, and it will give off a high-voltage current around the grey (Catania, 2019; Catania, 2015b). The hunting behavior observed in EEFO involves a curling movement, whereby an eel's position is updated to the new position of the prey. The curling behavior exhibited by eels during hunting can be described as follows:

$$v_i(t+1) = H_{prey}(t+1) + \eta \times (H_{prey}(t+1) - \text{round}(rand) \times x_i(t)) \quad (22)$$

Where η indicate the curling factor, which is defined as:

$$\eta = e^{\frac{r_4(1-t)}{t}} \times \cos(2\pi r_4) \quad (23)$$

Where r_4 is a random number within (0,1).

The hunting behavior of an eel is demonstrated in Fig. 4. When the prey is circled by eels, some position footprints of the grey noted by the red dots are generated since the prey perform a swan dive. At this point, an eel shocks the prey by curling behavior and a position footprint is used to update the new position of the eel in the next iteration.

3.2.4. Migrating

When eels find prey, they tend to migrate from the resting area to the hunting area. To mathematically model the migrating behavior of eels, the following equation is used:

$$v_i(t+1) = -r_5 \times R_i(t+1) + r_6 \times H_r(t+1) - L \times (H_r(t+1) - x_i(t)) \quad (24)$$

$$H_r(t+1) = x_{prey}(t) + \beta \times |\bar{x}(t) - x_{prey}(t)| \quad (25)$$

Where H_r can be considered as any position within the hunting area, r_5 and r_6 are random numbers within (0,1). The term $(H_r(t+1) - x_i(t))$ indicates that eels move towards the hunting area. L is the Levy flight function, which is introduced to the exploitation phase of EEFO to avoid trapping into local optima. L can be obtained as (Viswanathan et al., 1996):

$$L = 0.01 \times \left| \frac{u \cdot \sigma}{|v|^{\frac{1}{b}}} \right| \quad (26)$$

$$u, v \sim N(0, 1) \quad (27)$$

$$\sigma = \left(\frac{\Gamma(1+b) \times \sin\left(\frac{\pi b}{2}\right)}{\Gamma\left(\frac{1+b}{2}\right) \times b \times 2^{\frac{b-1}{2}}} \right)^{\frac{1}{b}} \quad (28)$$

Where Γ is the standard Gamma function and $b = 1.5$. A pattern diagram of the migrating behavior of an eel is described in Fig. 5.

An eel can perceive the position of the prey via low electric discharge and thus it can adjust its own position at any moment. If the eel senses the approach of the prey in the foraging process, they move to the candidate position; otherwise, the eels stay at the current position. The

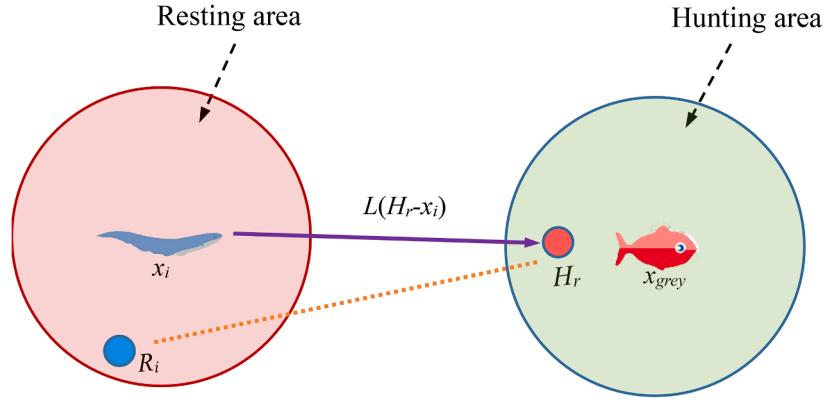
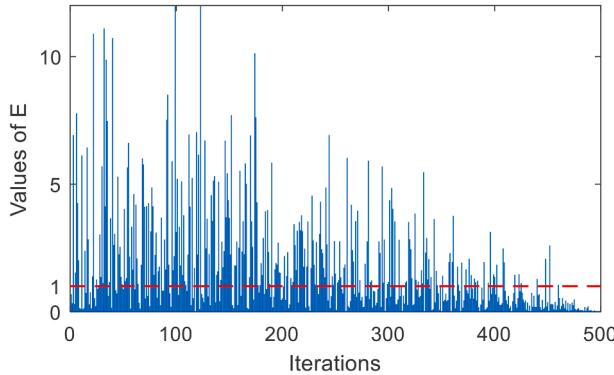


Fig. 5. Pattern diagram of migrating behavior.

Fig. 6. Behavior of E during 500 iterations.

positions of eels are updated by (Zhao et al., 2022b):

$$x_i(t+1) = \begin{cases} x_i(t) & fit(x_i(t)) \leq fit(v_i(t+1)) \\ v_i(t+1) & fit(x_i(t)) > fit(v_i(t+1)) \end{cases} \quad (29)$$

3.2.5. Transition from exploration to exploitation

In EEFO, search behaviors are determined by an energy factor, which can effectively manage the transition between exploration and exploitation to improve the algorithm's optimization performance (Wang et al., 2019; Heidari et al., 2019). The value of an eel's energy factor is used to choose between exploration and exploitation. The energy factor is defined as:

$$E(t) = 4 \times \sin\left(1 - \frac{t}{T}\right) \times \ln \frac{1}{r_7} \quad (30)$$

Where r_7 is a random number within $(0,1)$. From Eq. (30), $E(t)$ decreases as the iterations increase, and the energy factor $E(t)$ shows a decreasing trend with oscillation during the iterations. When the energy factor $E(t)$ is greater than 1, eels perform global search in the entire variable space by the interacting behavior, resulting in exploration. When the energy factor $E(t)$ is less than or equal to 1, eels tend to perform local search in a promising subregion by migrating, resting, or hunting behaviors, resulting in exploitation. In the first half of the iterations, the exploration occurs with a higher probability, while in the second half of the iterations, exploitation occurs with a higher probability. The behavior of E during 500 iterations is depicted in Fig. 6.

To study the search behavior of EEFO, the probability of $E > 1$ is evaluated throughout the optimization process. Let

$$\theta = 1 - \frac{t}{T} \quad (31)$$

Then

$$E(t) = \sin(\theta) \ln \frac{1}{r_7} \quad (32)$$

The probability of $E > 1$ is obtained by:

$$P\{E(t) > 1\} = \frac{\int_0^1 \int_0^{\frac{1}{\sqrt{4\sin(\theta)}}} dr d\theta}{1} = - \int_{-\infty}^{\frac{1}{\sqrt{4\sin(1)}}} \frac{e^x dx}{x\sqrt{16x^2 - 1}} \approx 0.5035 \quad (33)$$

According to the result in Eq. (33), there is a probability of approximately 50 % to perform between either exploration or exploitation during the optimization process. It contributes greatly to balancing exploration and exploitation.

3.2.6. Procedure of EEFO

EEFO starts by initializing several control parameters, including the population size of electric eels and the maximum number of iterations. Meanwhile, a set of eel population is randomly produced in a uniform distribution. At each iteration, when the energy factor $E > 1$, each eel performs exploration by using the interacting behavior. When the energy factor $E \leq 1$, each eel performs exploitation by using the resting behavior, migrating behavior, or hunting behavior with the same probability. Each case is applied to all eels to produce new candidate solutions, which are compared with the current solutions. Meanwhile, the best solution found so far is updated. E decreases as the iteration continues, which forces each eel to switch from exploration to exploitation. This procedure is interactively achieved until the stop condition is reached. Once this happens, the best solution found so far is saved. The flowchart and pseudo-code of EEFO are provided in Algorithm 1 and Fig. 7, respectively.

Algorithm 1 Pseudo-code of EEFO.

```

Set parameters  $n$  and  $T$ .
Randomly initialize the eel population  $X_i$  ( $i = 1, \dots, n$ ) and evaluate their fitness  $Fit_i$ , and  $X_{prey}$  is the best solution found so far.
While the stopping condition is not satisfied do
  For each eel  $X_i$  do
    Calculate  $E$  using Eq. (30).
    If  $E > 1$ 
      Perform the interacting behavior using Eq. (7).
      Evaluate the fitness  $Fit_i$ .
    Else
      If  $rand > 1/3$ 
        Determining the resting region using Eq. (14).
        Perform the resting behavior using Eq. (16).
        Evaluate the fitness  $Fit_i$ .
      Else If  $rand > 2/3$ 
        Perform the migrating behavior using Eq. (24).
      Else
        Determining the hunting region using Eq. (20).
        Perform the hunting behavior using Eq. (22).
    End If
  End For
End While

```

(continued on next page)

(continued)

Algorithm 1 Pseudo-code of EEFO.

```

End If
Update each eel's position using Eq. (29).
End For
Update the best solution found so far  $X_{prey}$ .
End While
Return  $X_{prey}$ .

```

3.3. Computational complexity analysis

To comprehensively understand and assess an optimizer, it is critical to examine the computational complexity since it determines the algorithms' efficiency. Some key parameters, including the number of electric eel individuals (n), the variable dimensionality of the considered problem (d), and the maximum number of iterations (T), affect the algorithm's computational complexity. The total computational complexity of EEFO is:

$$\begin{aligned}
O(\text{EEFO}) &= O(\text{problemdefinition}) + O(\text{initialization}) \\
&\quad + O(\text{functionevaluation}) + O(\text{positionupdatinginteracting}) \\
&\quad + O(\text{positionupdatinginresting}) + O(\text{positionupdatinginmigrating}) \\
&\quad + O(\text{positionupdatinginhunting}) \\
&= O(1) + O(n) + O(Tn) + O\left(\frac{1}{2}Tnd\right) + O\left(\frac{1}{6}Tnd\right) + O\left(\frac{1}{6}Tnd\right) + O\left(\frac{1}{6}Tnd\right) \\
&= O(Tnd + Tn + n + 1) \cong O(Tnd)
\end{aligned}$$

3.4. Conceptual comparison of EEFO with marine predators algorithm (MPA)

As two newly developed optimization algorithms, EEFO and MPA (Faramarzi et al., 2020a) share some similarities, which are listed below.

- EEFO and MPA are both swarm-based, bio-inspired metaheuristic algorithms. Thus, they exhibit population search characteristics such as interactivity and diversity.
- Both EEFO and MPA are inspired by foraging and movement behaviors of aquatic animals, and they both have memory capacity when performing optimization, allowing them to remember previous successful solutions and improve upon them.
- EEFO and MPA both use the Levy flight strategy to enhance their capability of avoiding local optimal solutions.

Despite their similarities, EEFO and MPA also exhibit some substantial differences, which are as follows.

- Their foraging behaviors are different. MPA simulates the common foraging behavior of marine animals, including three scenarios, scenario 1: in a high-velocity ratio or when the predator is moving faster than its prey; scenario 2: in a unit velocity ratio or when both the predator and prey are moving at the same speed; and scenario 3: in low-velocity ratio or when the predator is moving faster than its prey. However, EEFO simulates the unique foraging behaviors of electric eels as an independent species, including four foraging behaviors: interacting, resting, hunting, and migrating.
- Their search characteristics are different. MPA divides the entire iteration process evenly into three stages, with scenario 1 executed for exploration in the first stage, scenario 2 executed for switching from exploration to exploitation in the middle stage, and scenario 3 executed for exploitation in the third stage. In EEFO, the selection of search behaviors is controlled by the energy factor. When the energy factor is greater than 1, interaction is executed for exploration. When the energy factor is less than or equal to 1, one of the resting, hunting, and migrating behaviors is selected with equal probability and executed for exploitation. The energy factor is a variable that varies with iterations. In the early stages of iterations, the energy factor will enable EEFO to perform exploration with high probability. With the increase in iterations, the energy factor will force EEFO to perform exploration with high probability. In EEFO, the energy

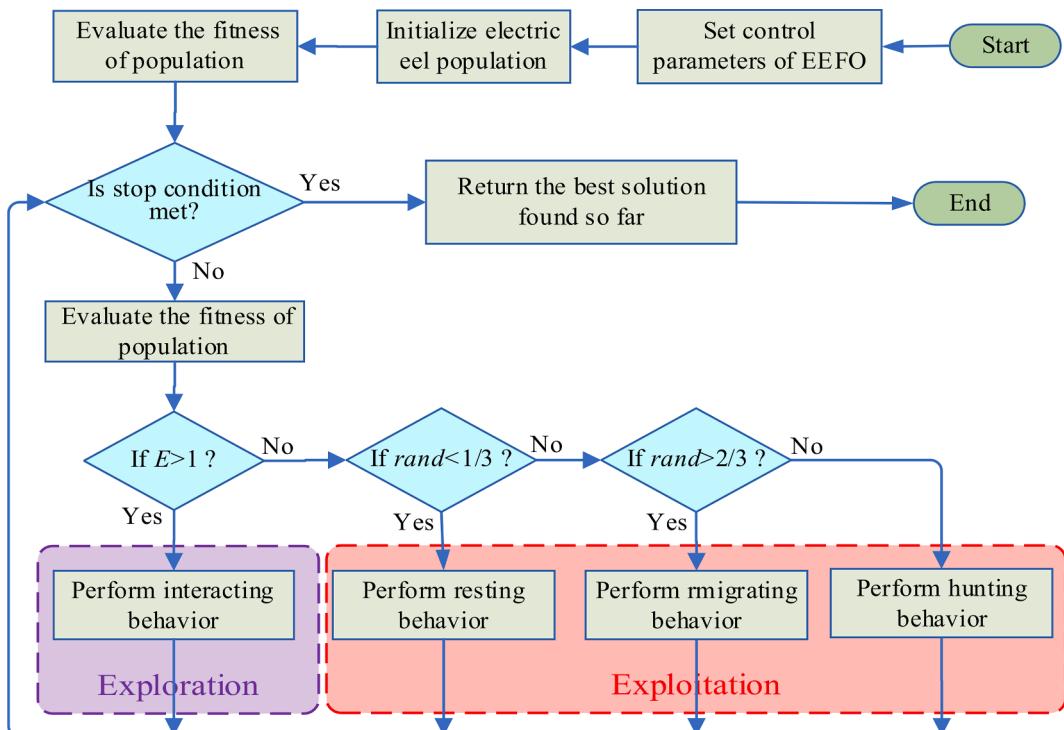
**Fig. 7.** Flowchart of EEFO.

Table 5
Comparisons of different algorithms for 23 functions.

Fun.	Index	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
F_1	AVE	3.105E-276	9.935E-09	3.258E-46	3.633E-84	8.5210	3.126E-112	2.686E-07	2.513E-25	2.636E-07	6.106E-01	3.905E-17	7.829E + 03	1.056E-18
	STD	0	4.625E-09	1.785E-45	1.677E-83	2.144E + 01	1.643E-111	3.613E-08	6.296E-25	1.185E-07	3.806E-01	1.208E-17	1.263E + 03	1.678E-18
F_2	AVE	2.755E-140	9.120E-03	0	5.716E-54	2.267E-02	1.744E-58	3.242E-05	3.067E-13	7.826E-04	7.311E-02	3.209E-08	3.553E + 01	6.128E-08
	STD	1.493E-139	6.027E-03	0	2.876E-53	3.723E-02	8.604E-58	4.778E-06	5.108E-13	2.916E-04	3.184E-02	6.079E-09	2.976E + 00	2.495E-08
F_3	AVE	9.135E-217	1.036E-04	3.959E-03	3.698E + 04	5.617E + 03	4.174E-104	1.688E-07	3.719E-19	4.575E + 03	1.987E + 03	5.801E + 02	2.481E + 04	2.576E + 03
	STD	0	8.708E-05	7.932E-03	1.193E + 04	3.972E + 03	1.554E-103	3.822E-08	4.156E-19	2.377E + 04	7.842E + 02	1.878E + 02	3.650E + 03	5.801E + 02
F_4	AVE	2.653E-132	3.566E-03	1.943E-02	4.027E + 01	3.967E + 01	8.103E-57	5.735E-05	1.876E-12	7.114E-03	3.237E + 01	3.429E + 00	5.212E + 01	2.73E-07
	STD	8.085E-132	1.944E-03	2.091E-02	3.103E + 01	1.343E + 01	3.083E-56	8.912E-06	1.682E-12	3.886E-03	7.035E + 00	1.573E + 00	3.364E + 00	4.28E-07
F_5	AVE	5.857E-10	2.796E + 01	2.835E + 01	2.754E + 01	1.967E + 04	2.691E-01	2.993E + 01	3.716E + 01	4.287E + 01	2.984E + 02	3.435E + 01	6.090E + 06	4.349E + 01
	STD	3.086E-10	1.547E-01	4.139E-01	4.654E-01	8.407E + 04	2.063E-01	3.399E-02	6.920E-02	3.259E + 01	2.609E + 02	2.135E + 01	1.991E + 06	5.243E + 01
F_6	AVE	0	0	0	0	3.9	0	0	0	0	6.467E + 00	4.333E-01	7.714E + 03	3.336E-02
	STD	0	0	0	0	5.1351	0	0	0	0	2.210E + 01	8.172E-01	1.219E + 03	1.838E-01
F_7	AVE	1.265E-04	1.021E-02	5.462E-05	1.736E-03	7.518E-02	2.701E-04	5.625E-03	2.364E-04	2.435E-02	1.062E-01	2.927E-02	3.613E + 00	4.761E-02
	STD	1.387E-04	6.808E-03	4.832E-05	3.177E-03	5.455E-02	2.601E-04	2.903E-03	3.154E-04	3.719E-03	5.365E-02	1.478E-02	5.914E-01	1.755E-02
F_8	AVE	-12569.4866	-8122.9497	-5611.6413	-11630.3628	-3916.1696	-12568.8759	-4474.0654	-5734.4837	-4313.6084	-9687.9218	-2830.4006	-7560.4686	-7134.6495
	STD	1.857E-12	5.663E + 02	4.147E + 02	1.484E + 03	3.595E + 02	9.941E-01	3.457E + 02	8.244E + 02	3.895E + 02	6.140E + 02	4.014E + 02	2.582E + 02	6.797E + 02
F_9	AVE	0	2.243E + 00	0	7.585E-15	5.173E + 01	0	1.026E + 02	6.427E + 01	2.545E + 02	6.148E + 01	1.681E + 01	1.652E + 02	8.737E + 01
	STD	0	7.129E + 00	0	2.88E-14	4.405E + 01	0	9.826E + 01	3.114E + 01	5.153E + 01	1.247E + 01	4.059E + 00	1.201E + 01	5.6267
F_{10}	AVE	8.88E-16	1.416E-04	8.882E-16	3.977E-15	1.517E + 01	8.882E-16	2.973E-05	3.778E-14	1.584E-04	1.371E + 00	4.844E-09	1.478E + 01	5.481E-10
	STD	0	3.014E-05	0	2.424E-15	8.3861	0	8.304E-06	6.205E-14	3.256E-05	1.379E + 00	6.463E-10	6.055E-01	4.786E-10
F_{11}	AVE	0	1.250E-03	1.232E-01	2.656E-03	7.294E-01	0	4.487E-08	1.766E-02	3.305E-06	4.468E-01	1.668E + 01	6.970E + 01	6.575E-04
	STD	0	3.968E-03	1.016E-01	2.563E-02	3.267E-01	0	2.416E-08	4.597E-02	1.935E-06	2.428E-01	4.146E + 00	9.351E + 00	2.585E-03
F_{12}	AVE	1.578E-32	2.013E-03	4.314E-01	7.615E-03	7.926E + 01	5.652E-05	3.555E-01	5.126E-02	2.778E-08	2.313E + 00	5.782E-01	2.352E + 06	6.917E-03
	STD	5.574E-48	1.149E-03	4.284E-02	4.946E-03	3.915E + 02	5.125E-05	6.003E-02	2.466E-01	8.115E-09	1.710E + 00	4.662E-01	1.290E + 06	3.793E-02
F_{13}	AVE	1.358E-32	6.492E-01	2.816E + 00	3.287E-01	5.318E + 04	1.031E-03	3.2518	3.735E-01	4.676E-04	4.187E + 00	1.625E + 00	1.274E + 07	2.058E-03
	STD	5.57E-48	1.135E + 00	8.517E-02	2.714E-01	2.838E + 05	1.111E-03	4.247E-01	7.185E-01	3.289E-03	5.087E + 00	2.139E + 00	3.959E + 06	4.245E-03
F_{14}	AVE	0.9980	1.0981	7.8984	1.3939	1.4999	0.9980	0.9983	5.9470	4.5545	1.1635	5.8568	0.9980	1.2062
	STD	1.091E-16	3.055E-01	4.526E + 00	1.0571	8.586E-01	2.523E-10	1.124E-03	4.8759	3.3743	4.578E-01	3.323E + 00	9.942E-06	4.094E-01
F_{15}	AVE	3.077E-04	1.016E-03	1.015E-02	7.747E-04	8.316E-04	3.584E-04	4.721E-04	2.884E-03	3.116E-03	5.995E-04	4.559E-03	7.652E-04	2.876E-03
	STD	3.833E-19	3.906E-04	2.258E-02	5.044E-04	3.343E-04	1.658E-04	7.795E-05	4.724E-03	2.673E-03	1.053E-04	3.525E-03	1.015E-04	7.028E-04
F_{16}	AVE	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	STD	1.256E-05	1.109E-10	9.129E-08	2.129E-10	2.417E-05	4.865E-07	1.945E-05	9.198E-06	6.789E-16	6.775E-16	5.133E-16	2.435E-05	6.453E-16
F_{17}	AVE	0.3979	0.3979	4.0468	0.3979	0.3992	0.3979	0.3981	0.3979	0.3983	0.3979	0.3979	0.3981	0.3979
	STD	6.551E-05	9.107E-10	8.660E-03	1.236E-06	1.627E-03	3.544E-05	1.905E-04	2.6-05	1.147E-03	0	0	2.860E-04	0
F_{18}	AVE	3.0000	3.0281	9.3020	3.0000	3.0000	3.9000	3.0031	3.0008	3.0006	3.0000	3.0000	3.0001	3.0000
	STD	1.937E-15	3.653E-02	1.161E + 01	5.854E-06	2.747E-05	4.929E + 00	4.169E-03	1.258E-03	4.157E-03	1.106E-15	3.625E-15	1.325E-04	1.725E-15
F_{19}	AVE	-3.8628	-3.8609	-3.8535	-3.8606	-3.8556	-3.8587	-3.8616	-3.8627	-3.8622	-3.8628	-3.8628	-3.8628	-3.8628
	STD	2.656E-15	1.443E-03	3.528E-03	3.686E-03	2.765E-03	4.934E-03	1.284E-03	5.943E-05	1.876E-03	2.710E-15	2.387E-15	9.393E-07	2.675E-15
F_{20}	AVE	-3.3180	-3.1933	-3.1018	-3.2706	-2.9831	-3.2710	-3.2313	-3.2590	-3.2943	-3.3022	-3.3220	-3.3103	-3.3220
	STD	2.176E-02	5.602E-02	8.994E-02	7.026E-02	1.528E-01	7.329E-02	5.667E-02	5.965E-02	5.114E-02	4.507E-02	1.355E-15	5.800E-03	1.348E-15
F_{21}	AVE	-10.1532	-8.8035	-4.1183	-9.3243	-3.0664	-10.1506	-9.4426	-6.3138	-6.9707	-8.2292	-6.0837	-9.5987	-6.8261
	STD	6.394E-15	2.546E + 00	7.805E-01	2.0103	1.9254	3.301E-03	4.875E-01	3.5163	3.5669	3.065E + 00	3.655E + 00	3.482E-01	3.6737
F_{22}	AVE	-10.4029	-8.3720	-4.0324	-8.8575	-3.9739	-10.4002	-9.5990	-7.8353	-10.1484	-9.3979	-10.1943	-9.9010	-10.0927
	STD	7.384E-16	2.961E + 00	1.518E + 00	2.6315	1.7690	4.011E-03	0.5692	3.4588	1.3943	2.321E + 00	1.1428	3.821E-01	1.1932
F_{23}	AVE	-10.5364	-8.0372	-4.1201	-7.7063	-4.4570	-10.5336	-9.7614	-6.6647	-10.3130	-9.3553	-10.5364	-9.9881	-10.4202
	STD	2.569E-15	3.151E + 00	1.638E + 00	3.1546	1.6015	3.654E-03	4.913E-01	3.7625	1.2234	2.693E + 00	2.295E-15	3.865E-01	6.376E-01

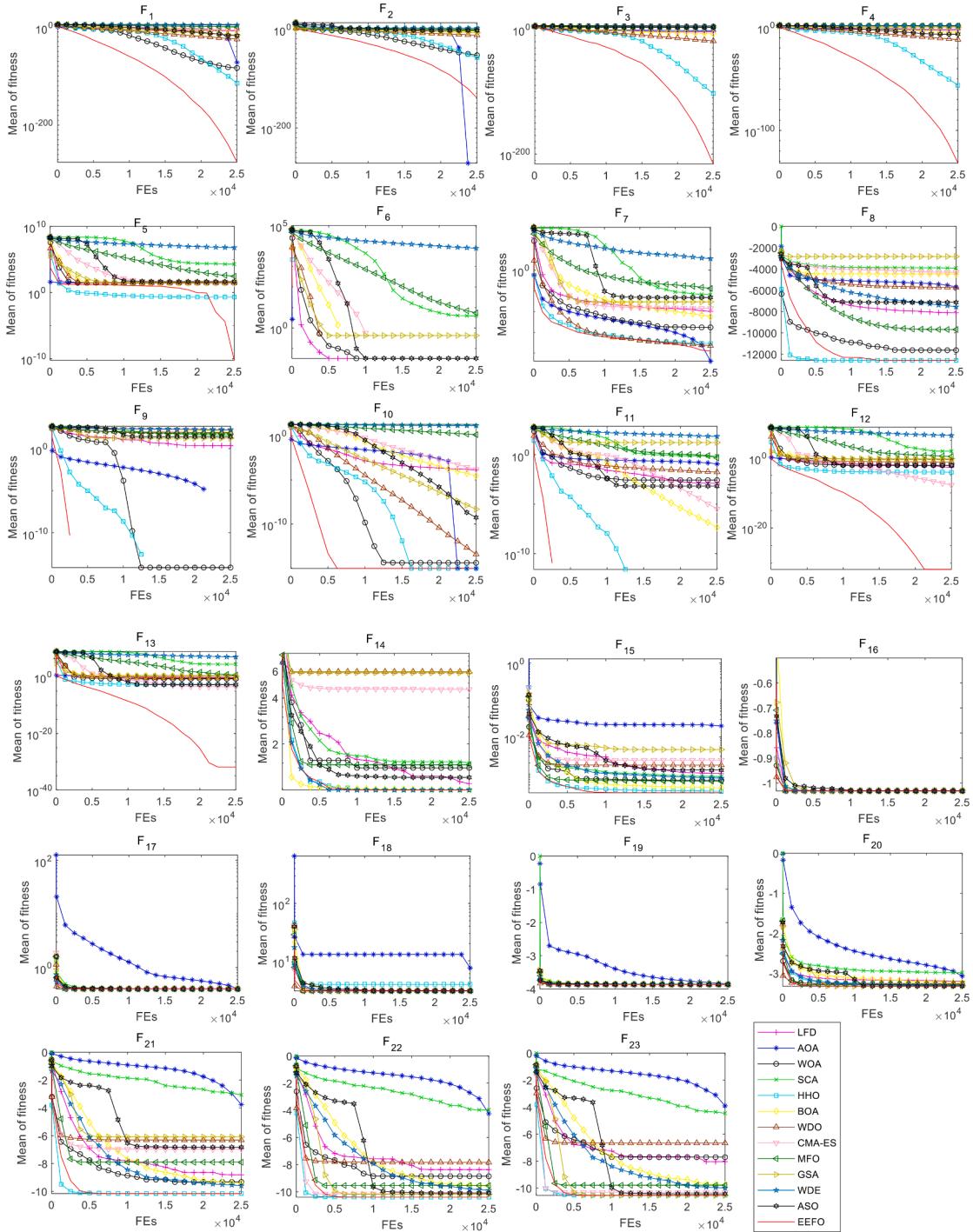


Fig. 8. Comparisons of convergence curve variations of all algorithms for 23 functions.

- factor determines the algorithm's exploration, exploitation, and balancing capacities.
- MPA has two control parameters that are sensitive to the optimization performance, while EEFO is a parameter-free optimizer and easy to implement.

4. Experiment and results

4.1. Numerical benchmarks and compared algorithms

The performance of EEFO is evaluated using three well-studied sets of numerical benchmarks: (1) 23 classical test functions (Mirjalili et al.,

2014; Kaveh et al., 2020), (2) 30 CEC2017 test functions (Wu et al., 2017), and (3) 22 CEC2011 test functions (Das & Suganthan, 2010). The first benchmark set includes 7 unimodal functions (UFs) (F₁-F₇) and 16 multimodal functions (MFs) (F₈-F₂₃). The UFs with only one global optimum are used to demonstrate the exploitation of algorithms, while the MFs with numerous local optima are utilized to test exploration and local optima avoidance. Appendix A.1 describes the mathematical formulas of UFs and MFs. The second benchmark set covers different characteristics: hybrid composite, rotated, and shifted MFs. Therefore, these functions can comprehensively demonstrate the search ability of algorithms, including exploration, exploitation, exploration-exploitation tradeoff, and local optima avoidance. It would be

Table 6

Comparisons of Wilcoxon test for EEFO vs LFD, AOA, WOA, and SCA.

Fun.	LFD vs EEFO				AOA vs EEFO				WOA vs EEFO				SCA vs EEFO			
	p-value	T-	T+	Winner	p-value	T-	T+	Winner	p-value	T-	T+	Winner	p-value	T-	T+	Winner
F ₁	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₂	1.734E-06	0	465	+	1.734E-06	465	0	-	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₃	1.734E-06	0	465	+	1.921E-06	1	464	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₄	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₅	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₆	1	0	465	=	1	0	465	=	1	0	465	=	5.708E-05	0	465	+
F ₇	1.734E-06	0	465	+	1.382E-03	388	77	-	2.353E-06	3	462	+	1.734E-06	0	465	+
F ₈	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₉	1.734E-06	0	465	+	1	0	465	=	5.000E-01	0	465	=	1.734E-06	0	465	+
F ₁₀	1.734E-06	0	465	+	1	0	465	=	2.184E-05	0	465	+	1.734E-06	0	465	+
F ₁₁	1.734E-06	0	465	+	1.734E-06	0	465	+	1	0	465	=	1.7344E-06	0	465	+
F ₁₂	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₃	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₄	1.909E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₅	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₆	2.597E-05	437	28	-	1.204E-01	308	157	=	2.597E-05	437	28	-	1.799E-05	24	441	+
F ₁₇	1.734E-06	465	0	-	1.734E-06	0	465	+	5.752E-06	453	12	-	2.879E-06	5	460	+
F ₁₈	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₉	1.734E-06	0	465	+	1.734E-06	0	465	+	1.7344E-06	0	465	+	1.734E-06	0	465	+
F ₂₀	1.734E-06	0	465	+	1.734E-06	0	465	+	1.360E-05	21	444	+	1.734E-06	0	465	+
F ₂₁	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₂₂	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₂₃	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.7344E-06	0	465	+

particularly challenging for an algorithm to obtain high-quality solutions to these complex functions. The third benchmark set includes 22 various real-world engineering problems with distinct functional characteristics.

The performance of EEFO is compared with that of other well-established optimizers, including LFD (Houssein et al., 2020), AOA (Abualigah et al., 2021), WOA (Mirjalili & Lewis 2016), SCA (Mirjalili, 2016), HHO (Heidari et al., 2019), (Arora & Singh, 2019) BOA, (Bayraktar et al., 2013) WDO, evolution strategy with covariance matrix adaptation (CMA-ES) (Hansen et al., 2003), MFO (Mirjalili, 2015), GSA (Rashedi et al., 2009), weighted differential evolution (WDE) (Civicioglu et al., 2020), and ASO (Zhao et al., 2019). Among them, WOA, SCA, HHO, BOA, MFO, AOA, WDO, LFD, GSA, and ASO are recently developed potential optimizers; CMA-ES and WDE are two improved evolutionary algorithms. For the EEFO and WDE algorithms, only the population size and maximum number of iterations need to be adjusted, while the parameters for other compared algorithms are set according to recommendations in the literature (Houssein et al., 2020; Abualigah et al., 2021; Mirjalili & Lewis 2016; Mirjalili, 2016; Arora & Singh, 2019; Heidari et al., 2019; Bayraktar et al., 2013; Hansen et al., 2003; Mirjalili, 2015; Rashedi et al., 2009; Civicioglu et al., 2020; Zhao et al., 2019). For the 23 test functions and the CEC2017 test suite, the population size and the maximum number of fitness evaluations (FEs) of all the algorithms are set to 50 and 25000, respectively. For the CEC2011 test suite, the population size and the maximum number of fitness evaluations (FEs) of all the algorithms are set to 100 and 150000, respectively. All the quantitative results are based on 30 independent runs. All the algorithms were carried out using a MATLAB 9.7 (R2019b) desktop computer running Windows 10 64-bit with an Intel(R) Core(TM) i5-9400F CPU 2.9 GHz processor and 8.00 GB RAM.

4.2. Quantitative analysis

The solution quality of EEFO is quantitatively investigated in this subsection for the 23 test functions. The average (AVE) and standard deviation (STD) of the best solutions provided by all compared algorithms over 30 runs for the 23 benchmark functions are listed in Table 5. The best results for all test functions in the table are highlighted in bold. The UFs with only one optimum play the primary role in verifying the exploitation performance of the algorithms. According to the results

from EEFO and the other optimizers on 7 UFs (F₁- F₇) in Table 5, EEFO outperforms other competitors concerning AVE and STD for all UFs except for F₂ and F₇. The performance of EEFO is second only to that of HHO on F₂ and F₇. Accordingly, it is evident that EEFO shows higher exploitation compared to its competing algorithms in handling UFs. From the results of 16 MFs (F₈-F₂₃) in Table 5, EEFO achieves the best AVE in 6 MFs including F₈, F₁₂, F₁₃, F₁₅, F₂₁ and F₂₂. For 9 MFs including F₉-F₁₁, F₁₄, F₁₆-F₁₉, and F₂₃, EEFO and several competing optimizers perform similarly concerning AVE, which outperforms that of the remaining algorithms. In addition, EEFO shows the smallest STD in 7 MFs, including F₈, F₁₂-F₁₅, F₂₁ and F₂₂. In general, EEFO demonstrates higher exploration than its competitors in solving MFs F₈ to F₂₃.

4.3. Convergence analysis

Fig. 8 shows the convergence curve variations of EEFO and other optimizers. For UFs F₁ to F₇, compared with different algorithms, EEFO provides faster convergence speed toward the final solutions with better accuracy; this is because to that EEFO can easily position the optimal solution and quickly converge to it when local optima are absent. These curves on the UFs reveal excellent local search and exploitative ability of EEFO. For MFs F₈ to F₁₃, due to numerous local optima resulting from the higher dimensionality of variables, EEFO tends to perform more global searches to find the promising region in the initial iterative process. With the increase in iterations, EEFO gradually shifts to local search in the promising region, and EEFO converges toward the global optimum in the final iterative process. This searching characteristic is evident in F₁₂ and F₁₃. The MFs F₁₄ to F₂₃ have a smaller number of local optima owing to fixed-dimensional variables. Accordingly, EEFO needs less global search, and positioning the optimum solution is more accessible compared to F₈ to F₁₃ in the initial iterative process; thus, it performs local searches relatively quickly. This behavior can be observed in MFs F₁₄ to F₂₃. Convergence analysis based on Fig. 8 reveals that EEFO exhibits better exploration and exploitation capabilities with a higher convergence rate during the optimization process in solving the UFs and MFs compared to its competitors.

4.4. Statistic analysis

The Wilcoxon test (Derrac et al., 2011) with 5 % significance is

Table 7

Comparisons of Wilcoxon test for EEFO vs HHO, BOA, WDO, and CMA-ES.

Fun.	HHO vs EEFO				BOA vs EEFO				WDO vs EEFO				CMA-ES vs EEFO			
	p-value	T-	T+	Winner	p-value	T-	T+	Winner	p-value	T-	T+	Winner	p-value	T-	T+	Winner
F ₁	1.734E-06	0	465	+	1.734E-06	0	465	+	7.687E-06	15	450	+	1.734E-06	0	465	+
F ₂	1.734E-06	0	465	+	1.734E-06	0	465	+	2.352E-06	3	462	+	1.734E-06	0	465	+
F ₃	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₄	1.734E-06	0	465	+	1.734E-06	0	465	+	1.728E-06	0	465	+	1.734E-06	0	465	+
F ₅	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₆	1.000E + 00	0	465	=	1.000E + 00	0	465	=	1.000E + 00	0	465	=	1	0	465	=
F ₇	1.965E-03	82	383	+	1.734E-06	0	465	+	3.160E-02	128	337	+	1.734E-06	0	465	+
F ₈	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₉	1.000E + 00	0	465	=	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₀	1.000E + 00	0	465	=	1.734E-06	0	465	+	8.708E-05	0	465	+	1.734E-06	0	465	+
F ₁₁	1.000E + 00	0	465	=	1.734E-06	0	465	+	0.015625	0	465	+	1.7344E-06	0	465	+
F ₁₂	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₃	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₄	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.733E-06	0	465	+
F ₁₅	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₆	3.086E-01	282	183	=	2.412E-04	54	411	+	5.667E-03	98	367	+	8.270E-06	351	114	-
F ₁₇	7.190E-02	320	145	=	1.921E-06	1	464	+	2.623E-01	178	287	=	1.480E-02	351	114	-
F ₁₈	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	9.605E-03	174	291	+
F ₁₉	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	2.500E-01	0	465	=
F ₂₀	9.316E-06	17	448	+	6.339E-06	13	452	+	1.734E-06	0	465	+	2.668E-01	39.5	425.5	=
F ₂₁	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	4.325E-03	45	420	+
F ₂₂	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	2.386E-04	210	255	+
F ₂₃	1.734E-06	0	465	+	1.73E-06	0	465	+	1.7344E-06	0	465	+	3.320E-02	66	399	+

Table 8

Comparisons of Wilcoxon test for EEFO vs MFO, GSA, WDE, and ASO.

Fun.	MFO vs EEFO				GSA vs EEFO				WDE vs EEFO				ASO vs EEFO			
	p-value	T-	T+	Winner	p-value	T-	T+	Winner	p-value	T-	T+	Winner	p-value	T-	T+	Winner
F ₁	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₂	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₃	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₄	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₅	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₆	1.58E-05	0	465	+	1.953E-03	0	465	+	1.733E-06	0	465	+	1.000E + 00	0	465	=
F ₇	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₈	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₉	1.73E-06	0	465	+	1.718E-06	0	465	+	1.734E-06	0	465	+	1.721E-06	0	465	+
F ₁₀	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₁	1.73E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	2.500E-01	0	465	=
F ₁₂	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₃	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.732E-06	0	465	+
F ₁₄	0.03125	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	3.787E-06	0	465	+
F ₁₅	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+	1.734E-06	0	465	+
F ₁₆	2.702E-05	276	189	-	1.322E-05	347	118	-	1.359E-04	47	418	+	8.270E-06	351	114	-
F ₁₇	1.734E-06	465	0	-	1.734E-06	465	0	-	3.112E-05	30	435	+	1.734E-06	465	0	-
F ₁₈	8.084E-01	72	393	=	1.682E-05	10.5	454.5	+	1.734E-06	0	465	+	2.935E-02	98.5	366.5	+
F ₁₉	1	0	465	=	2.727E-06	0	465	+	1.734E-06	0	465	+	2.500E-01	0	465	=
F ₂₀	3.906E-03	0	465	+	1.000E + 00	1	464	=	3.112E-05	30	435	+	5.000E-01	3	462	=
F ₂₁	1.953E-03	0	465	+	2.754E-05	4	461	+	1.734E-06	0	465	+	7.385E-04	33	432	+
F ₂₂	1.250E-01	0	465	=	2.441E-04	0	465	+	1.734E-06	0	465	+	4.688E-02	3	462	+
F ₂₃	2.500E-01	0	465	=	1.080E-05	0	465	+	1.734E-06	0	465	+	1.191E-01	18	447	=

utilized to determine the significant difference between EEFO and other considered optimizers. The Wilcoxon test results of all the optimizers for 23 functions in 30 runs are listed in [Tables 6–8](#). In the tables, the *p* value indicates the significance difference between two results. '+' signifies that EEFO has significant difference with a *p* value of 5 % with the competing algorithm, and '=' indicates the opposite. '=' indicates that there is no significant difference between the performance of EEFO and that of the competing algorithms. [Table 9](#) summarizes the results of the Wilcoxon test. According to the statistically significant results from [Table 9](#), among the 276 piecewise comparisons, the EEFO algorithm outperforms the other algorithms in 232 (84.06 %) of them. For 30 (10.87 %) comparisons, EEFO and the compared algorithms show similar performance. For only 14 comparisons (5.07 %), other algorithms perform better than EEFO. Obviously, EEFO offers significant

improvement over all other optimizers for both UFs and MFs.

The Friedman test ([Friedman, 1937](#)) can assess the performance rank of EEFO among all considered algorithms. In [Table 10](#), the ranking results of EEFO and other considered algorithms based on the mean of the best solutions from [Table 5](#) are summarized using the Friedman test with a significance level of $\alpha = 0.05$. The *p* value calculated for the results in [Table 10](#) is 5.6268E-15, indicating that there is a statistically significant difference in the performances of EEFO and other compared algorithms. According to [Table 10](#), EEFO obtains the best rank, followed by HHO, ASO, WOA, LFD, GSA, BOA, CMA-ES, WDO, MFO, AOA, WDE and SCA in descending order. It can also be seen that HHO, ASO and WOA have outstanding performance with relatively high rankings. However, according to the mean ranks of EEFO (2.20), HHO (3.87), ASO (5.85) and WOA (6.15) in [Table 10](#), there is still a large gap between EEFO and each

Table 9
Statistical results of Wilcoxon test based on Tables 6–8.

Function type	LFD vs EEFO (+/-=)	AOA vs EEFO (+/-=)	WOA vs EEFO (+/-=)	SCA vs EEFO (+/-=)	HHO vs EEFO (+/-=)	BOA vs EEFO (+/-=)
UFs	6/1/0	4/1/2	6/1/0	7/0/0	6/1/0	6/1/0
MFs	14/0/2	13/3/0	12/2/2	16/0/0	11/5/0	16/0/0
Total	20/1/2	17/4/2	18/3/2	23/0/0	17/6/0	22/1/0
Function type	WDO vs EEFO (+/-=)	CMA-ES vs EEFO (+/-=)	MFO vs EEFO (+/-=)	GSA vs EEFO (+/-=)	WDE vs EEFO (+/-=)	ASO vs EEFO (+/-=)
UFs	6/1/0	6/1/0	7/0/0	7/0/0	7/0/0	6/1/0
MFs	15/1/0	12/2/2	10/4/2	13/1/2	16/0/0	10/4/2
Total	21/2/0	18/3/2	17/4/2	20/1/2	23/0/0	16/5/2

of other three algorithms. This reveals that EEFO can surpass promising optimization algorithms by achieving significantly better performance than many other competitive algorithms.

The Nemenyi post-hoc test (Derrac et al., 2014), a powerful nonparametric method, was used to rank EEFO and the above 12 algorithms. The Nemenyi post-hoc can identify whether there is a significant difference between EEFO and each of the compared algorithms. If the difference in the average ranking values of the two algorithms exceeds the critical difference (CD), there is a significant difference between the two algorithms. Otherwise, there is no significant difference between the two algorithms. The CD can be given as (Nemenyi, 1963; Veysari, 2022)

$$CD = q_a \sqrt{\frac{k(k+1)}{6N}} \quad (34)$$

Where q_a is the critical value, which can be obtained from a predetermined table, k is the number of compared algorithms, and N is the total number of test functions. According to the mean ranks of each algorithm for 23 functions in the Friedman test, the Nemenyi post-hoc test is conducted and the statistical results of different algorithms are listed in Table 11, in which '1' represents a significant difference between two algorithms and '0' otherwise. One can observe that there exists a significant difference between the EEFO algorithm and the other 11 algorithms, followed by SCA and HHO with 9 and 4 other algorithms, respectively. LFD, AOA, WOA, BOA, WDO, CMA-ES, MFO and GSA have significant differences with only one algorithm; this indicates that EEFO

is the best-performing optimization algorithm among all algorithms for this test suite.

4.5. CEC2017 test function analysis

To test the effectiveness of EEFO in tackling complex optimization problems, a comprehensive suite of 30 functions is employed (Awad et al., 2017). This test suite includes two unimodal (C17F₁ and C17F₃), seven multimodal (C17F₄-C17F₁₀), 10 hybrid (C17F₁₁-C17F₂₀), and 10 composition functions (C17F₂₁-C17F₃₀). This section presents the CEC2017 test suite with dimensions of 10, 30, and 50. Tables 12, 13, and 14 display the mean and standard deviation of the best solutions obtained by EEFO and other compared algorithms on the CEC2017 test suite at each dimensionality. The best results for all test problems in these tables are highlighted in bold.

The convergence curves of EEFO and other algorithms for the CEC2017 test suite with 10, 30 and 50 dimensions are depicted in Figs. 9–11. From these figures, EEFO exhibits two types of convergence behaviors. First, throughout the iterative process, EEFO demonstrates an accelerating convergence rate and successfully converges to the optimum. This implies that EEFO has the capability to locate promising regions in the initial iterations and subsequently expedite the convergence. This effect is considered a result of the resting, hunting, and migrating behaviors serving as an exploratory search. These behaviors enable the algorithm to explore the search space in search of promising regions during the initial iterations and accelerate convergence toward the optimum once nearly half of the iterations have been completed. This behavior is evident in C17F₃, C17F₄, C17F₆, C17F₁₀, C17F₁₁, and C17F₁₃-C17F₁₅ for 10 dimensions, in C17F₄, C17F₁₁-C17F₁₃, and C17F₁₅ for 30 dimensions, and in C17F₁₁, C17F₁₃, and C17F₁₅ for 50 dimensions.

The second behavior of EEFO is characterized by its frequent failure to obtain promising regions for exploitation during more than half of the iterations, which ultimately results in convergence toward optimal solutions only in the final iterations. The observed effect may likely be attributed to the interplay between the interacting behavior and the energy factor in EEFO. These components play a crucial role in facilitating the algorithm's global search capability and balancing the trade-off between exploration and exploitation when tackling the composition functions. This behavior is evident in C17F₁₂, C17F₂₀, C17F₂₁, and C17F₃₀ for 10 dimensions, in C17F₁₄, C17F₁₈, and C17F₃₀ for 30 dimensions.

Table 10
Ranks of all algorithms using Friedman test for 23 functions.

	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
Sum of rank	50.5	179.5	204	153.5	271	89.5	181.5	183.5	185.5	101	184.5	149	153.5
Mean of rank	2.20	6.93	8.33	6.15	11.13	3.87	7.24	7.33	7.30	7.91	7.15	9.61	5.85
Overall rank	1	5	11	4	13	2	7	9	8	10	6	12	3

Table 11
Statistical results of different algorithms using Nemenyi test for 23 functions.

	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
EEFO	NaN	1	1	1	1	0	1	1	1	1	1	1	1
LFD	1	NaN	0	0	1	0	0	0	0	0	0	0	0
AOA	1	0	NaN	0	0	1	0	0	0	0	0	0	0
WOA	1	0	0	NaN	1	0	0	0	0	0	0	0	0
SCA	1	1	0	1	NaN	1	1	1	1	0	1	0	1
HHO	0	0	1	0	1	NaN	0	0	0	1	0	1	0
BOA	1	0	0	0	1	0	NaN	0	0	0	0	0	0
WDO	1	0	0	0	1	0	0	NaN	0	0	0	0	0
CMA-ES	1	0	0	0	1	0	0	0	NaN	0	0	0	0
MFO	1	0	0	0	0	1	0	0	0	NaN	0	0	0
GSA	1	0	0	0	1	0	0	0	0	0	NaN	0	0
WDE	1	0	0	0	0	1	0	0	0	0	0	NaN	1
ASO	1	0	0	0	1	0	0	0	0	0	0	1	NaN
Sum of significant difference	11	2	2	2	9	4	2	2	2	2	2	3	3

Table 12

Results of EEFO and other algorithms for CEC2017 test suite with 10 dimensions.

Fun.	Index	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
<i>C17F₁</i>	AVE	1.468E + 03	1.837E + 08	7.126E + 09	1.023E + 07	8.008E + 08	6.969E + 08	8.482E + 08	8.614E + 06	2.322E + 08	6.711E + 07	2.808E + 02	6.944E + 07	6.413E + 02
	STD	1.490E + 03	5.860E + 08	3.342E + 09	1.104E + 07	4.140E + 08	9.645E + 08	4.074E + 08	9.172E + 06	7.443E + 08	3.545E + 04	2.357E + 04	2.139E + 03	6.553E + 02
<i>C17F₃</i>	AVE	3.012E + 02	2.515E + 03	1.067E + 04	5.022E + 03	1.761E + 03	4.506E + 03	1.271E + 03	4.042E + 02	1.117E + 04	5.596E + 03	1.359E + 04	6.405E + 03	8.027E + 02
	STD	2.672E + 00	1.299E + 03	3.338E + 03	4.388E + 03	8.025E + 02	1.788E + 02	5.719E + 03	1.114E + 02	3.613E + 03	7.797E + 03	3.123E + 03	1.999E + 03	6.221E + 02
<i>C17F₄</i>	AVE	4.042E + 02	4.506E + 02	8.864E + 02	4.623E + 02	4.534E + 02	4.831E + 02	5.026E + 02	4.164E + 02	4.295E + 02	4.196E + 02	4.170E + 02	4.166E + 02	4.054E + 02
	STD	1.162E + 00	4.909E + 01	3.105E + 02	4.789E + 01	2.460E + 01	8.866E + 01	4.851E + 01	1.942E + 01	8.294E + 01	2.547E + 01	1.740E + 01	3.551E + 00	6.936E-01
<i>C17F₅</i>	AVE	5.122E + 02	5.432E + 02	5.543E + 02	5.532E + 02	5.487E + 02	5.568E + 02	5.758E + 02	5.432E + 02	5.298E + 02	5.303E + 02	5.694E + 02	5.295E + 02	5.082E
	STD	5.352E + 00	1.525E + 01	1.616E + 01	2.343E + 01	8.161E + 01	1.931E + 01	7.739E + 01	1.382E + 01	1.598E + 01	9.995E + 00	1.420E + 01	3.424E + 00	2.677E
<i>C17F₆</i>	AVE	6.000E + 02	6.315E + 02	6.363E + 02	6.346E + 02	6.197E + 02	6.315E + 02	6.245E + 02	6.197E + 02	6.054E + 02	6.014E + 02	6.379E + 02	6.081E + 02	6.000E
	STD	5.611E-02 + 00	7.732E + 00	8.613E + 00	1.292E + 00	4.036E + 00	1.179E + 00	6.186E + 00	1.324E + 01	1.285E + 01	2.139E + 00	7.610E + 00	1.891E + 00	5.806E-05
<i>C17F₇</i>	AVE	7.248E + 02	7.930E + 02	8.009E + 02	7.801E + 02	7.810E + 02	7.916E + 02	7.858E + 02	7.460E + 02	7.353E + 02	7.319E + 02	7.245E + 02	7.576E + 02	7.154E
	STD	6.236E + 00	2.498E + 01	1.385E + 01	2.205E + 01	1.065E + 01	2.170E + 02	1.253E + 02	9.972E + 02	4.650E + 01	1.127E + 01	6.454E + 00	6.994E + 00	2.259E
<i>C17F₈</i>	AVE	8.125E + 02	8.398E + 02	8.350E + 02	8.448E + 02	8.437E + 02	8.373E + 02	8.499E + 02	8.289E + 02	8.221E + 02	8.301E + 02	8.222E + 02	8.278E + 02	8.074E
	STD	4.240E + 00	1.368E + 01	8.305E + 01	1.887E + 01	7.040E + 01	9.081E + 00	7.079E + 00	9.144E + 00	9.763E + 00	1.340E + 01	4.566E + 00	4.896E + 00	3.968E
<i>C17F₉</i>	AVE	9.006E + 02	1.444E + 03	1.411E + 03	1.506E + 03	1.025E + 03	1.384E + 03	1.016E + 03	9.938E + 03	9.000E + 02	9.846E + 02	1.049E + 02	9.897E + 02	9.000E
	STD	1.091E + 00	2.625E + 02	1.845E + 02	4.485E + 02	6.817E + 02	2.318E + 02	7.036E + 02	1.223E + 02	0.000E + 02	1.537E + 02	1.091E + 02	3.227E + 02	0.000E
<i>C17F₁₀</i>	AVE	1.428E + 03	2.033E + 03	2.124E + 03	2.133E + 03	2.418E + 03	2.205E + 03	2.513E + 03	2.299E + 03	2.516E + 03	1.897E + 03	2.807E + 03	1.915E + 03	1.731E
	STD	2.049E + 02	2.447E + 02	2.649E + 02	3.792E + 02	2.520E + 02	1.983E + 02	1.954E + 02	3.630E + 02	3.397E + 02	3.883E + 02	2.731E + 02	1.602E + 02	4.001E
<i>C17F₁₁</i>	AVE	1.107E + 03	1.237E + 03	2.749E + 03	1.225E + 03	1.248E + 03	1.251E + 03	1.308E + 03	1.148E + 03	1.315E + 03	1.143E + 03	1.668E + 03	1.133E + 03	1.128E
	STD	4.361E + 00	9.160E + 01	2.241E + 01	7.304E + 01	8.262E + 01	8.441E + 01	6.713E + 01	3.250E + 01	1.496E + 02	6.125E + 01	4.843E + 02	6.775E + 01	1.691E
<i>C17F₁₂</i>	AVE	1.191E + 04	6.010E + 06	5.904E + 07	2.387E + 06	2.180E + 07	3.935E + 06	8.907E + 06	1.677E + 06	8.501E + 06	1.454E + 06	1.084E + 06	1.593E + 06	5.847E
	STD	7.313E + 03	1.675E + 07	1.203E + 08	3.481E + 07	1.820E + 08	5.180E + 07	4.668E + 07	1.270E + 07	2.038E + 07	3.308E + 07	5.745E + 07	8.347E + 07	5.626E
<i>C17F₁₃</i>	AVE	1.329E + 03	1.172E + 04	1.442E + 04	2.227E + 04	4.925E + 04	1.372E + 04	7.560E + 04	1.358E + 04	3.944E + 04	1.278E + 04	1.143E + 04	1.668E + 04	2.176E
	STD	4.064E + 01	1.125E + 04	8.923E + 04	1.289E + 04	5.098E + 04	9.566E + 04	4.719E + 04	8.062E + 04	4.437E + 04	1.310E + 04	2.276E + 04	3.811E + 04	9.522E
<i>C17F₁₄</i>	AVE	1.405E + 03	2.081E + 03	9.982E + 03	2.242E + 03	2.161E + 03	1.876E + 03	2.465E + 03	2.524E + 03	1.900E + 03	2.656E + 03	5.914E + 03	1.443E + 03	2.440E
	STD	4.423E + 00	1.088E + 03	9.193E + 03	1.192E + 03	9.474E + 03	4.971E + 03	9.810E + 03	1.412E + 03	3.327E + 03	1.871E + 03	1.510E + 03	5.987E + 03	1.478E
<i>C17F₁₅</i>	AVE	1.504E + 03	7.642E + 03	1.486E + 04	7.113E + 03	3.316E + 03	1.094E + 04	5.704E + 03	4.013E + 03	3.930E + 03	4.214E + 03	2.063E + 03	1.560E + 03	4.260E
	STD	2.454E + 00	5.584E + 03	6.651E + 03	6.346E + 03	1.406E + 03	3.987E + 03	2.721E + 03	2.390E + 03	1.295E + 03	2.042E + 03	5.202E + 03	2.160E + 03	3.591E
<i>C17F₁₆</i>	AVE	1.646E + 03	1.820E + 03	2.044E + 03	1.890E + 03	1.772E + 03	1.875E + 03	1.853E + 03	1.908E + 03	1.807E + 03	1.733E + 03	2.233E + 03	1.672E	1.728E
	STD	6.732E + 01	1.330E + 02	1.494E + 02	1.505E + 02	7.873E + 02	1.400E + 02	1.126E + 02	1.023E + 02	1.328E + 02	1.067E + 02	1.274E + 02	3.569E + 01	1.197E
<i>C17F₁₇</i>	AVE	1.711E + 03	1.775E + 03	1.868E + 03	1.785E + 03	1.782E + 03	1.784E + 03	1.794E + 03	1.797E + 03	1.795E + 03	1.750E + 03	1.883E + 03	1.747E	1.754E
	STD	9.543E + 00	4.280E + 01	1.058E + 02	4.391E + 01	1.184E + 02	3.097E + 02	1.367E + 02	6.029E + 02	2.490E + 02	2.753E + 02	1.012E + 03	7.103E + 02	2.578E
<i>C17F₁₈</i>	AVE	1.820E + 03	1.608E + 04	1.490E + 04	1.686E + 04	1.752E + 04	1.554E + 04	8.382E + 04	1.938E + 04	4.581E + 04	2.331E + 04	1.048E + 04	6.014E + 04	1.011E
	STD	9.750E + 00	1.055E + 04	8.210E + 04	1.191E + 04	2.245E + 04	1.177E + 04	7.614E + 04	1.171E + 04	1.160E + 04	1.678E + 04	5.279E + 04	2.573E	7.225E
<i>C17F₁₉</i>	AVE	1.902E + 03	1.162E + 04	4.127E + 04	5.676E + 04	6.454E + 04	2.402E + 04	1.691E + 04	6.588E + 04	6.251E + 04	1.317E + 04	1.634E + 04	1.931E + 04	5.890E
	STD	9.795E-01 + 04	1.246E + 04	2.700E + 04	1.223E + 04	5.584E + 04	4.161E + 04	1.288E + 04	3.508E + 04	5.471E + 04	1.304E + 04	1.173E + 04	1.270E	4.437E
<i>C17F₂₀</i>	AVE	2.004E + 03	2.093E + 03	2.153E + 03	2.173E + 03	2.114E + 03	2.168E + 03	2.142E + 03	2.155E + 03	2.110E + 03	2.049E + 03	2.305E + 03	2.053E + 03	2.062E
	STD	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03

(continued on next page)

Table 12 (continued)

Fun.	Index	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
C17F ₂₁	STD	5.681E	5.276E	6.776E	7.966E	3.590E	6.799E	2.725E	6.345E	4.151E	4.053E	9.331E	7.821E	3.320E
	+ 00	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 00	+ 01
	AVE	2.226E	2.228E	2.321E	2.320E	2.260E	2.307E	2.210E	2.316E	2.287E	2.295E	2.391E	2.225E	2.205E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	4.552E	1.616E	3.795E	5.578E	6.389E	5.942E	9.896E	5.934E	3.302E	5.567E	1.974E	8.811E	1.955E
	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 00	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01
C17F ₂₂	AVE	2.297E	2.314E	2.982E	2.318E	2.403E	2.359E	2.324E	2.313E	2.352E	2.312E	2.356E	2.281E	2.282E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	1.860E	2.800E	2.617E	1.441E	4.019E	9.157E	2.686E	1.473E	1.151E	1.561E	2.837E	1.880E	3.615E
	+ 01	+ 01	+ 02	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 02	+ 01	+ 02	+ 01	+ 01
	AVE	2.614E	2.661E	2.720E	2.645E	2.661E	2.659E	2.652E	2.670E	2.682E	2.625E	2.822E	2.616E	2.603E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
C17F ₂₃	STD	5.768E	2.003E	3.351E	2.271E	7.541E	2.540E	5.561E	2.453E	2.737E	8.863E	4.462E	5.976E	5.755E
	+ 00	+ 01	+ 01	+ 01	+ 00	+ 01	+ 01	+ 01	+ 01	+ 01	+ 00	+ 01	+ 01	+ 01
	AVE	2.669E	2.661E	2.819E	2.757E	2.786E	2.762E	2.558E	2.702E	2.740E	2.759E	2.625E	2.598E	2.575E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	1.127E	1.363E	8.548E	7.163E	9.899E	7.864E	1.306E	1.245E	3.871E	1.059E	1.758E	3.042E	1.163E
	+ 02	+ 02	+ 01	+ 01	+ 00	+ 01	+ 01	+ 01	+ 02	+ 01	+ 01	+ 02	+ 01	+ 02
C17F ₂₅	AVE	2.922E	2.960E	3.255E	2.966E	2.970E	2.999E	2.967E	2.903E	2.949E	2.938E	2.938E	2.924E	2.946E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	2.372E	5.223E	1.277E	4.952E	1.997E	5.391E	3.974E	6.957E	4.247E	2.629E	1.663E	1.910E	1.757E
	+ 01	+ 01	+ 02	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 00
	AVE	2.899E	3.260E	3.901E	3.420E	3.102E	3.494E	3.014E	2.903E	2.975E	3.007E	4.151E	2.985E	2.902E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
C17F ₂₆	STD	6.520E	1.695E	3.871E	4.419E	3.947E	5.251E	6.493E	1.016E	2.125E	7.136E	5.400E	6.551E	8.608E
	+ 01	+ 02	+ 02	+ 02	+ 01	+ 01	+ 02	+ 01	+ 02	+ 02	+ 01	+ 02	+ 01	+ 00
	AVE	3.094E	3.132E	3.255E	3.140E	3.104E	3.126E	3.126E	3.121E	3.123E	3.094E	3.330E	3.108E	3.104E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	2.388E	2.482E	5.806E	4.144E	2.790E	3.422E	7.910E	4.318E	2.762E	2.854E	5.572E	2.635E	1.817E
	+ 00	+ 01	+ 01	+ 01	+ 00	+ 01	+ 01	+ 00	+ 01	+ 01	+ 00	+ 01	+ 00	+ 01
C17F ₂₈	AVE	3.129E	3.329E	3.753E	3.450E	3.302E	3.426E	3.240E	3.293E	3.358E	3.302E	3.515E	3.196E	3.163E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	5.994E	1.425E	1.302E	1.364E	7.833E	1.404E	5.120E	1.317E	1.147E	9.426E	3.830E	4.241E	1.095E
	+ 01	+ 02	+ 02	+ 02	+ 01	+ 01	+ 02	+ 01	+ 02	+ 02	+ 01	+ 01	+ 01	+ 02
	AVE	3.167E	3.295E	3.366E	3.376E	3.243E	3.288E	3.295E	3.269E	3.249E	3.216E	3.539E	3.211E	3.194E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
C17F ₂₉	STD	1.918E	6.232E	1.060E	8.230E	3.512E	7.773E	4.687E	7.249E	4.155E	5.713E	1.470E	1.867E	2.834E
	+ 01	+ 01	+ 02	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 02	+ 01	+ 01
	AVE	8.500E	5.107E	1.889E	1.264E	1.533E	1.371E	1.964E	4.080E	3.290E	6.953E	2.133E	1.704E	3.903E
	+ 04	+ 06	+ 07	+ 06	+ 06	+ 06	+ 06	+ 06	+ 05	+ 06	+ 05	+ 06	+ 05	+ 05
	STD	2.695E	4.141E	1.479E	1.854E	9.682E	1.327E	1.604E	5.302E	1.784E	6.233E	7.066E	1.634E	8.877E
	+ 05	+ 06	+ 07	+ 06	+ 05	+ 06	+ 06	+ 05	+ 06	+ 05	+ 05	+ 05	+ 05	+ 05

dimensions, and in C17F₁₄, C17F₂₀, C17F₂₂ and C17F₃₀ for 50 dimensions. Overall, these results from the convergence curves of the CEC2017 test suite with 10, 30 and 50 dimensions reveal that EEFO shows superior exploration, exploitation, and balancing over its competitors.

When observing the mean of the best solutions in Tables 12-14, for 10 dimensions, the numbers of problems for which EEFO, BOA, WDO, CMA-ES, GSA, WDE and ASO achieve the best results are 19, 1, 1, 1, 1, 1 and 7 out of 29 test functions, respectively. Other algorithms such as LFD, AOA, WOA, SCA, HHO and MFO do not provide the best results for any tested problems. For 30 dimensions, the numbers of problems for which EEFO, WDO, CMA-ES, MFO and ASO achieve the best results are 15, 1, 2, 1 and 10 out of 29 test functions, respectively, and other algorithms such as LFD, AOA, WOA, SCA, HHO, BOA, GSA and WDE do not provide the best results for any problems. For 50 dimensions, the numbers of problems that EEFO, WDO, CMA-ES, MFO and ASO achieves the best results are 13, 1, 1, 2 and 12 out of 29 test functions, respectively, and other algorithms such as LFD, AOA, WOA, SCA, HHO, BOA, GSA and WDE do not provide the best results for any problems. It is evident from the results that EEFO is the superior optimizer compared to all other algorithms, with the highest number of best mean solutions obtained in dimensions 10, 30, and 50.

As shown in Tables 12-14, the standard deviation of the best solutions for 10 dimensions, the numbers of problems for which EEFO, WOA, SCA, CMA-ES, GSA, WDE and ASO achieve the best results are 11, 1, 1, 1, 2, 6 and 8 out of 29 test functions, respectively, and other algorithms such as LFD, AOA, HHO, BOA, WDO and MFO do not provide the best

results for any problems. These results show that EEFO has the highest degree of stability when testing the functions with 10 dimensions. For 30 dimensions, the numbers of problems for which EEFO, BOA, WDO, CMA-ES, MFO, GSA, WDE and ASO achieve the best results are 7, 1, 3, 1, 2, 1, 6 and 8 out of 29 test functions, respectively, and other algorithms such as LFD, AOA, WOA, SCA and HHO do not provide the best results for any problems. ASO has the highest degree of stability and EEFO is the second most stable optimizer when testing functions with 30 dimensions. For 50 dimensions, the numbers of problems for which EEFO, SCA, BOA, WDO, CMA-ES, MFO, GSA, WDE and ASO achieve the best results are 7, 1, 1, 1, 2, 1, 3, 6 and 7 out of 29 test functions, respectively, and other algorithms such as LFD, AOA, WOA and HHO do not provide the best results for any problems. EEFO and ASO have the same highest degree of stability when testing the functions with 50 dimensions. In addition, again observing Tables 12-14, the numbers of problems EEFO simultaneously offers the best mean and standard deviation are 10 for 10 dimensions, 7 for 30 dimensions and 6 for 50 dimensions, respectively, verifying EEFO's superior optimization performance on the CEC2017 suite in various dimensions.

Last, the numbers of problems that ASO simultaneously offers the best mean and standard deviation are 5 for 10 dimensions, 7 for 30 dimensions and 5 for 50 dimensions, making ASO the second-best optimizer in tackling the CEC2017 test suite. MFO and CMA-ES perform much better than the others in 2 test functions in all the dimensions considered. GSA and WDE provide superior solutions over other optimizers in several test functions in some of the dimensions considered. However, for all considered dimensions, LFD, AOA, WOA, and SCA

Table 13

Results of EEFO and other algorithms for CEC2017 test suite with 30 dimensions.

Fun.	Index	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
C17F ₁	AVE	4.896E + 06	4.237E + 09	5.272E + 10	2.332E + 09	1.801E + 10	1.925E + 10	2.798E + 10	2.769E + 08	1.484E + 10	7.534E + 09	1.567E + 10	1.256E + 10	2.327E + 03
	STD	5.161E + 06	1.716E + 09	1.054E + 10	6.961E + 08	3.572E + 09	5.723E + 09	6.092E + 09	1.132E + 08	2.123E + 10	6.008E + 09	2.726E + 09	1.842E + 09	2.728E + 03
C17F ₃	AVE	5.106E + 04	8.861E + 04	7.993E + 04	2.604E + 05	6.916E + 04	7.807E + 04	6.768E + 04	1.984E + 04	1.190E + 05	1.452E + 05	9.563E + 04	1.229E + 05	6.991E + 04
	STD	7.808E + 03	1.569E + 04	8.260E + 03	5.993E + 04	1.517E + 04	7.074E + 03	7.972E + 03	5.369E + 03	1.844E + 04	3.740E + 04	8.075E + 03	9.759E + 03	2.045E + 04
C17F ₄	AVE	5.243E + 02	1.812E + 03	1.461E + 04	9.627E + 02	2.539E + 03	3.424E + 03	6.543E + 03	5.803E + 02	2.977E + 03	9.156E + 02	3.693E + 03	2.085E + 03	5.368E + 02
	STD	2.361E + 01	1.055E + 03	4.537E + 03	1.801E + 02	6.370E + 02	1.250E + 03	1.868E + 03	3.556E + 02	3.506E + 03	4.039E + 02	1.054E + 03	3.328E + 02	3.567E + 01
C17F ₅	AVE	5.925E + 02	7.367E + 02	8.892E + 02	8.319E + 02	8.162E + 02	8.356E + 02	8.846E + 02	7.037E + 02	7.087E + 02	6.957E + 02	7.401E + 02	7.695E + 02	5.504E + 01
	STD	1.831E + 01	4.344E + 01	3.088E + 01	5.780E + 01	2.061E + 01	3.523E + 01	2.375E + 01	3.082E + 01	1.204E + 01	4.726E + 01	2.758E + 01	2.203E + 01	1.108E + 01
C17F ₆	AVE	6.081E + 02	6.588E + 02	6.801E + 02	6.782E + 02	6.611E + 02	6.734E + 02	6.734E + 02	6.620E + 02	6.458E + 02	6.379E + 02	6.628E + 02	6.442E + 02	6.007E + 02
	STD	2.875E + 00	7.762E + 00	7.477E + 00	1.136E + 01	7.049E + 00	9.281E + 00	6.009E + 00	8.500E + 00	3.412E + 01	1.251E + 01	3.742E + 01	4.544E + 01	1.174E + 00
C17F ₇	AVE	8.684E + 02	1.273E + 03	1.406E + 03	1.275E + 03	1.223E + 03	1.341E + 03	1.317E + 03	1.028E + 02	8.987E + 02	1.085E + 02	1.087E + 02	1.299E + 02	7.626E + 02
	STD	2.951E + 01	6.341E + 01	4.509E + 01	1.119E + 01	5.622E + 01	7.253E + 01	4.569E + 01	5.221E + 01	3.002E + 01	1.799E + 01	5.948E + 01	4.396E + 01	7.468E + 01
C17F ₈	AVE	9.031E + 02	9.846E + 02	1.111E + 03	1.054E + 03	1.088E + 03	1.056E + 03	1.121E + 03	9.596E + 03	9.735E + 03	1.002E + 03	9.612E + 03	1.062E + 03	8.468E + 02
	STD	2.488E + 01	3.908E + 01	2.965E + 01	5.875E + 01	2.840E + 01	2.936E + 01	1.754E + 01	3.363E + 01	1.015E + 01	5.005E + 01	1.708E + 01	1.500E + 01	1.302E + 01
C17F ₉	AVE	1.734E + 03	6.850E + 03	7.547E + 03	1.066E + 04	7.821E + 03	8.968E + 03	9.220E + 03	6.783E + 03	2.899E + 03	6.617E + 03	4.742E + 03	6.999E + 03	9.053E + 02
	STD	4.592E + 02	2.318E + 03	1.075E + 03	3.111E + 03	1.685E + 03	1.673E + 03	1.470E + 03	1.768E + 03	4.114E + 03	2.226E + 03	4.242E + 03	8.621E + 03	1.676E + 01
C17F ₁₀	AVE	4.960E + 03	6.134E + 03	7.723E + 03	7.084E + 03	8.787E + 03	7.569E + 03	8.782E + 03	6.369E + 03	8.728E + 03	5.279E + 03	5.300E + 03	7.211E + 03	4.677E + 03
	STD	9.826E + 02	7.303E + 02	4.814E + 02	7.231E + 02	3.376E + 02	6.234E + 02	3.548E + 02	7.968E + 02	3.178E + 02	7.070E + 02	5.664E + 02	2.523E 8.160E	
C17F ₁₁	AVE	1.235E + 03	6.137E + 03	9.882E + 03	7.390E + 03	3.369E + 03	6.229E + 03	6.092E + 03	1.398E + 03	4.098E + 03	5.846E + 03	7.726E + 03	2.841E + 03	1.261E + 03
	STD	3.182E + 01	2.080E + 03	3.871E + 03	3.304E + 03	7.182E + 03	1.459E + 03	1.528E + 03	5.088E + 03	1.637E + 03	5.721E + 03	1.118E + 03	4.013E + 03	7.297E + 01
C17F ₁₂	AVE	1.694E + 06	4.290E + 08	1.310E + 10	3.350E + 09	2.438E + 09	1.611E + 09	3.417E + 09	5.365E + 07	1.074E + 07	1.665E + 09	2.919E + 08	8.460E + 08	2.551E + 06
	STD	1.327E + 06	6.313E + 08	3.699E + 10	2.409E + 09	8.721E + 09	1.273E + 09	1.468E + 09	2.443E + 09	2.104E + 09	3.610E + 09	9.596E + 08	2.045E + 08	2.694E + 06
C17F ₁₃	AVE	9.674E + 03	1.272E + 08	1.046E + 10	2.708E + 06	9.423E + 06	2.609E + 06	9.311E + 06	3.026E + 06	2.639E + 06	3.825E + 07	4.246E + 07	1.655E + 07	4.409E + 04
	STD	8.908E + 03	3.552E + 08	4.393E + 09	2.768E + 06	4.675E + 06	5.761E + 06	5.322E + 06	1.534E + 06	8.020E + 06	1.933E + 07	4.137E + 07	6.539E + 07	2.234E + 04
C17F ₁₄	AVE	3.487E + 04	1.110E + 06	3.070E + 06	2.129E + 06	6.356E + 05	1.443E + 06	9.276E + 05	4.477E + 05	2.683E + 05	2.156E + 05	1.447E + 05	1.586E + 05	1.134E + 05
	STD	4.398E + 04	1.041E + 06	3.951E + 06	2.352E + 06	5.851E + 05	1.195E + 06	6.522E + 05	4.073E + 05	4.328E + 05	1.694E + 05	4.960E + 05	8.502E + 05	8.092E + 05
C17F ₁₅	AVE	4.463E + 03	9.823E + 03	7.518E + 07	3.237E + 07	4.322E + 07	1.419E + 07	1.730E + 07	1.367E + 05	1.514E + 06	4.879E + 04	1.642E + 04	7.382E + 04	2.055E + 04
	STD	4.597E + 03	5.148E + 07	1.502E + 08	1.166E + 08	3.530E + 07	2.488E + 07	1.060E + 07	6.008E + 05	9.661E + 05	4.905E + 04	3.861E + 03	2.705E + 03	9.666E + 03
C17F ₁₆	AVE	2.378E + 03	3.566E + 03	5.160E + 03	4.061E + 03	4.174E + 07	3.973E + 07	5.008E + 07	3.323E + 03	4.226E + 03	3.017E + 03	4.704E + 03	3.502E + 03	2.612E + 03
	STD	2.805E + 02	3.673E + 02	1.255E + 03	8.256E + 02	2.932E + 02	4.845E + 02	5.007E + 02	3.918E + 02	1.098E + 03	3.050E + 02	5.348E + 03	1.745E + 02	2.437E + 02
C17F ₁₇	AVE	2.018E + 03	2.638E + 03	3.701E + 03	2.775E + 03	2.752E + 02	2.694E + 02	3.138E + 02	2.344E + 02	2.531E + 02	2.412E + 02	2.999E + 02	2.459E + 02	2.117E + 03
	STD	1.752E + 02	3.313E + 03	1.174E + 02	2.742E + 03	1.939E + 02	2.973E + 03	3.320E + 02	2.158E + 02	5.911E + 02	3.167E + 02	2.408E + 02	1.211E 1.864E	
C17F ₁₈	AVE	3.051E + 05	1.018E + 07	2.106E + 07	1.298E + 07	1.228E + 07	1.667E + 07	8.175E + 07	5.672E + 06	3.235E + 06	6.510E + 06	4.635E + 06	2.640E + 06	8.176E + 05
	STD	2.808E + 05	1.224E + 07	1.851E + 07	1.251E + 07	7.603E + 07	1.795E + 07	4.179E + 07	5.117E + 06	6.625E + 06	8.716E + 06	3.103E + 06	1.032E + 06	7.902E + 05
C17F ₁₉	AVE	9.381E + 03	2.442E + 06	1.274E + 08	1.539E + 07	8.588E + 07	2.770E + 07	2.107E + 07	1.757E + 07	1.509E + 07	1.525E + 06	1.479E + 06	1.272E + 06	1.063E + 04
	STD	1.054E + 04	3.340E + 06	2.489E + 08	1.614E + 07	7.576E + 07	4.997E + 07	9.789E + 07	1.464E + 06	1.177E + 06	4.671E + 06	1.233E + 06	6.698E + 06	8.284E + 03
C17F ₂₀	AVE	2.357E + 03	2.710E + 03	2.792E + 03	2.843E + 03	2.878E + 03	2.820E + 03	2.913E + 03	2.787E + 03	2.881E + 03	2.567E + 03	3.149E + 03	2.685E + 03	2.523E + 03

(continued on next page)

Table 13 (continued)

Fun.	Index	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
C17F ₂₁	STD	1.433E + 02	1.586E + 02	2.301E + 02	2.657E + 02	1.503E + 02	2.041E + 02	8.139E + 01	2.462E + 02	1.685E + 02	2.261E + 02	2.041E + 02	1.026E + 02	2.164E + 02
	AVE	2.382E + 03	2.563E + 03	2.667E + 03	2.608E + 03	2.591E + 03	2.623E + 03	2.390E + 03	2.508E + 03	2.609E + 03	2.492E + 03	2.677E + 03	2.560E + 03	2.359E + 03
	STD	3.756E + 01	6.664E + 01	4.445E + 01	6.731E + 01	2.137E + 01	4.506E + 01	8.301E + 01	3.938E + 01	1.150E + 01	4.880E + 02	3.721E + 01	2.493E + 01	1.851E + 01
	AVE	2.317E + 03	6.400E + 03	8.841E + 03	7.467E + 03	9.325E + 03	8.010E + 03	3.770E + 03	4.775E + 03	3.544E + 03	6.324E + 03	7.443E + 03	4.577E + 03	3.001E + 03
	STD	2.760E + 00	1.731E + 03	9.521E + 03	2.009E + 03	2.052E + 03	1.553E + 03	4.294E + 02	2.811E + 03	1.858E + 03	1.508E + 03	5.331E + 02	3.522E + 02	1.613E + 03
	AVE	2.772E + 03	3.151E + 03	3.499E + 03	3.129E + 03	3.059E + 03	3.217E + 03	3.177E + 03	3.044E + 03	3.269E + 03	2.829E + 03	3.941E + 03	2.986E + 03	2.778E + 03
C17F ₂₃	STD	3.350E + 01	1.106E + 02	1.818E + 02	9.011E + 01	4.431E + 01	1.107E + 02	9.884E + 01	9.911E + 01	8.383E + 01	4.175E + 01	1.758E + 01	3.105E + 01	5.869E + 01
	AVE	2.940E + 03	3.407E + 03	3.823E + 03	3.250E + 03	3.232E + 03	3.278E + 03	3.392E + 03	3.169E + 03	3.413E + 03	2.983E + 03	3.973E + 03	3.182E + 03	2.924E + 03
	STD	2.859E + 01	1.214E + 02	2.193E + 02	8.150E + 01	4.425E + 01	1.246E + 02	1.310E + 02	9.489E + 01	9.268E + 01	3.348E + 01	1.356E + 02	2.247E + 01	7.754E + 01
	AVE	2.934E + 03	3.122E + 03	5.636E + 03	3.137E + 03	3.536E + 03	3.386E + 03	3.766E + 03	3.019E + 03	2.887E + 03	3.299E + 03	3.308E + 03	3.583E + 03	2.921E + 03
	STD	2.808E + 01	9.008E + 01	7.967E + 02	7.002E + 02	2.081E + 02	1.754E + 02	2.541E + 02	3.959E + 01	1.197E + 01	3.416E + 02	1.223E + 02	1.412E + 02	1.868E + 01
	AVE	4.537E + 03	8.240E + 03	1.038E + 04	8.154E + 03	7.686E + 03	9.066E + 03	6.588E + 03	6.036E + 03	9.173E + 03	5.689E + 03	9.018E + 03	6.211E + 03	3.959E + 03
C17F ₂₆	STD	9.453E + 02	1.368E + 03	8.496E + 03	1.035E + 02	5.600E + 02	9.260E + 02	1.155E + 03	2.000E + 03	1.239E + 03	4.863E + 02	5.059E + 02	6.526E + 02	8.947E + 02
	AVE	3.267E + 03	3.275E + 03	4.423E + 03	3.463E + 03	3.518E + 03	3.486E + 03	3.776E + 03	3.461E + 03	3.844E + 03	3.239E + 03	5.334E + 03	3.443E + 03	3.390E + 03
	STD	2.607E + 01	1.802E + 02	2.829E + 02	1.216E + 02	7.340E + 02	1.375E + 02	1.109E + 03	1.635E + 03	1.891E + 03	1.849E + 02	4.246E + 02	3.269E + 02	1.195E + 02
	AVE	3.281E + 03	3.964E + 03	6.800E + 03	3.592E + 03	4.319E + 03	4.740E + 03	4.854E + 03	3.337E + 03	4.313E + 03	4.002E + 03	4.611E + 03	4.073E + 03	3.288E + 03
	STD	2.808E + 01	3.902E + 02	1.018E + 02	1.152E + 02	2.840E + 02	4.459E + 02	5.246E + 02	2.669E + 01	1.552E + 03	9.571E + 02	2.533E + 02	1.710E + 02	5.733E + 02
	AVE	3.809E + 03	5.038E + 03	6.874E + 03	5.535E + 03	5.096E + 03	5.247E + 03	5.928E + 03	4.617E + 03	3.780E + 03	4.077E + 03	6.834E + 03	4.495E + 03	3.884E + 03
C17F ₂₉	STD	2.058E + 02	4.892E + 02	1.383E + 03	4.487E + 02	2.619E + 02	5.024E + 02	4.321E + 02	3.553E + 02	5.134E + 02	3.025E + 02	4.865E + 02	1.230E + 02	1.997E + 02
	AVE	3.016E + 04	3.233E + 07	1.624E + 09	4.157E + 07	1.839E + 08	8.207E + 07	1.155E + 08	1.017E + 07	8.257E + 07	2.469E + 06	2.900E + 07	1.529E + 07	2.987E + 05
	STD	3.007E + 04	4.216E + 07	9.310E + 09	3.122E + 07	7.651E + 08	7.686E + 07	4.555E + 07	4.820E + 06	1.669E + 08	8.568E + 06	3.193E + 07	6.033E + 06	3.241E + 05
	AVE	3.016E + 04	3.233E + 07	1.624E + 09	4.157E + 07	1.839E + 08	8.207E + 07	1.155E + 08	1.017E + 07	8.257E + 07	2.469E + 06	2.900E + 07	1.529E + 07	2.987E + 05
	STD	3.007E + 04	4.216E + 07	9.310E + 09	3.122E + 07	7.651E + 08	7.686E + 07	4.555E + 07	4.820E + 06	1.669E + 08	8.568E + 06	3.193E + 07	6.033E + 06	3.241E + 05
	AVE	3.016E + 04	3.233E + 07	1.624E + 09	4.157E + 07	1.839E + 08	8.207E + 07	1.155E + 08	1.017E + 07	8.257E + 07	2.469E + 06	2.900E + 07	1.529E + 07	2.987E + 05

exhibit poor performance in all test functions. Obviously, compared to other algorithms, LFD, AOA, WOA, and SCA lack sufficient optimization capacity in terms of exploration, exploitation and avoiding local optima when solving the CEC2017 suite. Experimental studies demonstrate that EEFO competes strongly with ASO, BOA, MFO, GSA, and WDE algorithms in many test functions with dimensions of 10, 30, and 50. Moreover, EEFO outperforms most other algorithms in the majority of test functions across all evaluated dimensions. This achievement serves as additional proof that EEFO has the capability to outperform relatively new and well-designed metaheuristics, including ASO, BOA, MFO, LFD, GSA, and WDE, in solving broadly well-known test functions.

The ranking results of EEFO and other considered algorithms based on the mean of the best solutions from Tables 12–14 using the Friedman test are summarized in Table 15. From Table 15, for 10 dimensions, the overall rank of the algorithms is EEFO > ASO > WDE > MFO > WDO > CMA-ES > LFD > SCA > BOA > HHO > WOA > GSA > AOA; for 30 dimensions, the overall rank of the algorithms is EEFO > ASO > WDO > MFO > WDE > LFD > CMA-ES > WOA > GSA > SCA > HHO > BOA > AOA; for 50 dimensions, the overall rank of the algorithms is EEFO > ASO > WDO > MFO > LFD > WOA > WDE > CMA-ES > GSA > HHO > SCA > BOA > AOA. It is evident that EEFO ranks first for all the considered dimensions, demonstrating its stable performance. ASO also shows relatively good and stable performance and ranks second for different dimensions. AOA performs the worst for different dimensions, ranking last. Other algorithms such as WDO, MFO, LFD, WOA, WDE, CMA-ES, GSA, HHO, SCA and BOA rank differently as the dimensions change, indicating their unstable performance.

Tables 16–18 show the statistical results of different algorithms using the Nemenyi test for CEC2017 with 10, 30 and 50 dimensions. According to these tables, for 10 and 30 dimensions, the number of EEFO that has a significant difference with other algorithms is 10 out of 12 algorithms, ranking first. For the case of 50 dimensions, both EEFO and ASO obtain the same highest number of significant differences with other algorithms, which is 9 out of 12 algorithms. These results demonstrate that EEFO is a highly robust and efficient algorithm compared to its competitors, capable of effectively solving problems across diverse dimensions and complexities.

The exploration and exploitation behaviors can be visualized by the exploration-exploitation ratio (Hussain Salleh Cheng Shi 2018). The exploration-exploitation ratio of EEFO versus iterations for the CEC2017 functions with 30 dimensions is depicted in Fig. 12. Observing Fig. 12, in the initial iterative steps, EEFO performs more exploration and less exploitation. EEFO gradually performs more exploitation and less exploration with increasing iteration. It can be seen that EEFO exhibits a good balance between exploration and exploitation in solving various complex problems.

4.6. CEC2011 test function analysis

To further examine the effectiveness of the EEFO algorithm, the CEC2011 test suite, which consists of 22 challenging benchmark functions, is utilized. These problems feature numerous local optima and diverse shapes that vary across different dimensions and regions, creating an additional challenge for the optimization method to solve

Table 14

Results of EEFO and other algorithms for CEC2017 test suite with 50 dimensions.

(continued on next page)

Table 14 (continued)

Fun.	Index	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
C17F ₂₁	STD	2.847E	3.152E	2.182E	3.560E	1.835E	3.148E	1.771E	3.430E	2.220E	4.196E	3.737E	1.430E	3.541E
	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02
	AVE	2.509E	2.882E	3.097E	3.034E	2.938E	3.005E	2.895E	2.744E	2.996E	2.732E	2.984E	2.893E	2.422E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	4.529E	9.643E	7.149E	9.509E	4.345E	8.247E	1.395E	6.464E	9.262E	6.973E	4.973E	2.585E	2.516E
	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01	+ 02	+ 01	+ 01	+ 01	+ 01	+ 01	+ 01
C17F ₂₂	AVE	8.465E	1.209E	1.606E	1.395E	1.696E	1.475E	1.517E	1.307E	1.645E	1.036E	1.247E	1.436E	8.491E
	+ 03	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 03
	STD	3.168E	1.579E	5.843E	1.359E	3.671E	9.498E	2.682E	1.340E	2.692E	9.485E	6.432E	1.166E	2.346E
	+ 03	+ 03	+ 02	+ 03	+ 02	+ 02	+ 02	+ 03	+ 03	+ 03	+ 02	+ 02	+ 03	+ 03
C17F ₂₃	AVE	3.021E	3.907E	4.527E	3.751E	3.668E	3.809E	4.014E	3.719E	3.915E	3.152E	5.054E	3.536E	3.030E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	6.440E	1.938E	2.636E	1.836E	5.882E	1.352E	1.358E	1.712E	1.372E	5.347E	2.305E	4.696E	1.050E
C17F ₂₄	+ 01	+ 02	+ 02	+ 01	+ 01	+ 02	+ 02	+ 02	+ 02	+ 02	+ 01	+ 02	+ 01	+ 02
	AVE	3.238E	4.042E	4.948E	3.891E	3.835E	3.936E	4.386E	3.672E	4.035E	3.212E	5.377E	3.778E	3.259E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
C17F ₂₅	STD	5.928E	1.820E	2.590E	1.658E	5.670E	1.937E	2.162E	1.324E	1.448E	4.933E	1.612E	5.527E	1.841E
	+ 01	+ 02	+ 02	+ 02	+ 01	+ 02	+ 02	+ 02	+ 02	+ 02	+ 01	+ 02	+ 01	+ 02
	AVE	3.254E	4.943E	1.602E	4.498E	8.382E	8.440E	9.717E	3.656E	3.018E	4.799E	8.683E	9.211E	3.211E
C17F ₂₆	+ 03	+ 03	+ 04	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	4.904E	6.328E	1.922E	3.195E	1.131E	1.202E	1.013E	1.614E	2.093E	1.285E	6.441E	8.435E	8.706E
	+ 01	+ 02	+ 03	+ 02	+ 03	+ 03	+ 03	+ 03	+ 02	+ 01	+ 03	+ 02	+ 02	+ 01
C17F ₂₇	AVE	7.036E	1.275E	1.716E	1.421E	1.347E	1.479E	1.353E	1.035E	1.453E	8.120E	1.384E	1.227E	5.993E
	+ 03	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 04	+ 03	+ 04	+ 04	+ 03
	STD	2.238E	8.005E	1.059E	1.678E	8.105E	7.942E	1.626E	3.152E	1.238E	8.126E	5.889E	5.621E	7.556E
C17F ₂₈	+ 03	+ 02	+ 03	+ 03	+ 02	+ 02	+ 02	+ 03	+ 03	+ 03	+ 02	+ 02	+ 02	+ 02
	AVE	3.722E	5.335E	6.902E	4.732E	4.854E	4.501E	5.718E	4.506E	5.538E	3.586E	8.870E	4.486E	4.092E
	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
C17F ₂₉	STD	1.032E	6.038E	5.798E	6.141E	2.112E	4.387E	2.891E	6.445E	3.109E	1.398E	7.454E	9.839E	4.385E
	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 01	+ 02
	AVE	3.673E	6.294E	1.242E	5.512E	8.581E	7.807E	8.804E	4.184E	9.464E	7.959E	8.983E	7.945E	3.590E
C17F ₃₀	+ 03	+ 03	+ 04	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03	+ 03
	STD	9.423E	6.858E	1.376E	5.577E	8.269E	8.271E	9.477E	1.986E	3.278E	1.354E	5.712E	5.074E	9.765E
	+ 01	+ 02	+ 03	+ 02	+ 02	+ 02	+ 02	+ 02	+ 02	+ 03	+ 03	+ 02	+ 02	+ 01
C17F ₃₀	AVE	4.528E	9.176E	9.376E	8.905E	8.568E	1.002E	1.113E	6.459E	8.692E	5.158E	3.240E	6.985E	4.800E
	+ 03	+ 03	+ 04	+ 03	+ 03	+ 04	+ 04	+ 04	+ 03	+ 03	+ 03	+ 04	+ 03	+ 03
	STD	3.014E	2.871E	2.659E	1.595E	1.102E	2.291E	2.791E	5.709E	3.872E	5.059E	9.743E	3.488E	4.447E
C17F ₃₀	+ 02	+ 03	+ 05	+ 03	+ 03	+ 03	+ 03	+ 03	+ 02	+ 03	+ 02	+ 03	+ 02	+ 02
	AVE	4.412E	4.179E	7.930E	3.093E	1.127E	7.252E	1.208E	1.939E	5.611E	1.192E	1.339E	4.698E	5.234E
	+ 06	+ 08	+ 09	+ 08	+ 09	+ 08	+ 08	+ 09	+ 08	+ 08	+ 08	+ 09	+ 08	+ 07
C17F ₃₀	STD	1.166E	7.184E	3.326E	1.692E	4.239E	6.385E	4.267E	3.621E	1.073E	2.970E	5.591E	1.217E	1.503E
	+ 06	+ 08	+ 09	+ 08	+ 08	+ 08	+ 08	+ 08	+ 07	+ 09	+ 08	+ 08	+ 08	+ 07

real-world issues. Therefore, these complex characteristics render these problems more appropriate for assessing the potential of metaheuristic algorithms in terms of their exploration, exploitation, and local optima avoidance capabilities.

The results of different algorithms in terms of the mean and standard deviation of the best solutions for the CEC2011 test suite are provided in Table 19. Inspecting the table, out of 22 test functions, the numbers of problems for which EEFO, WOA, HHO, CMA-ES, MFO, WDE and ASO achieve the best results are 13, 1, 1, 1, 1, 2 and 5, respectively, while other algorithms such as LFD, AOA, SCA, BOA, WDO and GSA do not provide any best results for any problems. It is evident that EEFO is the most efficient algorithm for the majority of test problems.

Table 20 presents the results using the Friedman test to rank different algorithms on the CEC2011 test suite. The results show that EEFO has the best ranking, achieving an average rank of 2.23, indicating that it is the best-performing algorithm among all the algorithms considered. MFO and ASO follow closely behind, with average ranks of 3.89 and 4.25, respectively. In addition, SCA performs poorly, finishing in last place with an average rank of 10.36. The final ranking of the average rank provided by the Friedman test for all algorithms on the CEC2011 test suite is EEFO > MFO > ASO > WDE > HHO > WOA > WDO > GSA > CMA-ES > BOA > AOA > LFD > SCA. This result provides further insight into the superior performance of the proposed algorithm over its competitors.

The statistical results of different algorithms using the Nemenyi test for the CEC2011 test suite are provided in Table 21. From Table 21, it is evident that there is a significant difference between EEFO and the other

8 algorithms. This is followed by MFO, which has a significant difference with 7 other algorithms, while ASO and SCA have significant differences with 6 other algorithms each. On the other hand, WOA and WDO exhibit significant differences with only 2 algorithms. These observations demonstrate that our algorithm is more effective than others and has significant differences in performance when compared to other algorithms. The convergence curves of different algorithms for the CEC2011 test suite are depicted in Fig. 13. Inspecting Fig. 13, EEFO can achieve a high level of convergence rate and accuracy and converge steadily toward the optimal solutions for most functions, demonstrating EEFO's superior convergence ability when solving the CEC2011 test suite.

4.7. Application of EEFO to engineering design problems (EDPs)

The performance of EEFO in solving real-world optimization applications in 10 engineering design cases is verified and described in this section. Since these engineering cases have many constraints, a common penalty method is employed. When the decision variables of the considered problems are out of the acceptable ranges, a penalty coefficient is added to the objective function. These EDPs are constraint optimization problems, thus a constraint-handling method needs to be employed. For simplicity, a common penalty method is chosen for tackling constraints in these EDPs. The penalty method is described as (Wang et al., 2022c; Coello & Montes, 2002):

$$\text{Minimize } F(\vec{x}) = f(\vec{x}) \pm \left(\sum_{i=1}^p a_i G_i(\vec{x}) + \sum_{j=1}^q b_j H_j(\vec{x}) \right) \quad (35)$$

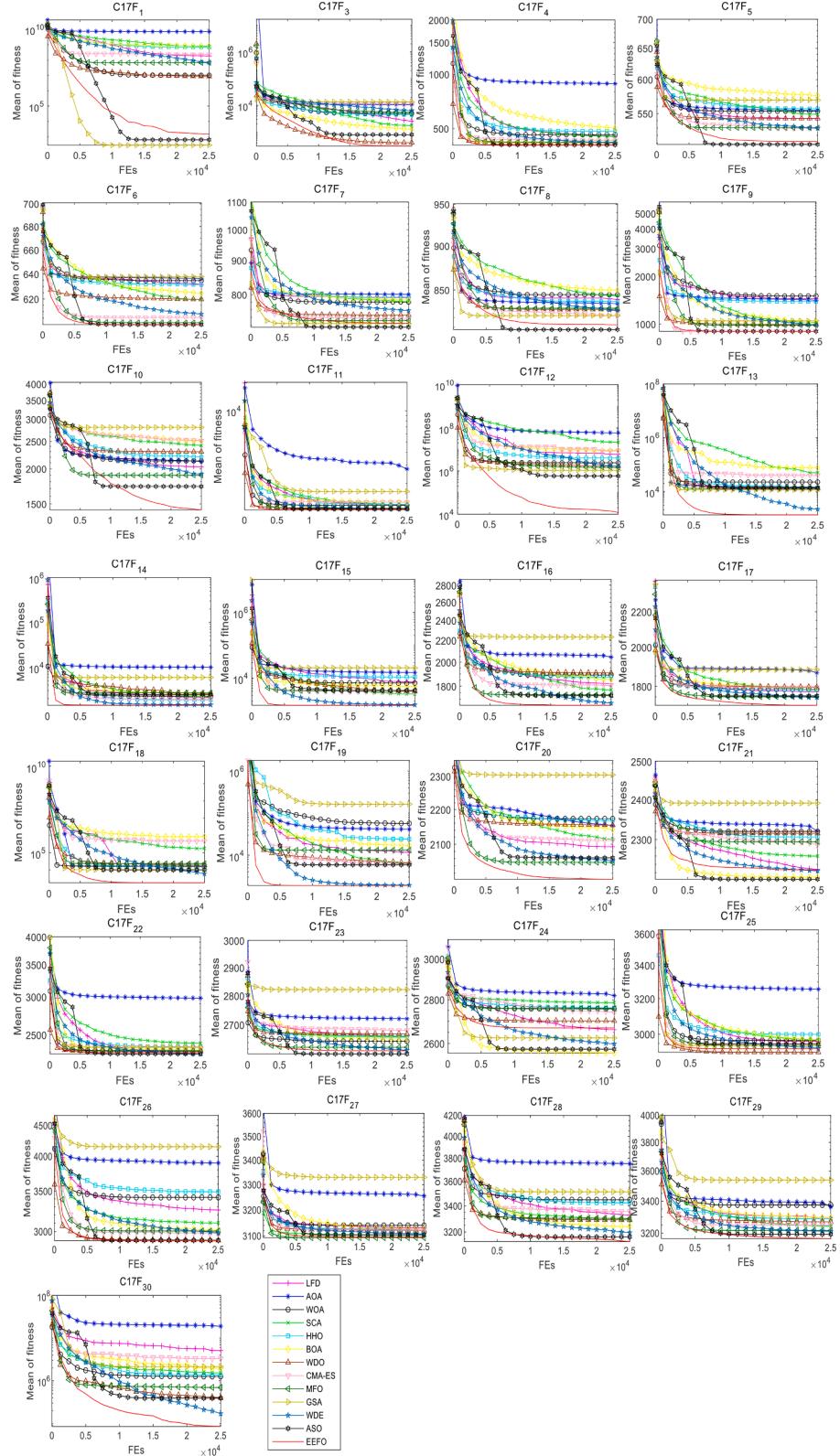


Fig. 9. Comparisons of convergence curve variations of all algorithms for CEC2017 test suite with 10 dimensions.

$$G_i(\vec{x}) = \max(0, g_i(\vec{x}))^\eta \quad (36)$$

$$H_j(\vec{x}) = |h_j(\vec{x})|^\lambda \quad (37)$$

Where $g_i(\vec{x})$ represents the inequality constraint, $h_i(\vec{x})$ represents

the equality constraint, p and q represent the numbers of inequality constraints and equality constraints, respectively, a_i and b_j represent the positive constants, respectively, and, η and λ are 1 or 2. For this penalty method, when the component value of a candidate solution exceeds the upper or lower limit, it will be replaced by a randomly generated value

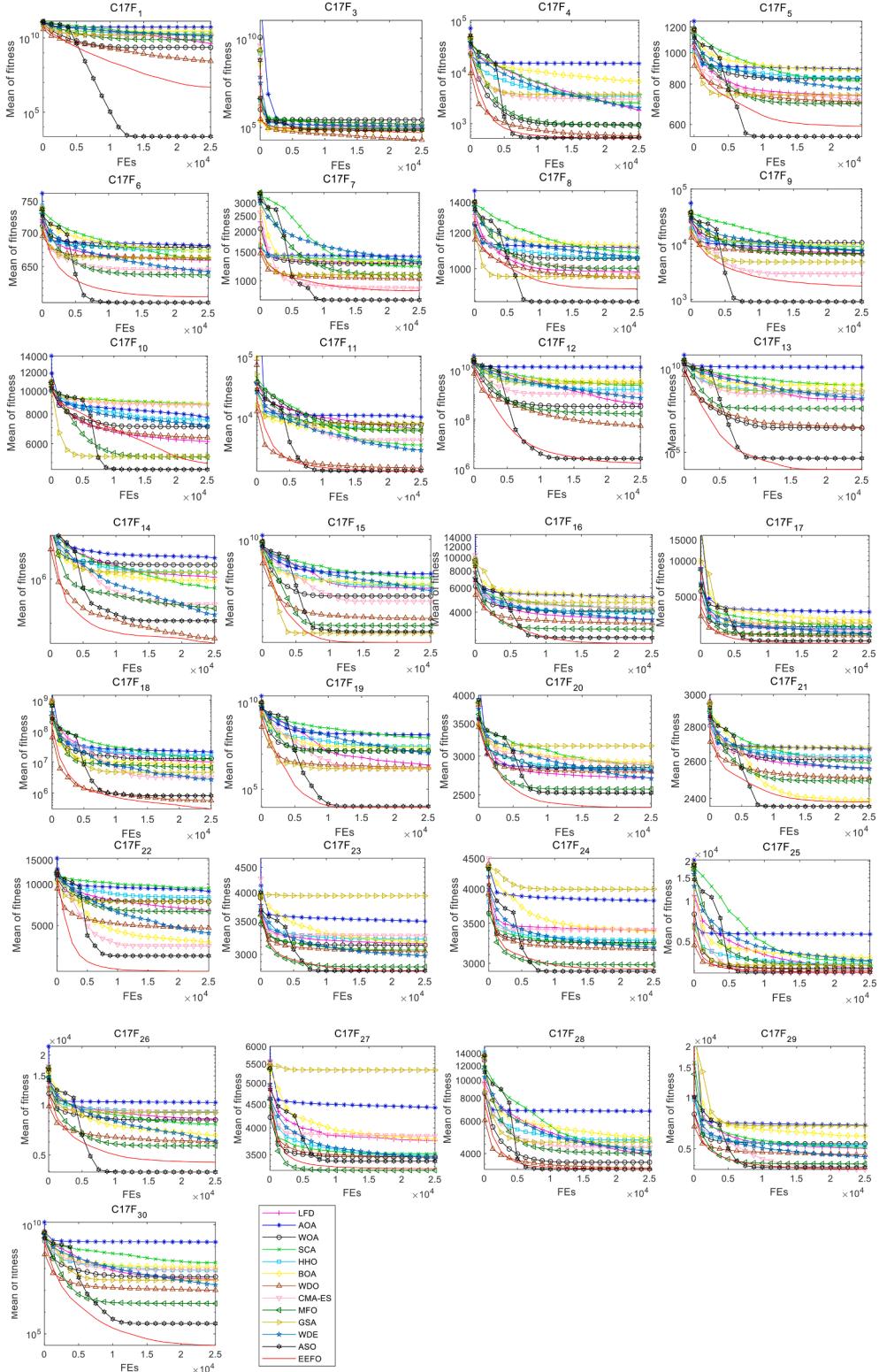


Fig. 10. Comparisons of convergence curve variations of all algorithms for CEC2017 test suite with 30 dimensions.

within the search space. When a candidate solution violates any constraint, the value of the objective function will significantly increase, which will prompt the algorithm to switch from the infeasible region to the feasible region.

4.7.1. Robot gripper design

The design is intended to minimize the difference between the

minimum and maximum force generated by the robot gripper (Osyczka & Krenich, 2004). This problem consists of seven design variables subject to six constraints related to the robot. The structure of this design is depicted in Fig. 14. This design is formulated below.

$$\begin{aligned} \text{Consider variables : } \vec{x} &= [x_1, x_2, x_3, x_4] \\ &= [a, b, c, e, f, l, \delta] \end{aligned}$$

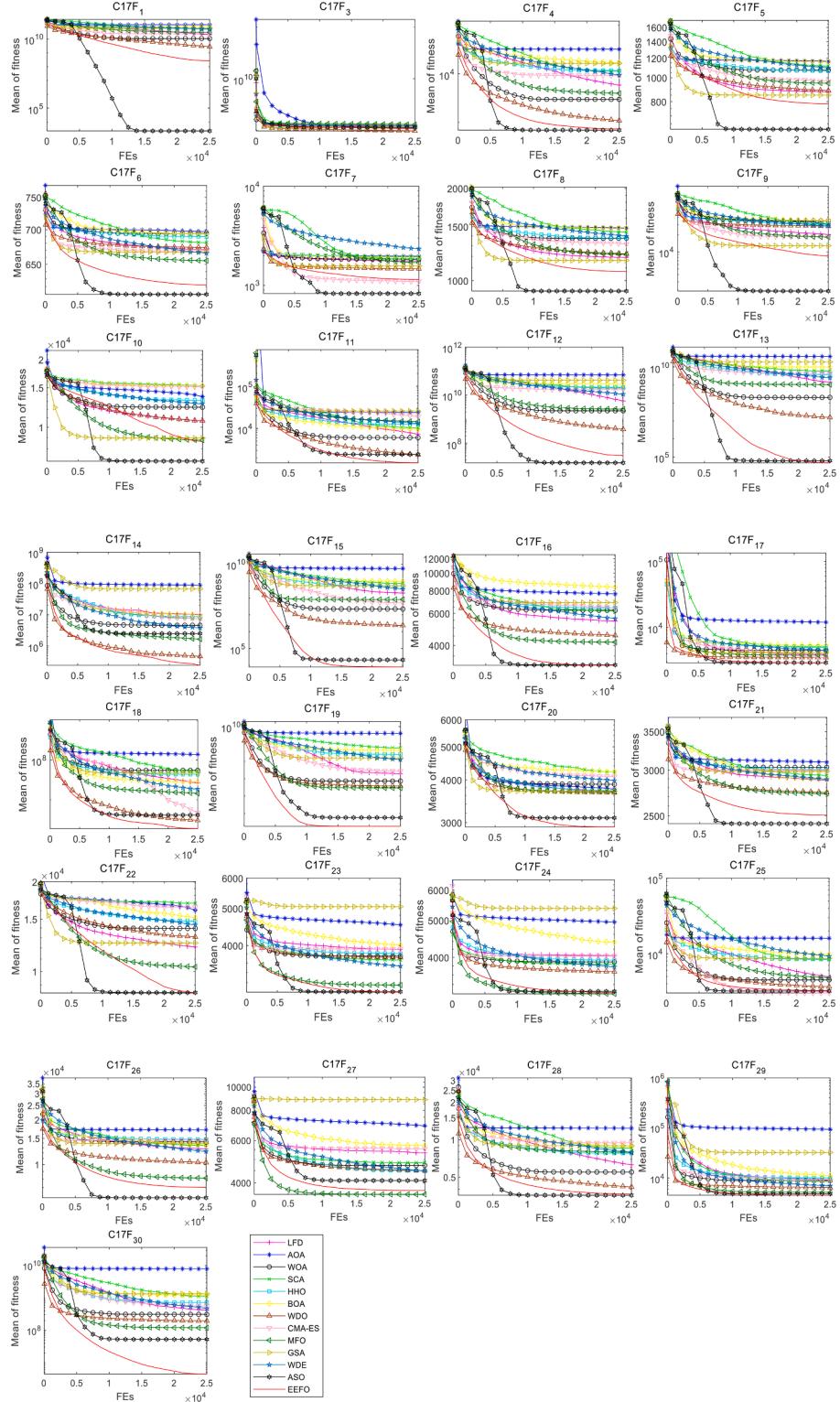


Fig. 11. Comparisons of convergence curve variations of all algorithms for CEC2017 test suite with 50 dimensions.

$$\text{Minimize : } f_1(\vec{x}) = - \min_z F_k(\vec{x}, z) + \max_z F_k(\vec{x}, z),$$

Subject to:

$$g_1(\vec{x}) = -Y_{\min} + y(\vec{x}, Z_{\max}) \leq 0,$$

$$g_2(\vec{x}) = -y(\vec{x}, Z_{\max}) \leq 0,$$

$$g_3(T) = Y_{\max} - y(T, 0) \leq 0,$$

$$g_4(T) = y(T, 0) - Y_G \leq 0,$$

$$g_5(\vec{x}) = l^2 + e^2 - (a + b)^2 \leq 0,$$

$$g_6(\vec{x}) = b^2 - (a - e)^2 - (l - Z_{\max})^2 \leq 0,$$

Table 15

Results of all algorithms using Friedman test for CEC2017 test suite with 10, 30 and 50 dimensions.

Dimensions		EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
10	Sum of rank	51	225	330	263	232	262	253	188	214.5	157	277	106	80.5
	Mean of rank	1.76	7.76	11.38	9.07	8	9.03	8.72	6.48	7.4	5.41	9.55	3.66	2.78
	Overall rank	1	7	13	11	8	10	9	5	6	4	12	3	2
30	Sum of rank	44	204	352	241	262	279	288	121	209	139	260	184	56
	Mean of rank	1.52	7.03	12.14	8.31	9.03	9.62	9.93	4.17	7.21	4.79	8.97	6.34	1.93
	Overall rank	1	6	13	8	10	11	12	3	7	4	9	5	2
50	Sum of rank	47	190	348	211	264	260	313	118	223	135	259	215	56
	Mean of rank	1.62	6.55	12	7.28	9.1	8.97	10.79	4.07	7.69	4.66	8.93	7.41	1.93
	Overall rank	1	5	13	6	11	10	12	3	8	4	9	7	2

Table 16

Statistical results of different algorithms using Nemenyi test for CEC2017 test suite with 10 dimensions.

Algorithms	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
EEFO	NaN	1	1	1	1	1	1	1	1	1	1	0	0
LFD	1	NaN	1	0	0	0	0	0	0	0	0	1	1
AOA	1	1	NaN	0	1	0	0	1	1	1	0	1	1
WOA	1	0	0	NaN	0	0	0	0	0	1	0	1	1
SCA	1	0	1	0	NaN	0	0	0	0	0	0	1	1
HHO	1	0	0	0	0	NaN	0	0	0	1	0	1	1
BOA	1	0	0	0	0	0	NaN	0	0	1	0	1	1
WDO	1	0	1	0	0	0	0	NaN	0	0	0	0	1
CMA-ES	1	0	1	0	0	0	0	0	NaN	0	0	1	1
MFO	1	0	1	1	0	1	1	0	0	NaN	1	0	0
GSA	1	0	0	0	0	0	0	0	0	1	NaN	1	1
WDE	0	1	1	1	1	1	1	0	1	0	1	NaN	0
ASO	0	1	1	1	1	1	1	1	1	0	1	0	NaN
Sum of significant difference	10	4	8	4	4	4	4	3	4	6	4	8	9

Table 17

Statistical results of different algorithms using Nemenyi test for CEC2017 test suite with 30 dimensions.

Algorithms	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
EEFO	NaN	1	1	1	1	1	1	0	1	1	1	1	0
LFD	1	NaN	1	0	0	0	0	0	0	0	0	0	1
AOA	1	1	NaN	1	0	0	0	1	1	1	0	1	1
WOA	1	0	1	NaN	0	0	0	1	0	1	0	0	1
SCA	1	0	0	0	NaN	0	0	1	0	1	0	0	1
HHO	1	0	0	0	0	NaN	0	1	0	1	0	1	1
BOA	1	0	0	0	0	0	NaN	1	0	1	0	1	1
WDO	0	0	1	1	1	1	1	NaN	0	0	1	0	0
CMA-ES	1	0	1	0	0	0	0	0	NaN	0	0	0	1
MFO	1	0	1	1	1	1	1	0	0	NaN	1	0	0
GSA	1	0	0	0	0	0	0	1	0	1	NaN	0	1
WDE	1	0	1	0	0	1	1	0	0	0	0	NaN	1
ASO	0	1	1	1	1	1	1	0	1	0	1	1	NaN
Sum of significant difference	10	3	8	5	4	5	5	6	3	7	4	5	9

Table 18

Statistical results of different algorithms using Non-parametric Nemenyi test for CEC2017 test suite with 50 dimensions.

Algorithms	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
EEFO	NaN	1	1	1	1	1	1	0	1	0	1	1	0
LFD	1	NaN	1	0	0	0	1	0	0	0	0	0	1
AOA	1	1	NaN	1	0	0	0	1	1	1	0	1	1
WOA	1	0	1	NaN	0	0	1	1	0	0	0	0	1
SCA	1	0	0	0	NaN	0	0	1	0	1	0	0	1
HHO	1	0	0	0	0	NaN	0	1	0	1	0	0	1
BOA	1	1	0	1	0	0	NaN	1	0	1	0	1	1
WDO	0	0	1	1	1	1	1	NaN	1	0	1	1	0
CMA-ES	1	0	1	0	0	0	0	1	NaN	0	0	0	1
MFO	0	0	1	0	1	1	1	0	0	NaN	1	0	0
GSA	1	0	0	0	0	0	0	1	0	1	NaN	0	1
WDE	1	0	1	0	0	0	1	1	0	0	0	NaN	1
ASO	0	1	1	1	1	1	1	0	1	0	1	1	NaN
Sum of significant difference	9	4	8	5	4	4	7	8	4	5	4	5	9

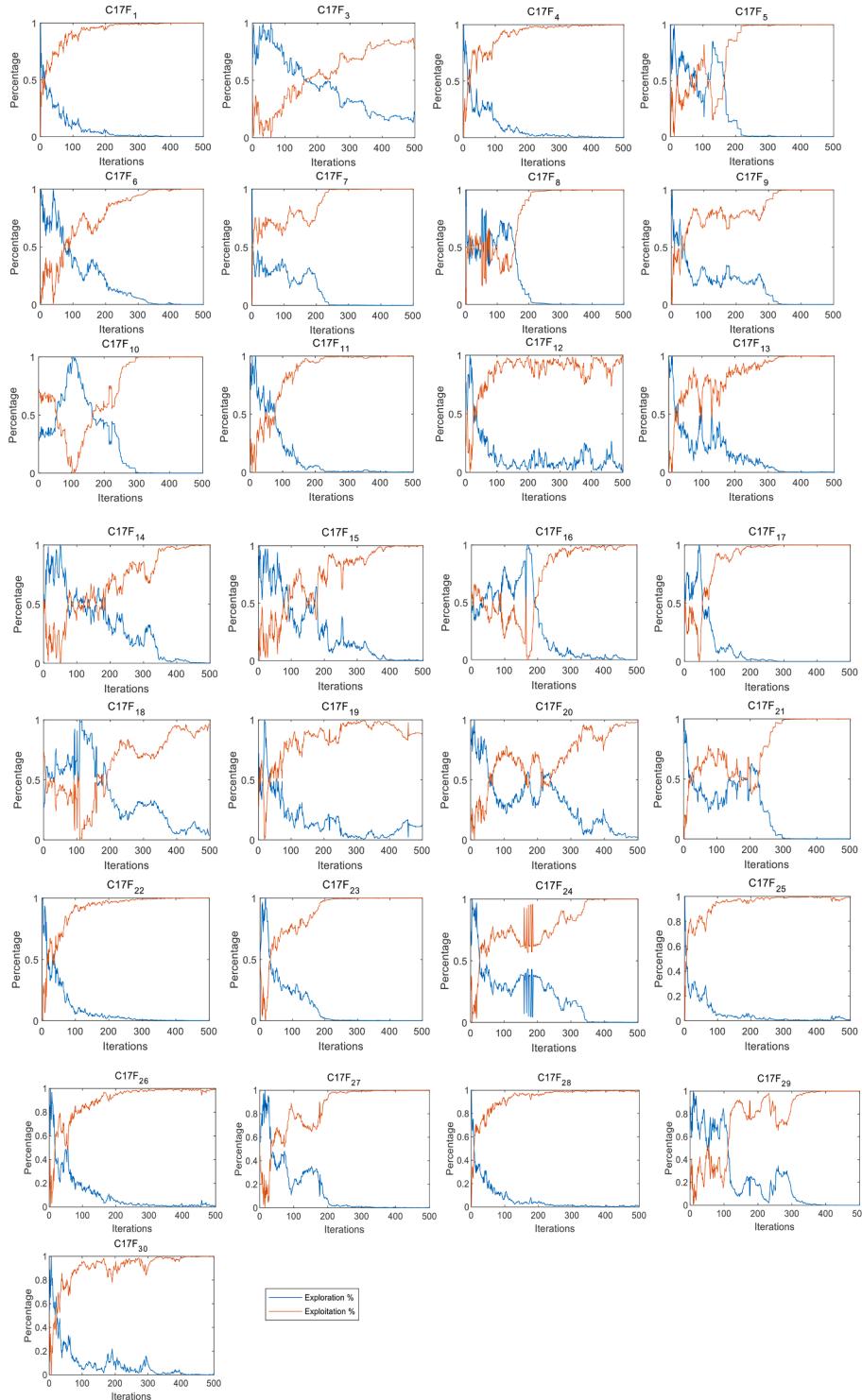


Fig. 12. Exploration and exploitation ratio for EEFO on the CEC2017 test suite with 30 dimensions.

$g_7(\vec{x}) = Z_{\max} - l \leq 0,$
where,

$$F_k = \frac{P \sin(\alpha + \beta)}{2c \cos(\alpha)},$$

$$\alpha = \cos^{-1}\left(\frac{a^2 + g^2 - b^2}{2ag}\right) + \varphi, g = \sqrt{e^2 + (z - l)^2},$$

$$\beta = \cos^{-1}\left(\frac{b^2 + g^2 - a^2}{2bg}\right) - \varphi, \varphi = \tan^{-1}\left(\frac{e}{l - z}\right),$$

$$y(\vec{x}, z) = 2(f + e + c \sin(\beta + \delta)),$$

$$Y_{\min} = 50, Y_{\max} = 100, Y_G = 150, Z_{\max} = 100, P = 100.$$

Variable range:

$$0 \leq e \leq 50, 100 \leq c \leq 200, 10 \leq f, a, b \leq 150, 1 \leq \delta \leq 3.14, 100 \leq l \leq 300.$$

Table 19

Results of different algorithms for CEC2011 test suite.

Fun.	Index	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
<i>C11F₁</i>	AVE	2.360E + 00	1.891E + 01	1.930E + 01	1.612E + 01	1.417E + 01	1.369E + 01	1.576E + 01	2.372E + 01	1.971E + 01	9.188E + 00	2.532E + 01	9.221E + 00	1.963E + 01
	STD	4.616E + 00	4.024E + 00	6.777E + 00	6.261E + 00	2.139E + 00	5.394E + 00	1.950E + 00	2.333E + 00	3.521E + 00	8.593E + 00	7.501E-01	4.016E + 00	3.442E + 00
<i>C11F₂</i>	AVE	-2.389E + 01	-9.170E + 00	-6.023E + 00	-2.104E + 01	-5.914E + 00	-2.352E + 01	-1.142E + 01	-2.395E + 01	-7.487E + 00	-1.933E + 01	-1.578E + 01	-1.189E + 01	-2.585E + 01
	STD	1.809E + 00	2.963E + 00	6.950E-01	3.722E + 00	8.901E-01	2.641E + 00	9.782E-01	2.305E + 00	1.951E + 00	4.127E + 00	3.717E + 00	7.119E-01	1.999E + 00
<i>C11F₃</i>	AVE	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05	1.151E-05
	STD	6.069E-19	6.069E-16	3.162E-15	5.946E-16	3.219E-15	1.283E-19	8.128E-16	6.826E-16	6.938E-16	3.661E-17	3.381E-19	2.813E-17	2.331E-19
<i>C11F₄</i>	AVE	1.384E + 01	1.975E + 01	1.462E + 01	1.383E + 01	1.409E + 01	1.400E + 01	1.657E + 01	1.965E + 01	1.831E + 01	1.603E + 01	1.416E + 01	1.405E + 01	1.410E + 01
	STD	5.400E-02	1.762E + 00	5.152E-01	1.766E-01	1.910E-01	2.284E-01	2.650E + 00	2.805E + 00	3.423E + 00	3.007E + 00	2.721E-01	1.996E-01	2.649E-01
<i>C11F₅</i>	AVE	-3.373E + 01	-2.001E + 01	-1.779E + 01	-3.042E + 01	-2.056E + 01	-2.749E + 01	-2.705E + 01	-2.643E + 01	-2.096E + 01	-3.368E + 01	-2.056E + 01	-3.185E + 01	-2.149E + 01
	STD	1.963E + 00	3.772E + 00	1.550E + 00	3.634E + 00	8.277E-01	3.409E + 00	1.816E + 00	3.525E + 00	1.562E + 00	1.806E + 00	1.720E + 00	4.623E-01	2.200E + 00
<i>C11F₆</i>	AVE	-1.936E + 01	1.647E + 01	2.131E + 01	1.028E + 01	2.896E + 01	2.940E + 00	3.810E + 01	-5.309E + 00	-1.256E + 01	-1.210E + 01	2.383E + 00	-7.769E + 00	-7.048E + 00
	STD	3.535E + 00	1.741E + 01	6.620E + 00	9.507E + 00	7.470E + 00	1.012E + 01	7.642E + 00	7.868E + 00	4.658E + 00	4.780E + 00	4.610E + 00	1.991E + 00	2.481E + 00
<i>C11F₇</i>	AVE	9.628E-01	1.511E + 00	1.767E + 00	1.613E + 00	1.719E + 00	1.476E + 00	1.701E + 00	1.348E + 00	1.528E + 00	1.070E + 00	1.035E + 00	1.359E + 00	7.784E-01
	STD	1.601E-01	1.903E-01	1.047E-01	1.743E-01	6.972E-02	1.969E-01	1.161E-01	2.425E-01	1.796E-01	2.481E-01	2.177E-01	6.675E-02	2.245E-01
<i>C11F₈</i>	AVE	2.200E + 02	2.314E + 02	2.229E + 02	2.218E + 02	2.311E + 02	2.236E + 02	2.311E + 02	5.736E + 02	2.271E + 02	2.200E + 02	2.370E + 02	2.200E + 02	2.231E + 02
	STD	0	1.095E + 01	6.332E + 00	5.692E + 00	1.481E + 01	7.589E + 00	1.405E + 01	2.992E + 02	1.376E + 01	0	1.118E + 01	0	6.871E + 00
<i>C11F₉</i>	AVE	1.732E + 04	1.723E + 06	3.888E + 05	5.891E + 05	9.882E + 05	4.959E + 05	6.017E + 05	1.589E + 06	1.832E + 06	4.937E + 05	2.520E + 05	1.245E + 06	5.804E + 05
	STD	1.324E + 04	2.015E + 05	4.502E + 04	8.310E + 04	1.422E + 05	7.180E + 04	5.593E + 04	3.413E + 05	1.598E + 05	7.771E + 04	3.563E + 04	5.040E + 04	6.673E + 04
<i>C11F₁₀</i>	AVE	-2.118E + 01	-1.042E + 01	-1.062E + 01	-1.118E + 01	-1.182E + 01	-1.148E + 01	-1.355E + 01	-1.429E + 01	-2.099E + 01	-2.059E + 01	-1.137E + 01	-1.641E + 01	-1.532E + 01
	STD	3.245E-01	4.020E-01	4.439E-01	1.350E + 00	1.530E + 00	4.514E-01	6.692E-01	3.365E + 00	1.351E-01	5.258E-01	5.224E-01	8.430E-01	1.236E + 00
<i>C11F₁₁</i>	AVE	5.211E + 04	5.884E + 06	1.577E + 08	1.146E + 06	9.772E + 07	1.240E + 06	4.241E + 06	1.198E + 06	5.304E + 04	1.315E + 05	1.008E + 06	1.856E + 07	5.244E + 04
	STD	6.008E + 02	8.437E + 05	1.166E + 07	1.133E + 05	7.567E + 06	1.454E + 05	2.689E + 06	1.270E + 05	7.552E + 02	1.210E + 05	5.193E + 04	2.122E + 06	5.583E + 02
<i>C11F₁₂</i>	AVE	6.715E + 06	1.130E + 07	1.114E + 07	5.500E + 06	9.943E + 06	5.458E + 06	7.251E + 06	7.460E + 06	1.470E + 06	1.690E + 06	6.484E + 06	9.731E + 06	6.269E + 06
	STD	1.868E + 05	5.838E + 05	2.428E + 05	4.958E + 05	1.786E + 05	2.673E + 05	4.095E + 05	2.474E + 05	4.780E + 04	2.787E + 05	6.288E + 04	3.493E + 05	2.681E + 04
<i>C11F₁₃</i>	AVE	1.545E + 04	1.549E + 04	1.554E + 04	1.549E + 04	1.574E + 04	1.546E + 04	1.613E + 04	1.860E + 04	1.550E + 04	1.546E + 04	1.229E + 05	1.546E + 04	1.549E + 04
	STD	2.062E + 00	1.678E + 01	5.065E + 01	2.556E + 01	2.544E + 02	1.139E + 01	5.049E + 02	9.776E + 03	5.239E + 01	1.325E + 01	5.788E + 04	2.298E + 00	2.076E + 01
<i>C11F₁₄</i>	AVE	1.916E + 04	1.935E + 04	1.954E + 04	1.924E + 04	1.937E + 04	1.965E + 04	1.981E + 04	1.921E + 04	2.010E + 04	1.931E + 04	1.914E + 04	1.903E + 04	1.927E + 04
	STD	8.548E + 01	2.317E + 02	3.266E + 02	1.831E + 02	2.580E + 02	4.525E + 02	3.670E + 02	1.507E + 02	7.765E + 02	1.755E + 02	1.142E + 02	9.912E + 01	2.320E + 02
<i>C11F₁₅</i>	AVE	3.299E + 04	1.364E + 05	1.074E + 05	1.101E + 05	1.694E + 05	1.059E + 05	1.702E + 05	1.150E + 05	1.434E + 05	3.313E + 04	3.368E + 05	3.301E + 04	3.315E + 04
	STD	4.134E + 01	4.741E + 04	6.138E + 04	8.201E + 04	4.099E + 04	5.588E + 04	4.451E + 04	7.802E + 04	6.459E + 04	6.989E + 01	3.807E + 04	3.803E + 01	6.054E + 01
<i>C11F₁₆</i>	AVE	1.370E + 05	1.479E + 05	4.522E + 05	1.418E + 05	8.502E + 06	1.496E + 05	7.921E + 06	1.446E + 05	2.518E + 06	1.439E + 05	1.118E + 06	1.375E + 05	1.419E + 05
	STD	2.386E + 03	5.403E + 03	4.307E + 05	5.106E + 03	6.276E + 06	6.697E + 03	6.049E + 06	3.850E + 03	3.180E + 06	4.260E + 03	3.077E + 06	2.105E + 03	1.600E + 03
<i>C11F₁₇</i>	AVE	2.057E + 06	6.439E + 09	5.939E + 09	2.420E + 09	7.562E + 09	2.399E + 09	8.227E + 09	2.237E + 08	3.831E + 09	2.925E + 06	9.002E + 09	3.281E + 08	2.082E + 06
	STD	1.710E + 05	9.684E + 08	7.435E + 08	9.959E + 08	1.491E + 09	8.126E + 08	6.179E + 08	6.859E + 08	1.099E + 09	7.037E + 05	8.980E + 08	2.126E + 08	2.476E + 05
<i>C11F₁₈</i>	AVE	1.000E + 06	5.685E + 07	5.962E + 07	2.060E + 07	8.425E + 07	1.663E + 07	2.460E + 07	2.239E + 07	6.817E + 07	1.230E + 06	1.189E + 06	2.120E + 07	9.428E + 05
	STD	1.015E + 05	2.400E + 07	8.043E + 06	1.082E + 07	6.966E + 06	5.292E + 06	3.005E + 06	1.784E + 07	1.483E + 07	3.624E + 05	2.168E + 05	4.228E + 06	9.722E + 02
<i>C11F₁₉</i>	AVE	1.520E + 06	5.708E + 07	6.194E + 07	2.196E + 07	8.199E + 07	1.608E + 07	2.604E + 07	3.485E + 07	6.844E + 07	2.181E + 06	1.577E + 06	2.161E + 07	1.247E + 06
	STD	1.948E + 05	1.764E + 07	5.003E + 06	1.399E + 07	8.900E + 06	6.106E + 06	5.183E + 06	2.437E + 07	1.262E + 07	3.686E + 05	3.504E + 05	3.209E + 06	9.472E + 04
<i>C11F₂₀</i>	AVE	1.003E + 06	5.656E + 07	5.759E + 07	2.290E + 07	8.564E + 07	1.442E + 07	2.687E + 07	3.398E + 07	7.182E + 07	1.190E + 06	1.101E + 06	2.081E + 07	9.455E + 05
	STD	1.377E + 05	2.068E + 07	8.324E + 06	1.056E + 07	6.631E + 06	8.935E + 06	4.080E + 06	1.411E + 07	7.920E + 06	2.492E + 05	1.603E + 05	3.489E + 06	4.193E + 03
<i>C11F₂₁</i>	AVE	1.417E + 01	3.544E + 01	3.917E + 01	3.115E + 01	3.851E + 01	2.702E + 01	3.086E + 01	2.852E + 01	3.201E + 01	1.826E + 01	4.612E + 01	2.262E + 01	2.101E + 01
	STD	3.129E + 00	5.394E + 00	6.748E + 00	4.112E + 00	3.119E + 00	2.722E + 00	3.556E + 00	7.121E + 00	5.370E + 00	1.791E + 00	7.887E + 00	2.763E + 00	6.958E + 00
<i>C11F₂₂</i>	AVE	1.738E + 01	3.657E + 01	3.607E + 01	3.191E + 01	3.743E + 01	3.173E + 01	3.644E + 01	3.120E + 01	3.963E + 01	2.241E + 01	5.394E + 01	2.091E + 01	2.570E + 01
	STD	4.961E + 00	3.694E + 00	2.905E + 00	4.099E + 00	1.762E + 00	4.467E + 00	4.123E + 00	6.806E + 00	4.082E + 00	3.290E + 00	9.496E + 00	1.778E + 00	1.844E + 00

Table 20

Ranks of different algorithms using Friedman test for CEC2011 test suite.

	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
Sum of rank	49	214.5	212.5	142	228	127.5	202	166.5	200	85.5	173.5	107.5	93.5
Mean of rank	2.23	9.75	9.66	6.45	10.36	5.80	9.18	7.57	9.09	3.89	7.89	4.89	4.25
Overall rank	1	12	11	6	13	5	10	7	9	2	8	4	3

Table 21

Statistical results of different algorithms using Nemenyi test for CEC 2011.

Algorithms	EEFO	LFD	AOA	WOA	SCA	HHO	BOA	WDO	CMA-ES	MFO	GSA	WDE	ASO
EEFO	NaN	1	1	1	1	0	1	1	1	0	1	0	0
LFD	1	NaN	0	0	0	1	0	0	0	1	0	1	1
AOA	1	0	NaN	0	0	1	0	0	0	1	0	1	1
WOA	1	0	0	NaN	1	0	0	0	0	0	0	0	0
SCA	1	0	0	1	NaN	1	0	0	0	1	0	1	1
HHO	0	1	1	0	1	NaN	0	0	0	0	0	0	0
BOA	1	0	0	0	0	0	NaN	0	0	1	0	1	1
WDO	1	0	0	0	0	0	0	NaN	0	1	0	0	0
CMA-ES	1	0	0	0	0	0	0	0	NaN	1	0	1	1
MFO	0	1	1	0	1	0	1	1	1	NaN	1	0	0
GSA	1	0	0	0	0	0	0	0	0	1	NaN	0	1
WDE	0	1	1	0	1	0	1	0	1	0	0	NaN	0
ASO	0	1	1	0	1	0	1	0	1	0	1	0	NaN
Sum of significant difference	8	5	5	2	6	3	4	2	4	7	3	5	6

All compared algorithms, including EEFO, LFD, AOA, WOA, SCA, HHO, BOA, WDO, CMA-ES, MFO, GSA, WDE, and ASO, are run 30 times independently and the results are presented in Table 22. The 'Best' and 'AVE' metrics in Table 22 show that EEFO is the most effective method for approximating the optimal solution. Table 23 lists the minimum costs and their corresponding design variables for EEFO and other competing algorithms. This result shows that EEFO is more effective in tackling the robot gripping problem. Fig. 15 illustrates the constraint values and convergence curve variations of EEFO versus the iteration number.

4.7.2. Step-cone pulley design

This problem is the minimization of the weight of the step-cone pulley by optimizing the five design variables (Kumar et al., 2020); it consists of eleven constraints, of which three are equality constraints and eight are inequality constraints. The structure of the step-cone pulley is depicted in Fig. 16. This design is formulated below.

Consider : $\vec{x} = [x_1, x_2, x_3, x_4, x_5] = [d_1, d_2, d_3, d_4, \omega]$,

Minimize:

$$f_1(\vec{x}) = \rho x_5 \left[x_1^2 \left\{ 1 + \left(\frac{N_1}{N} \right)^2 \right\} + x_2^2 \left\{ 1 + \left(\frac{N_2}{N} \right)^2 \right\} + x_3^2 \left\{ 1 + \left(\frac{N_3}{N} \right)^2 \right\} + x_4^2 \left\{ 1 + \left(\frac{N_4}{N} \right)^2 \right\} \right],$$

Subject to:

$$h_1(\vec{x}) = C_1 - C_2 = 0,$$

$$h_2(\vec{x}) = C_1 - C_3 = 0,$$

$$h_3(\vec{x}) = C_1 - C_4 = 0,$$

$$g_i(\vec{x}) = R_i \geq 2, \quad i = 1, 2, 3, 4,$$

Variable range: $50 \leq x_i \leq 400$, $i = 1, 2, 3, 4$, $50 \leq x_5 \leq 400$.

where, C_i indicates the length of the belt to obtain speed N_i :

$$C_i = \frac{\pi x_i}{2} \left(1 + \frac{N_i}{N} \right) + \frac{\left(\frac{N_i}{N} - 1 \right)^2}{4a} + 2a, \quad i = 1, 2, 3, 4.$$

R_i is the tension ratio:

$$R_i = \exp \left[\mu \left\{ \pi - 2 \sin^{-1} \left\{ \left(\frac{N_i}{N} - 1 \right) \frac{x_i}{2a} \right\} \right\} \right], \quad i = 1, 2, 3, 4.$$

P_i is the power transmitted at each step:

$$P_i = stx_5 \left[1 - \exp \left[-\mu \left\{ \pi - 2 \sin^{-1} \left\{ \left(\frac{N_i}{N} - 1 \right) \frac{x_i}{2a} \right\} \right\} \right] \right] \frac{\pi x_i N_i}{60}, \quad i = 1, 2, 3, 4.$$

$$\rho = 7200 \text{ kg/m}^3, \quad a = 3 \text{ m}, \quad \mu = 0.35, \quad s = 1.75 \text{ MPa}, \quad t = 8 \text{ mm}.$$

EEFO is compared with other competing algorithms over 30 independent runs. The results from various optimizers are shown in Table 24. EEFO offers better performance than all other competitors for the 'Best', 'Worst', 'AVE', and 'STD' metrics, highlighting the superior optimization ability of this optimizer. The best solutions of all the algorithms and their corresponding design variables for different algorithms are provided in Table 25. Fig. 17 illustrates the constraint values and convergence curve variations of EEFO versus iteration number.

4.7.3. Four-stage gearbox design

The objective of this challenging problem is to minimize the weight of the gearbox (Hu et al., 2022b). This design consists of 22 discrete structure variables subject to 86 constraints. The schematic of the design is presented in Fig. 18. There are numerous local solutions in the feasible search space with a ratio less than 0.0001. This design is formulated below.

Consider variable

$$\vec{x} = \{N_{p1}, N_{g1}, N_{p2}, N_{g2}, \dots, b_1, b_2, \dots, x_{p1}, x_{g1}, x_{g2}, \dots, y_{p1}, y_{g1}, \dots, y_{g4}\},$$

Minimize:

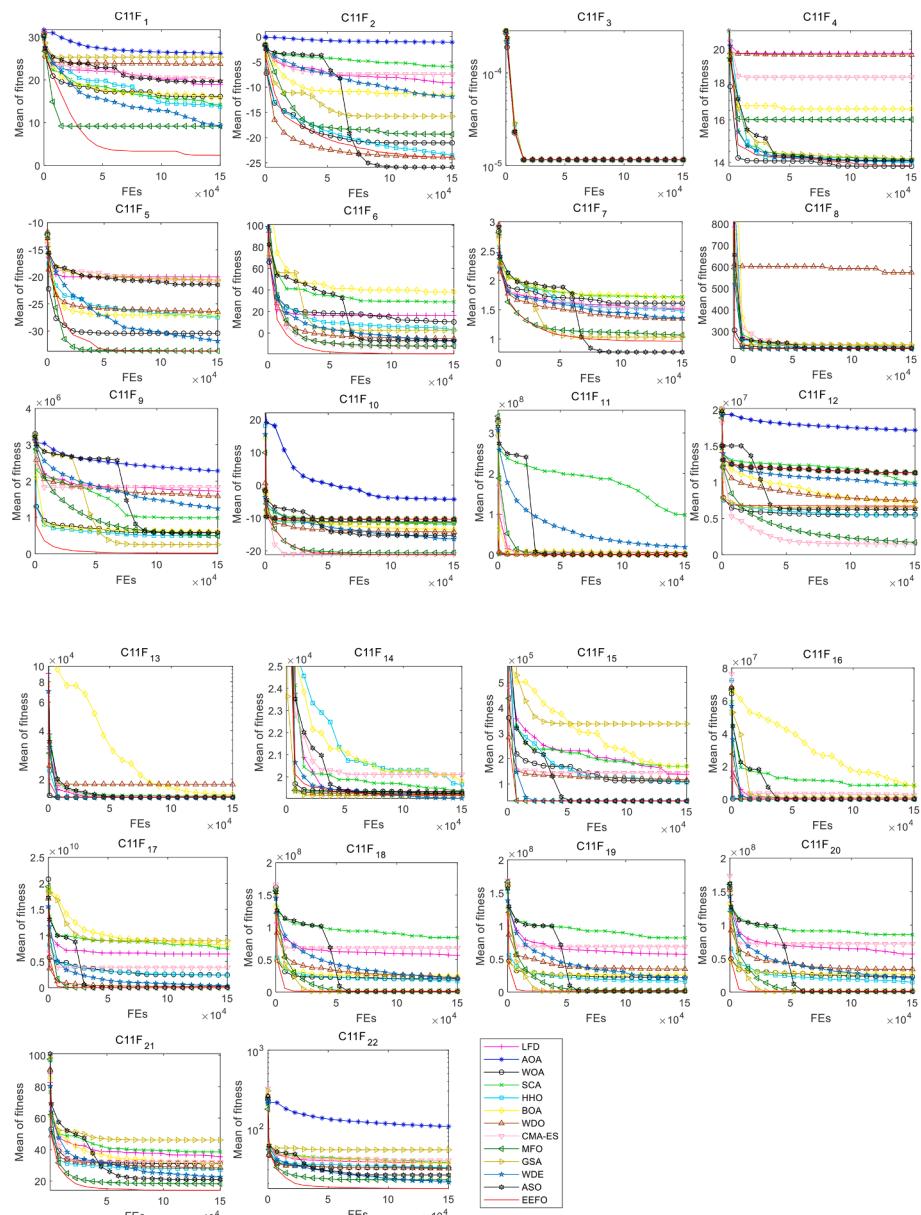


Fig. 13. Comparisons of convergence curve variations of all algorithms for CEC2011 test suite.

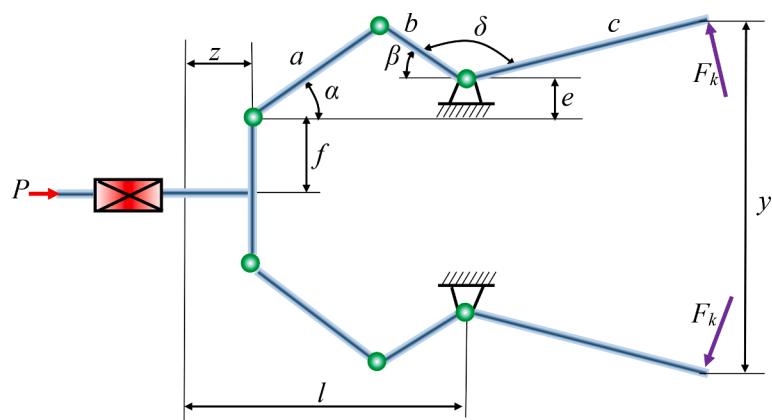


Fig. 14. Robot gripper design (Osyczka & Krenich, 2004).

$$f_3(\bar{x}) = \left(\frac{\pi}{1000} \right) \sum_{i=1}^4 \frac{b_i c_i^2 (N_{pi}^2 + N_{gi}^2)}{(N_{pi} + N_{gi})^2}, i = (1, 2, 3, 4)$$

Subject to:

$$g_{30-32}(\bar{x}) = -L_{\max} + \left(\frac{c_i(2 + N_{pi})}{N_{pi} + N_{gi}} + y_{g(i-1)} \right)_{i=2,3,4} \leq 0,$$

$$\begin{aligned} g_1(\bar{x}) &= \left(\frac{366000}{\pi \omega_1} + \frac{2c_1 N_{p1}}{N_{p1} + N_{g1}} \right) \left(\frac{(N_{p1} + N_{g1})^2}{4b_1 c_1^2 N_{p1}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0, \\ g_2(\bar{x}) &= \left(\frac{366000 N_{g1}}{\pi \omega_1 N_{p1}} + \frac{2c_2 N_{p2}}{N_{p2} + N_{g2}} \right) \left(\frac{(N_{p2} + N_{g2})^2}{4b_2 c_2^2 N_{p2}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0, \\ g_3(\bar{x}) &= \left(\frac{366000 N_{g1} N_{g2}}{\pi \omega_1 N_{p1} N_{p2}} + \frac{2c_3 N_{p3}}{N_{p3} + N_{g3}} \right) \left(\frac{(N_{p3} + N_{g3})^2}{4b_3 c_3^2 N_{p3}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0, \\ g_4(\bar{x}) &= \left(\frac{366000 N_{g1} N_{g2} N_{g3}}{\pi \omega_1 N_{p1} N_{p2} N_{p3}} + \frac{2c_4 N_{p4}}{N_{p4} + N_{g4}} \right) \left(\frac{(N_{p4} + N_{g4})^2}{4b_4 c_4^2 N_{p4}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0, \\ g_5(\bar{x}) &= \left(\frac{366000}{\pi \omega_1} + \frac{2c_1 N_{p1}}{N_{p1} + N_{g1}} \right) \left(\frac{(N_{p1} + N_{g1})^3}{4b_1 c_1^2 N_{g1} N_{p1}^2} \right) - \left(\frac{\sigma_H}{C_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334 W K_0 K_m} \right) \leq 0, \\ g_6(\bar{x}) &= \left(\frac{366000 N_{g1}}{\pi \omega_1 N_{p1}} + \frac{2c_2 N_{p2}}{N_{p2} + N_{g2}} \right) \left(\frac{(N_{p2} + N_{g2})^3}{4b_2 c_2^2 N_{g2} N_{p2}^2} \right) - \left(\frac{\sigma_H}{C_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334 W K_0 K_m} \right) \leq 0, \end{aligned}$$

$$\begin{aligned} g_7(\bar{x}) &= \left(\frac{366000 N_{g1} N_{g2}}{\pi \omega_1 N_{p1} N_{p2}} + \frac{2c_3 N_{p3}}{N_{p3} + N_{g3}} \right) \left(\frac{(N_{p3} + N_{g3})^3}{4b_3 c_3^2 N_{g3} N_{p3}^2} \right) \\ &\quad - \left(\frac{\sigma_H}{C_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334 W K_0 K_m} \right) \leq 0, \end{aligned}$$

$$\begin{aligned} g_8(\bar{x}) &= \left(\frac{366000 N_{g1} N_{g2} N_{g3}}{\pi \omega_1 N_{p1} N_{p2} N_{p3}} + \frac{2c_4 N_{p4}}{N_{p4} + N_{g4}} \right) \left(\frac{(N_{p4} + N_{g4})^3}{4b_4 c_4^2 N_{g4} N_{p4}^2} \right) \\ &\quad - \left(\frac{\sigma_H}{C_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334 W K_0 K_m} \right) \leq 0, \end{aligned}$$

$$\begin{aligned} g_{9-12}(\bar{x}) &= -N_{pi} \sqrt{\frac{\sin^2(\phi)}{4} - \frac{1}{N_{pi}} + \left(\frac{1}{N_{pi}} \right)^2} + N_{gi} \sqrt{\frac{\sin^2(\phi)}{4} + \frac{1}{N_{gi}} \left(\frac{1}{N_{gi}} \right)^2} + \\ &\quad \frac{\sin(\phi)(N_{pi} + N_{gi})}{2} + CR_{\min} \pi \cos(\phi) \leq 0, \end{aligned}$$

$$g_{13-16}(\bar{x}) = d_{\min} - \frac{2c_i N_{pi}}{N_{pi} + N_{gi}} \leq 0,$$

$$g_{17-20}(\bar{x}) = d_{\min} - \frac{2c_i N_{gi}}{N_{pi} + N_{gi}} \leq 0,$$

$$g_{21}(\bar{x}) = x_{p1} + \left(\frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \right) - L_{\max} \leq 0,$$

$$g_{22-24}(\bar{x}) = -L_{\max} + \left(\frac{(N_{pi} + 2)c_i}{N_{gi} + N_{pi}} \right)_{i=2,3,4} - x_{g(i-1)} \leq 0,$$

$$g_{25}(\bar{x}) = -x_{p1} + \left(\frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \right) \leq 0,$$

$$g_{26-28}(\bar{x}) = \left(\frac{(N_{pi} + 2)c_i}{N_{pi} + N_{gi}} - x_{g(i-1)} \right)_{i=2,3,4} \leq 0,$$

$$g_{29}(\bar{x}) = y_{p1} + \frac{(N_{p1} + 2)c_1}{N_{pi} + N_{gi}} - L_{\max} \leq 0,$$

$$g_{33}(\bar{x}) = \frac{(2 + N_{pi})c_1}{N_{pi} + N_{g1}} - y_{p1} \leq 0,$$

$$g_{34-36}(\bar{x}) = \left(\frac{c_i(2 + N_{pi})}{N_{pi} + N_{gi}} - y_{g(i-1)} \right)_{i=2,3,4} \leq 0,$$

$$g_{37-40}(\bar{x}) = -L_{\max} + \frac{c_i(2 + N_{gi})}{N_{pi} + N_{gi}} + x_{gi} \leq 0,$$

$$g_{41-44}(\bar{x}) = -x_{gi} + \left(\frac{(N_{gi} + 2)c_i}{N_{pi} + N_{gi}} \right) \leq 0,$$

$$g_{45-48}(\bar{x}) = y_{gi} + \left(\frac{(N_{gi} + 2)c_i}{N_{pi} + N_{gi}} \right) - L_{\max} \leq 0,$$

$$g_{49-52}(\bar{x}) = -y_{gi} + \left(\frac{(N_{gi} + 2)c_i}{N_{pi} + N_{gi}} \right) \leq 0,$$

$$g_{53-56}(\bar{x}) = (b_i - 8.255)(b_i - 5.715)(b_i - 12.70)(-N_{pi} + 0.945c_i - N_{gi})(-1) \leq 0,$$

$$g_{57-60}(\bar{x}) = (b_i - 8.255)(b_i - 3.175)(b_i - 12.70)(-N_{pi} + 0.646c_i - N_{gi}) \leq 0, g_{61-64}(\bar{x})$$

$$= (b_i - 5.715)(b_i - 3.175)(b_i - 12.70)(-N_{pi} + 0.504c_i - N_{gi}) \leq 0, g_{65-68}(\bar{x})$$

$$= (b_i - 5.715)(b_i - 3.175)(b_i - 8.255)(0c_i - N_{gi} - N_{pi}) \leq 0, g_{69-72}(\bar{x})$$

$$= (b_i - 8.255)(b_i - 5.715)(b_i - 12.70)(N_{gi} + N_{pi} - 1.812c_i) (-1) \leq 0, g_{73-76}(\bar{x})$$

$$= (b_i - 8.255)(b_i - 3.175)(b_i - 12.70)(-0.945c_i + N_{pi} + N_{gi}) \leq 0, g_{77-80}(\bar{x})$$

$$= (b_i - 5.715)(b_i - 3.175)(b_i - 12.70)(-0.646c_i + N_{pi} + N_{gi})(-1) \leq 0, g_{81-84}(\bar{x})$$

$$= (b_i - 5.715)(b_i - 3.175)(b_i - 8.255)(N_{pi} + N_{gi} - 0.504c_i) \leq 0,$$

Table 22

Comparison of the optimal results from various optimizers for the robot gripper problem.

Algorithms	Best	Worst	AVE	STD
EEFO	2.676837	3.623577	3.106154	0.280196
LFD	4.013065	10.634300	6.136945	1.473679
AOA	3.569583	11.203094	6.259847	1.736447
WOA	3.874202	9.359430	5.469862	1.437848
SCA	3.793475	6.761174	4.313431	0.483812
HHO	3.043292	6.510139	4.725789	0.998501
BOA	4.182258	11.478552	7.269363	2.119233
WDO	3.166186	4.846850	4.007565	0.387332
CMA-ES	3.860366	10.845009	5.247432	1.803557
MFO	3.464998	6.226422	4.404941	0.618033
GSA	11.130429	6.5095E + 16	2.7639E + 15	1.2209E + 16
WDE	3.343901	4.730015	4.106675	0.329738
ASO	4.618985	7.972438	5.794113	0.879577

$$g_{85}(\bar{x}) = \omega_{\min} - \frac{\omega_1(N_{p1}N_{p2}N_{p3}N_{p4})}{(N_{g1}N_{g2}N_{g3}N_{g4})} \leq 0,$$

$$g_{86}(\bar{x}) = \frac{\omega_1(N_{p1}N_{p2}N_{p3}N_{p4})}{(N_{g1}N_{g2}N_{g3}N_{g4})} - \omega_{\max} \leq 0,$$

Where, $c_i = \sqrt{(y_{gi} - y_{pi})^2 + (x_{gi} - x_{pi})^2}$, $K_0 = 1.5$, $d_{\min} = 25$, $J_R = 0.2$, $\phi = 2\pi/3$, $W = 55.9$, $K_M = 1.6$, $CR_{\min} = 1.4$, $L_{\max} = 127$, $C_p = 464$, $\sigma_H = 3290$, $\omega_{\max} = 255$, $\omega_1 = 5000$, $\sigma_N = 2090$, $\omega_{\min} = 245$.

Variable range:

$$b_i \in \{3.175, 12.7, 8.255, 5.715\},$$

$$y_{p1}, x_{p1}, y_{gi}, x_{gi} \in \{12.7, 38.1, 25.4, 50.8, 76.2, 63.5, 88.9, 114.3, 101.6\},$$

$$7 \leq N_{gi}, N_{pi} \leq 76 \text{ is integer.}$$

Table 26 shows the results of all considered optimizers by independently running 30 runs for this design. From **Table 26**, EEFO shows the most competitive results compared to other optimizers. The optimal results of these algorithms and their corresponding design variables for this design are provided in **Table 27**. It is evident that EEFO can provide better optimal structure parameters compared to other algorithms. **Fig. 19** illustrates the constraint values and convergence curve variations of EEFO versus the iteration number for this design.

Table 23

Comparison of the optimal results from various optimizers for robot gripper problem.

Algorithms	Value of optimal variable							Optimal value
	a	b	c	e	f	l	δ	
EEFO	148.701604	145.432573	198.454712	3.102672	100.785491	104.546235	2.099427	2.676837
LFD	149.009110	127.077024	168.862003	21.227655	63.439721	125.347345	2.115996	4.013065
AOA	149.504323	135.403898	172.924238	13.657374	120.231373	118.476668	2.433515	3.569583
WOA	149.503274	135.482067	175.327825	11.878704	92.860458	146.342582	2.329882	3.874202
SCA	150	148.970321	162.998324	0	58.146902	134.466420	2.008253	3.793475
HHO	149.921577	149.737087	173.601305	3.5542E-03	149.311991	106.353919	2.504840	3.043292
BOA	142.345329	137.865103	154.976172	3.856287	36.516064	122.899218	1.809930	4.182258
WDO	149.796546	141.376545	175.039459	8.191227	73.767450	108.694923	2.023545	3.166186
CMA-ES	149.777223	131.138662	178.070748	16.473357	119.160938	145.638531	2.521285	3.860366
MFO	146.717917	115.505228	199.999991	29.888467	136.515814	131.362701	2.602075	3.464998
GSA	124.336029	101.661045	146.032491	21.336709	80.266399	147.398251	2.527056	11.130429
WDE	147.955764	145.673261	183.540265	1.535545	129.565362	127.606903	2.407808	3.343901
ASO	145.690990	130.607206	157.559243	11.827954	119.514688	153.307744	2.627464	4.618985

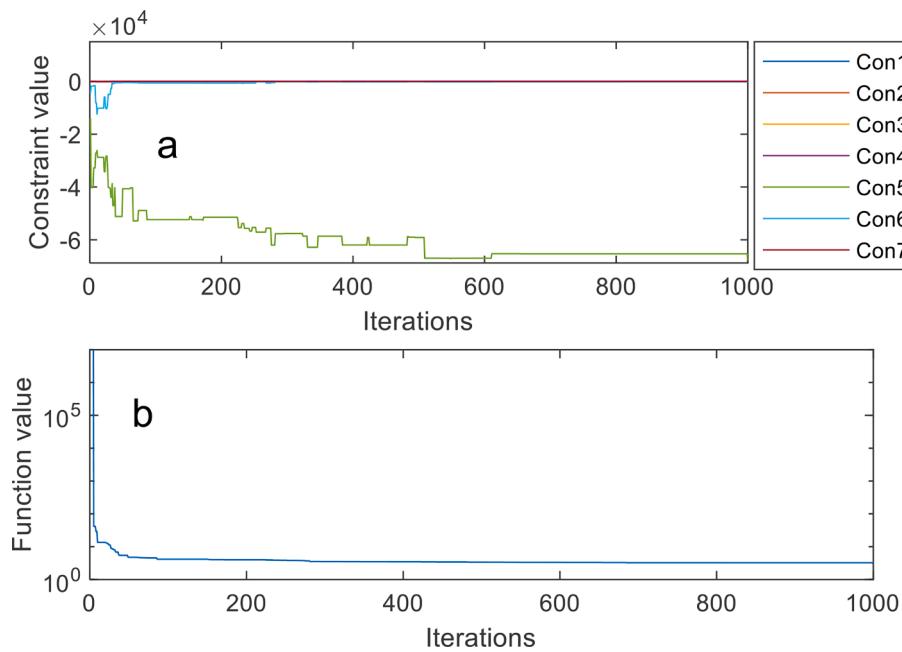


Fig. 15. (a) Constraint values versus iteration number, (b) convergence curve.

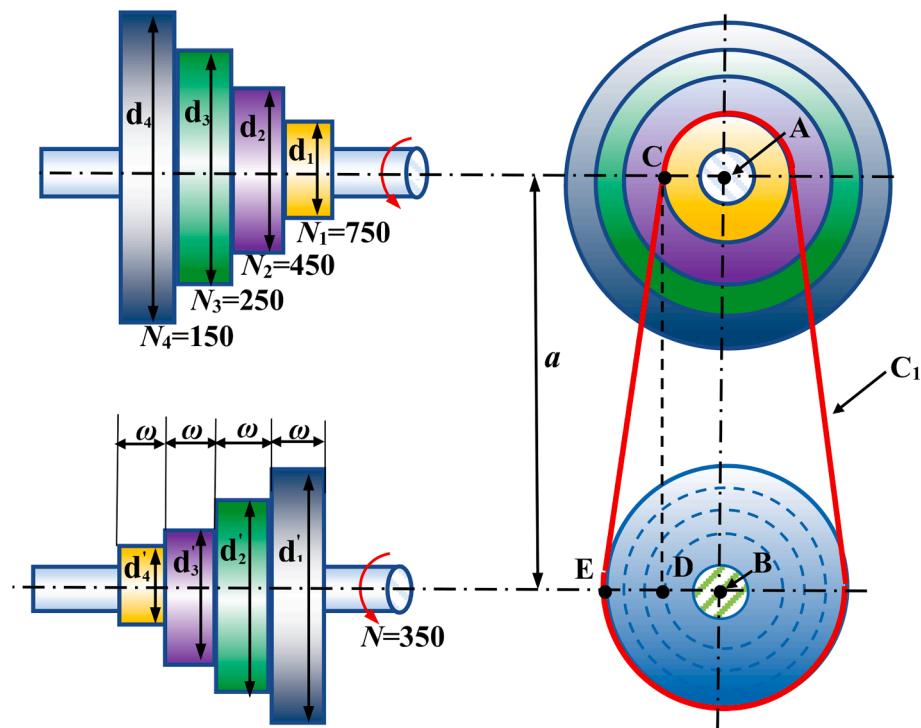


Fig. 16. Step-cone pulley design.

Table 24

Results from various optimizers for step-cone pulley problem.

Algorithms	Best	Worst	AVE	STD
EEFO	16.206355	17.000961	16.598541	0.211052
LFD	16.955916	3.92936E + 13	2.26911E + 12	7.4795E + 12
AOA	7.8393E + 09	1.01101E + 12	2.75416E + 11	2.7789E + 11
WOA	16.491043	2.1843E + 03	9.7200E + 01	3.9554E + 02
SCA	2.5095E + 11	4.5889E + 12	1.41304E + 12	1.0023E + 12
HHO	1.7853E + 3	1.3392E + 09	5.8918E + 07	2.5389E + 08
BOA	1.6506E + 11	3.3610E + 12	1.3303E + 12	9.0338E + 11
WDO	1.4685E + 07	2.9286E + 13	9.7721E + 11	5.3466E + 12
CMA-ES	16.275662	2.0267E + 13	9.8727E + 11	3.7733E + 12
MFO	16.993192	17.129580	16.708443	0.223333
GSA	16.239356	5.1229E + 11	7.5527E + 10	1.1107E + 11
WDE	8.8346E + 04	5.2913E + 06	1.5453E + 06	1.5880E + 06
ASO	16.531502	6.5290E + 09	2.1763E + 08	1.1920E + 09

Table 25

Comparison of the optimal results from various optimizers for step-cone pulley problem.

Algorithms	Optimal values for variables					Optimal weight
	d_1	d_2	d_3	d_4	ω	
EEFO	38.651446	53.185667	70.908667	85.017994	89.445575	16.206355
LFD	40.008020	55.053809	73.399126	88.001375	87.382883	16.955916
AOA	39.356550	53.571977	71.515652	86.552040	89.390715	7.8393E + 09
WOA	39.313884	54.097901	72.124784	86.474835	87.950321	16.491043
SCA	37.403905	51.633368	64.104121	86.781667	89.223636	2.5095E + 11
HHO	40.776062	56.111350	74.809536	89.690134	89.618459	1.7853E + 3
BOA	43.316331	59.729673	79.484228	89.567607	88.881936	1.6506E + 11
WDO	39.781391	54.739605	73.027458	87.510984	87.353293	1.4685E + 07
CMA-ES	38.854658	53.465505	71.281725	85.464902	88.976557	16.275662
MFO	40.574244	55.833585	74.438657	89.246603	85.195649	16.993192
GSA	38.680929	53.226267	70.962792	85.082834	89.523531	16.239356
WDE	40.666329	55.959984	74.607959	89.445000	85.381319	8.8346E + 04
ASO	39.086109	53.784232	71.706628	85.973911	89.307792	16.531502

4.7.4. Wind farm layout design

Wind farm layout is very important to determine the power output over the life cycle. This problem is the maximization of the total power

output by finding the best arrangement of wind turbines. This problem consists of 30 design variables and 91 constraints. The schematic of this design is presented in Fig. 20 (Wang et al., 2017). This design is

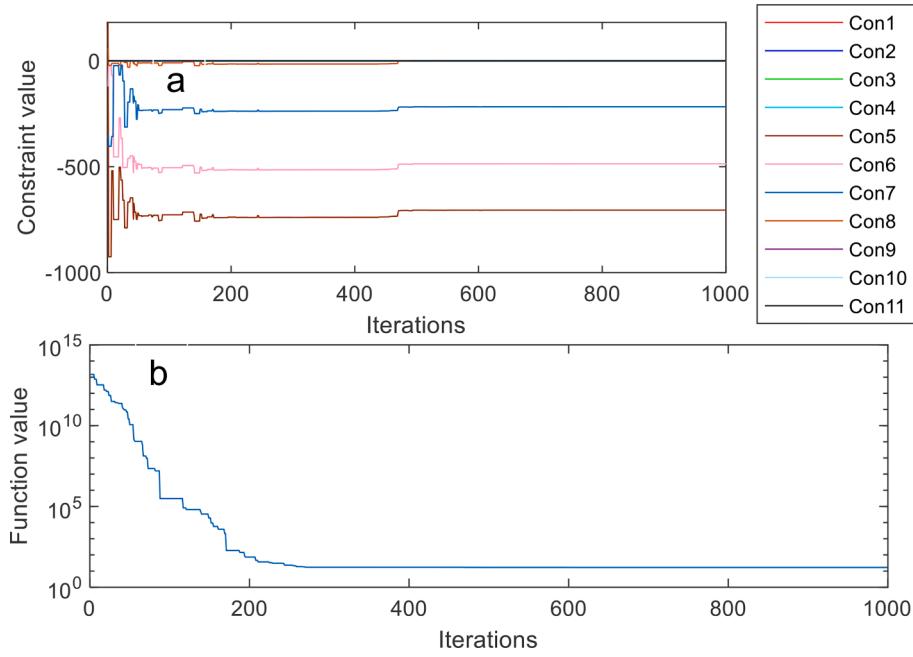


Fig. 17. (a) Constraint values versus iteration number, (b) convergence curve.

formulated below.

Consider : $\vec{x} = [x_1, y_1, x_2, y_2, \dots, x_{15}, y_{15}]$

$$\text{Minimize} : f_4 = \sum_{k=1}^{\text{Num}} E(P_i),$$

$$E(P_i) = \sum_{t=1}^H \xi_t \left\{ P_r \left(e^{-\left(v_r/c'_i((\theta_{t-1}+\theta_t)/2)\right)^{k_i((\theta_{t-1}+\theta_t)/2)}} - e^{-\left(v_{co}/c'_i((\theta_{t-1}+\theta_t)/2)\right)^{k_i((\theta_{t-1}+\theta_t)/2)}} \right) \right. \\ \left. + \sum_{j=1}^S \left(e^{-\left(v_{j-1}/c'_i((\theta_{t-1}+\theta_t)/2)\right)^{k_i((\theta_{t-1}+\theta_t)/2)}} - e^{-\left(v_j/c'_i((\theta_{t-1}+\theta_t)/2)\right)^{k_i((\theta_{t-1}+\theta_t)/2)}} \right) \frac{e^{(v_{j-1}+v_j)/2}}{5 + e^{(v_{j-1}+v_j)/2}} \right\},$$

Subject to:

$$40 \leq x_i \leq 1960,$$

$$40 \leq y_i \leq 1960,$$

$$\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \geq 200, \quad j = 1, 2, \dots, \text{Num} \text{ and } j \neq i$$

where $j = 1, 2, \dots, \text{Num}$ and $j \neq i$, and ξ_n implies the spectrum of frequencies in the range $[\theta_{n-1}, \theta_n]$, $\text{Num} = H = 15$, $v_r = 14$, $v_{co} = 3.5$, $S = (v_r - v_{co}) / 0.3$.

This problem is attempted by all the algorithms over 30 independent runs and the results are shown in Table 28. From the table, EEFO achieves the best performance concerning the 'Best', 'Worst', and 'AVE' metrics, which indicates that EEFO can achieve superior results over other algorithms when solving this problem. The optimal results of all the algorithms and their corresponding design variables are provided in Table 29 and 30. According to the two tables, EEFO can find a better

layout scheme compared to other algorithms. Fig. 21 illustrates the constraint values and convergence curve variations of EEFO versus the iteration number.

4.7.5. Pressure vessel design

This design minimizes a pressure vessel's total cost, as depicted in Fig. 22 (Kannan & Kramer, 1993; Wang et al., 2022c). There are four

variables to be optimized. There are four constraints with four design variables in this problem. This design is formulated below.

Consider variable $\vec{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$.

$$\text{Minimize} f_5(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3.$$

Subject to

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0.$$

Variable range: $0 \leq x_1 \leq 99, 0 \leq x_2 \leq 99, 10 \leq x_3 \leq 200, 10 \leq x_4 \leq 200$.

EEFO and its competing algorithms are used to optimize the design variables over 30 independent runs, and the results in optimizing this

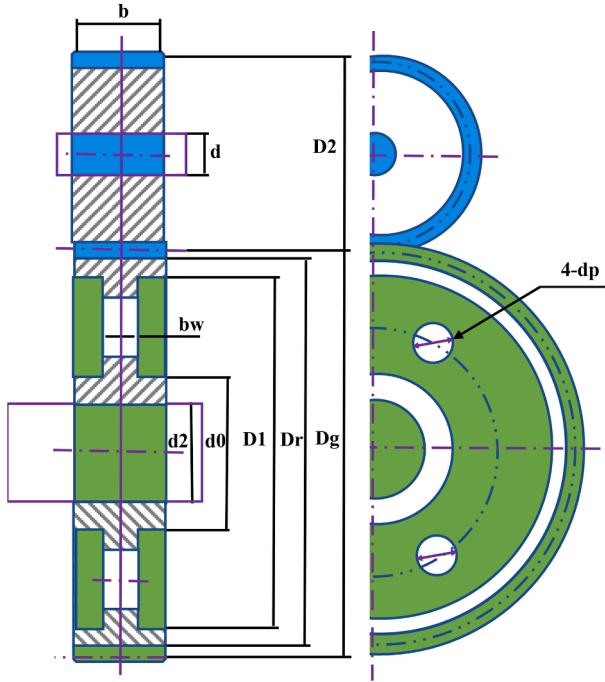


Fig. 18. Four-stage gearbox design.

problem are listed in Table 31. As shown in Table 31, EEFO shows superior performance over other optimizers in terms of the 'Best', 'Worst', 'AVE', and 'STD' metrics in solving the pressure vessel design. The best results of these algorithms and their corresponding design variables are provided in Table 32. It can be observed that the optimal cost offered by EEFO is superior to those offered by other algorithms. Thus, EEFO provides a more competitive design scheme compared to its competitors. Fig. 23 illustrates this engineering application's constraint values and convergence curve variations of EEFO versus the iteration number.

4.7.6. Rolling element bearing design

This problem has ten design variables and nine constraints subject to the maximum dynamic load-carrying capacity of the rolling element bearing (Rao & Tiwari, 2007). The schematic of this case is presented in Fig. 24. This design is formulated below.

Consider variable: $\vec{x} = [D_m, D_b, Z, f_i, f_o, K_{D\min}, K_{D\max}, \epsilon, e, \zeta]$.

$$\text{Maximize} \begin{cases} f_6(\vec{x}) = f_o Z^{2/3} D_b^{1.8} & \text{if } D_b \leq 25.4 \text{ mm} \\ f_6(\vec{x}) = 3.647 f_c Z^{2/3} D_b^{1.4} & \text{if } D_b > 25.4 \text{ mm} \end{cases}$$

Subject to

$$g_1(\vec{x}) = \frac{\phi_o}{2 \sin^{-1}(D_b/D_m)} - Z + 1 \geq 0,$$

$$g_2(\vec{x}) = 2D_b - K_{D\min}(D - d) \geq 0,$$

$$g_3(\vec{x}) = K_{D\max}(D - d) - 2D_b \geq 0,$$

$$g_4(\vec{x}) = D_m - (0.5 - e)(D + d) \geq 0,$$

$$g_5(\vec{x}) = (0.5 + e)(D + d) - D_m \geq 0,$$

$$g_6(\vec{x}) = D_m - 0.5(D + d) \geq 0,$$

$$g_7(\vec{x}) = 0.5(D - D_m - D_b) - \epsilon D_b \geq 0,$$

$$g_8(\vec{x}) = \zeta B_w - D_b \leq 0.$$

$$g_9(\vec{x}) = f_i \geq 0.515,$$

$$g_{10}(\vec{x}) = f_o \geq 0.515.$$

Where

Table 26
Results from various optimizers for four-stage gearbox design.

Algorithm	Best	Worst	AVE	STD
EEFO	37.432886	3.7369E + 17	1.7225E + 16	6.8265E + 16
LFD	2.4736E + 13	2.6393E + 18	6.5634E + 17	7.1382E + 17
AOA	1.5248E + 17	6.4896E + 18	2.8509E + 18	1.4618E + 18
WOA	66.623946	7.9369E + 18	8.3337E + 17	1.8770E + 18
SCA	2.2014E + 17	7.8757E + 18	2.5162E + 18	1.6792E + 18
HHO	2.4736E + 13	5.3663E + 18	1.1436E + 18	1.5119E + 18
BOA	1.6322E + 18	3.0778E + 19	1.1509E + 19	7.3268E + 18
WDO	87.326100	1.3975E + 22	5.6167E + 20	2.5859E + 21
CMA-ES	64.327821	2.8229E + 19	6.3011E + 18	7.0691E + 18
MFO	40.187791	1.1190E + 17	4.8384E + 16	2.0370E + 16
GSA	2.2503E + 18	2.6758E + 19	1.0535E + 19	6.9269E + 18
WDE	77.885287	4.7579E + 17	1.1739E + 17	9.5531E + 16
ASO	2.0656E + 18	3.9753E + 19	1.0652E + 19	7.3918E + 18

$$f_c = 37.91 \left[1 + \left\{ 1.04 \left(\frac{1-\gamma}{1+\gamma} \right)^{1.72} \left(\frac{f_i(2f_o-1)}{f_o(2f_i-1)} \right)^{0.4} \right\}^{10/3} \right]^{-0.3}$$

$$\left(\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{f_o(1+\gamma)^3} \right) \left(\frac{2f_i}{2f_i-1} \right)^{0.41},$$

$$\gamma = \frac{D_b}{D_m}, f_i = \frac{r_i}{D_b}, f_o = \frac{r_o}{D_b},$$

$$\phi_o = 2\pi - 2\cos^{-1} \frac{\left(\frac{D-d}{2} - \frac{3T}{4} \right)^2 + \left(\frac{D}{2} - \frac{T}{4} - D_b \right)^2 - \left(\frac{D}{2} + \frac{T}{4} \right)^2}{2 \left(\frac{D-d}{2} - \frac{3T}{4} \right) \left(\frac{T}{2} - \frac{T}{4} - D_b \right)},$$

$$T = D - d - 2D_b, D = 160, d = 90, B_w = 30, r_i = r_o = 11.033.$$

$$\text{Variable range: } 0.5(D+d) \leq D_m \leq 0.6(D+d), \quad 0.15(D-d) \leq D_b \leq 0.45(D-d), \quad 4 \leq Z \leq 50, \quad 0.515 \leq f_i \leq 0.6, \quad 0.515 \leq f_o \leq 0.6, \quad 0.4 \leq K_{D\min} \leq 0.5, \quad 0.6 \leq K_{D\max} \leq 0.7, \quad 0.3 \leq \epsilon \leq 0.4, \quad 0.02 \leq e \leq 0.1, \quad 0.6 \leq \zeta \leq 0.85, \quad 0.6 \leq \zeta \leq 0.85.$$

The results of EEFO are compared with those of other different algorithms over 30 independent runs. The results are provided in Table 33. EEFO provides very competitive results for this practical problem. Although DE achieves better results than EEFO in terms of the 'Best', 'Worst', 'AVE', and 'STD' metrics, EEFO offers better results regarding the 'Best' metric. Table 34 shows the optimal load-carrying capacity and their corresponding design variables. It is evident that EEFO provides the maximum load-carrying capacity. Fig. 25 depicts the design case's constraint values and convergence curve variations of EEFO versus the iteration number.

4.7.7. Tension/compression spring design

This problem addresses designing a spring with three geometrical parameters to minimize the coil weight, and it has four constraints (Coello & Montes, 2002). The schematic of this design is presented in Fig. 26. This design is formulated below.

Consider variable $\vec{x} = [x_1, x_2, x_3] = [d, D, N]$

$$\text{Minimize } f_7(\vec{x}) = (x_3 + 2)x_2 x_1^2$$

$$\text{Subject to } g_1(\vec{x}) = 1 - \frac{x_3 x_2^3}{71785 x_1^4} \leq 0,$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0.$$

Table 27

Comparison of the optimal results from various optimizers for four-stage gearbox problem.

Algorithm	Optimal value for variables						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
EEFO	17.445281	44.485975	14.783622	30.772381	24.324112	44.020792	20.874264
LFD	37.637717	46.834264	19.730586	70.467257	16.404021	31.471389	20.366435
AOA	18.029929	59.501287	23.332087	68.035371	20.045343	42.357201	44.460043
WOA	13.695435	27.550376	14.966530	47.558615	41.282046	54.136882	23.810513
SCA	28.313845	64.789538	16.770875	56.434390	24.780116	36.376940	20.163726
HHO	16.133799	24.895708	26.924145	57.725328	13.184951	21.776925	15.501485
BOA	35.356311	26.444026	6.832377	52.251871	30.635573	36.028237	11.686101
WDO	24.473295	53.048385	14.477326	45.054370	28.108811	59.563085	25.672978
CMA-ES	22.307150	52.049625	19.509448	38.620044	23.829203	57.709996	13.525669
MFO	17.693004	42.886681	17.079501	34.484502	23.622701	41.627571	22.896845
GSA	38.642299	39.019812	9.614480	59.610632	18.039299	60.996675	45.166700
WDE	19.837833	49.496507	21.046813	28.570988	24.753773	62.861762	27.293266
ASO	8.475008	51.187992	50.418801	51.549812	25.376971	60.090981	40.783913
Optimal value for variables.							
x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
41.838153	0.753971	0.836714	1.279379	1.188356	4.688477	3.282864	3.624574
47.022098	1.074221	1.276214	2.425720	1.376741	5.981362	3.123817	4.318832
40.263562	0.906577	0.768733	0.646864	0.877996	8.731293	5.857913	4.805690
56.477151	1.396877	1.294484	1.043288	1.416997	2.645023	3.069124	4.574661
36.250106	1.484830	1.475543	2.054789	1.654628	1.909066	2.775098	5.473082
56.218765	2.051061	1.134668	1.231168	1.374681	2.115733	5.866033	6.021710
37.628873	0.745402	2.336541	2.608258	0.918414	8.198838	8.206548	2.849927
34.190012	1.100616	1.263556	1.479523	2.071732	3.483059	4.155568	5.648477
24.935033	0.753296	0.626453	1.278771	2.004137	6.649128	4.783701	6.626441
53.990293	0.635968	1.248096	1.356063	0.782509	6.176685	2.721936	3.154595
40.155531	1.439706	1.535853	0.613808	1.348342	1.284761	2.850879	5.718719
62.695084	0.517273	1.800256	0.605036	0.647936	2.778621	3.238927	4.732536
47.959087	0.519693	1.426322	1.184840	1.185319	7.551948	5.404067	6.288541
Optimal value for variables.							
x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	Minimum cost
4.754881	4.912297	2.177180	4.600120	4.747928	4.805219	4.570616	37.432886
3.467535	5.609048	8.305588	5.387493	4.020067	4.933296	3.677498	2.4736E + 13
6.386582	3.197182	3.871099	5.716135	3.138843	4.055215	3.995468	1.5248E + 17
5.609279	6.357723	2.478595	4.706837	6.443806	5.302408	5.769719	66.623946
6.503795	7.233725	1.336924	5.370793	5.492490	4.337571	3.489064	2.2014E + 17
3.528526	4.416383	1.911595	3.972118	4.809154	3.521701	5.780892	2.4736E + 13
4.371292	5.064211	2.792283	6.891733	2.677795	7.846345	3.033374	1.6322E + 18
5.884737	6.486746	7.465970	2.627066	3.998992	3.880453	3.346995	87.326100
5.898063	3.738672	1.580149	4.843867	5.377133	5.801686	5.192123	64.327821
6.460278	3.275658	4.876974	7.158584	5.076158	7.656566	2.842956	40.187791
6.279568	6.635609	2.698913	6.497961	6.702368	5.273082	3.229945	2.2503E + 18
6.347275	6.966194	8.394204	4.288245	4.179625	5.362807	7.479845	77.885287
3.035683	1.678737	7.298103	6.303158	2.368085	2.780107	5.381558	2.0656E + 18

Variable range: $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, $2 \leq x_3 \leq 15$.

The results from EEFO and different optimizers over 30 independent runs are provided in Table 35. EEFO obtains better results in terms of the 'Best', 'Worst', 'AVE', and 'STD' metrics compared to other algorithms, indicating that EEFO is most stable in providing more accurate solutions. The optimal results of these algorithms and their corresponding design variables are shown in Table 36. Fig. 27 illustrates the constraint values and convergence curve variations of EEFO versus iteration number for this design.

4.7.8. Cantilever beam design

For this well-known case, this design aims to minimize the beam weight. Fig. 28 depicts the structure of the design (Chickermane & Gea, 1996). The design has five design variables and one constraint. The design is formulated below.

Consider variable $\vec{x} = [x_1, x_2, x_3, x_4, x_5]$

$$\text{Minimize } f_8(\vec{x}) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$$

$$\text{Subject to } g_1(\vec{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$$

Variable range $0.01 \leq x_i \leq 100$, $i = 1, \dots, 5$.

EEFO and its competitors tackle this case, and their results over 30 independent runs are presented in Table 37. Based on Table 37, the 'Best', 'Worst', 'AVE', and 'STD' metrics provided by EEFO are second only to TLBO. In addition, EEFO can provide the best result concerning the 'Best' metrics. The best results of these algorithms and their corresponding design variables are shown in Table 38. It is evident that EEFO provides the most accurate solution, which outperforms those provided by other optimizers in tackling this problem. Fig. 29 illustrates this design's constraint values and convergence curve variations of EEFO versus iteration number.

4.7.9. Welded beam design

This problem is to minimize cost with two constraints when designing a welded beam. There are four structure variables related to this problem. The schematic of the design is presented in Fig. 30 (Zhao et al., 2022b). This design is formulated below.

Consider variable: $\vec{x} = [h, l, t, b]$

$$\text{Minimize: } f_9(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

$$\text{Subject to: } g_1(\vec{x}) = \tau(\vec{x}) + \tau_{\max} \leq 0, \quad g_2(\vec{x}) = \sigma(\vec{x}) + \sigma_{\max} \leq 0,$$

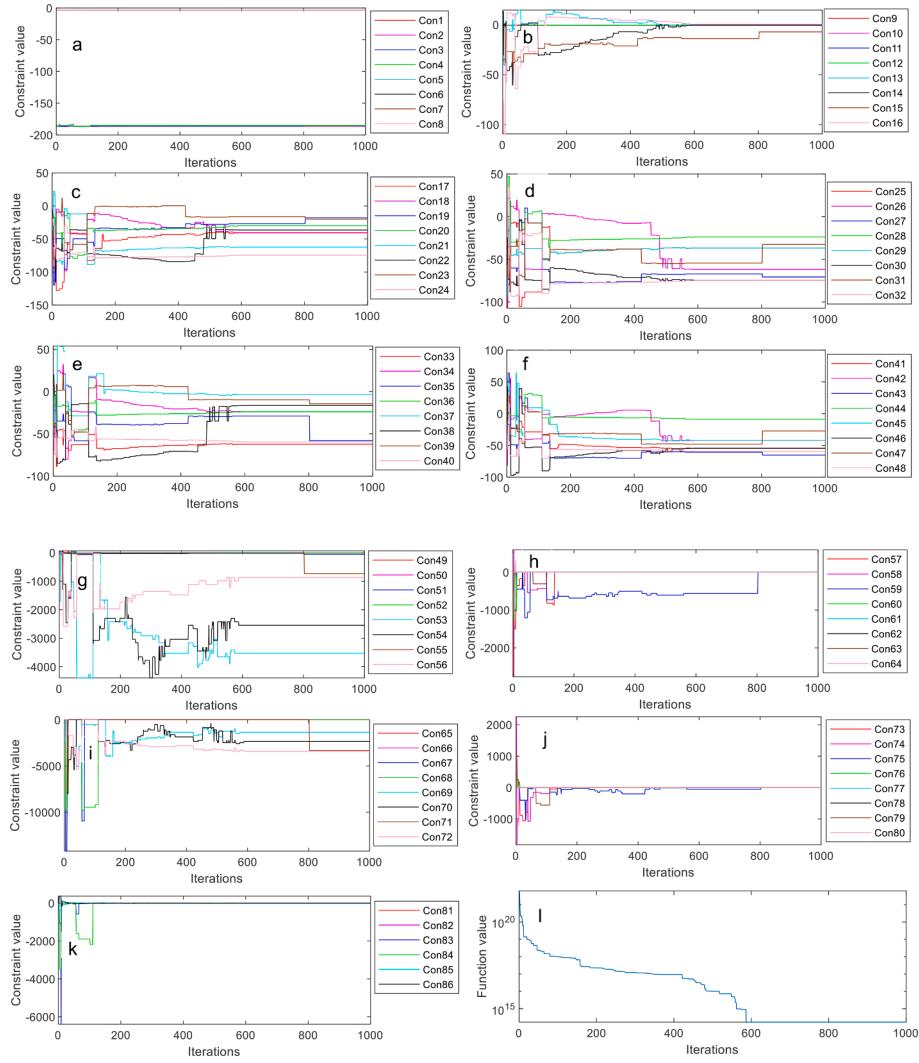


Fig. 19. (a-k) Constraint values versus iteration number, (l) convergence curve.

$$g_3(\vec{x}) = \delta(\vec{x}) + \delta_{\max} \leq 0, \quad g_4(\vec{x}) = x_1 - x_4 \leq 0, \quad g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0, \\ g_6(\vec{x}) = 0.125 - x_1 \leq 0, \quad g_7(\vec{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0.$$

Where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau' \cdot \frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right), \\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \quad J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^3}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \quad \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \\ \delta(\vec{x}) = \frac{4PL^3}{Ex_4x_3^3}, \quad P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3x_4}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$$P = 6000 \text{ lb}, \quad L = 14 \text{ in}, \quad E = 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi}, \quad \tau_{\max} \\ = 13600 \text{ psi}, \quad \sigma_{\max} = 30000 \text{ psi}, \quad \delta_{\max} = 0.25 \text{ in}.$$

Variable range: $0.1 \leq x_1 \leq 2$, $0.1 \leq x_2 \leq 10$, $0.1 \leq x_3 \leq 10$, $0.1 \leq x_4 \leq 2$.

The results of all considered algorithms for solving this design are presented in Table 39. EEFO provides the same performance as TLBO concerning the 'Worst', 'AVE', and 'STD' metrics, which are superior to those provided by the remaining algorithms. The optimal results of these algorithms and their corresponding design variables are provided in Table 40, and EEFO obtains the same structural parameters as MFO with the same minimum cost. Fig. 31 illustrates the constraint values and

convergence curve variations of EEFO versus the iteration number.

4.7.10. Speed reducer design

This problem is the minimization of the weight of the speed reducer. It has 7 geometric variables subject to 11 constraints (Coello Coello & Pulido, 2005). Fig. 32 shows the speed reducer schematic. The design is formulated below.

Design variable : $\vec{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2]$

$$\text{Minimize: } f_{10}(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\ - 1.508x_1(x_6^2 + x_7^2) + 0.7854x_1(x_4x_6^2 - x_5x_7^2)$$

$$\text{Subject to: } g_1(\vec{x}) = \frac{27}{x_1x_2x_3} - 1 \leq 0, \\ g_2(\vec{x}) = \frac{397.5}{x_1x_2x_3^2} - 1 \leq 0,$$

$$g_3(\vec{x}) = \frac{1.93x_4^2}{x_2x_6^4x_3} - 1 \leq 0,$$

$$g_4(\vec{x}) = \frac{1.93x_5^2}{x_2x_7^4x_3} - 1 \leq 0,$$

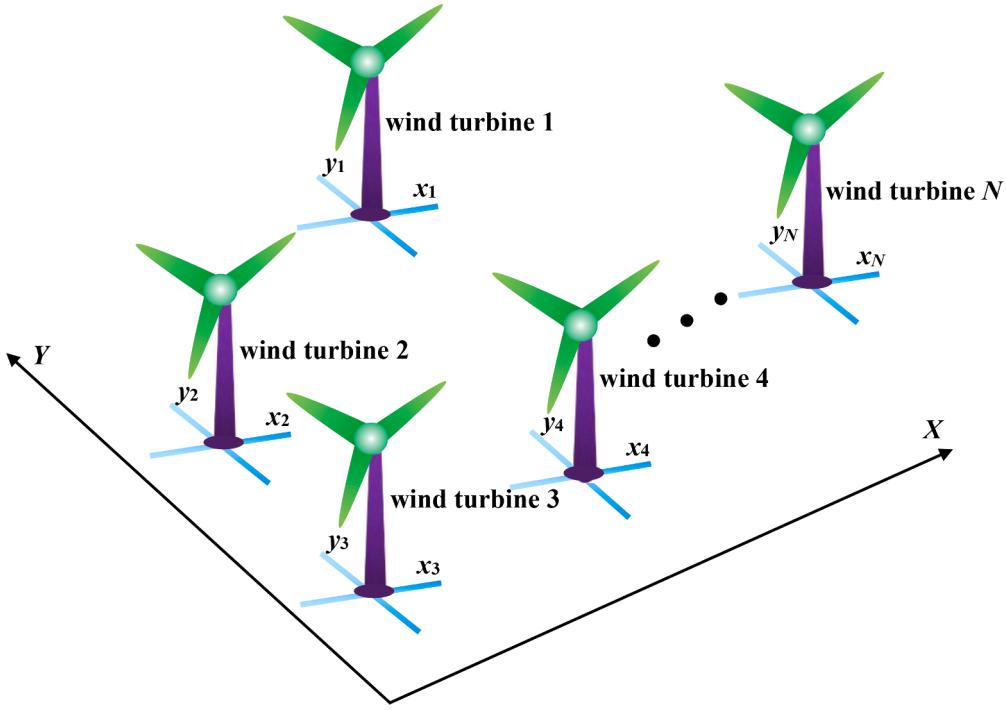


Fig. 20. Wind farm layout design.

Table 28

Results from various optimizers for the wind farm layout problem.

Algorithms	Best	Worst	AVE	STD
EEFO	6033.929894	5649.343793	5799.090075	99.442492
LFD	5515.028012	5179.503063	5369.359470	142.332946
AOA	5089.287758	4957.590806	5070.141852	76.082972
WOA	5422.624781	5155.976186	5320.474140	107.690918
SCA	5103.427557	4937.359105	4992.272953	36.653728
HHO	5800.778593	5144.857651	5328.026401	136.499189
BOA	5197.129277	5001.031744	5056.767985	61.361051
WDO	5633.798806	5194.493002	5333.945040	132.053014
CMA-ES	5111.589811	4956.574257	5069.883825	69.031188
MFO	5745.394216	5410.716260	5598.509025	139.412989
GSA	4832.737377	4230.413058	4528.130577	274.964942
WDE	5671.063524	5477.693325	5546.037279	48.855360
ASO	5355.807137	4948.643891	5155.510259	144.548734

$$g_5(\vec{x}) = \frac{\left(\left(\frac{745x_4}{x_2x_3} \right)^2 + 16.9 \times 10^6 \right)^{0.5}}{110x_6^3} - 1 \leq 0,$$

$$g_6(\vec{x}) = \frac{\left(\left(\frac{745x_5}{x_2x_3} \right)^2 + 157.5 \times 10^6 \right)^{0.5}}{85x_7^3} - 1 \leq 0,$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \leq 0,$$

$$g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \leq 0,$$

$$g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,$$

$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \sqrt{a^2 + b^2}.$$

Variable range: $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4 \leq 8.3$, $7.3 \leq x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, $5.0 \leq x_7 \leq 5.5$.

All the optimizers are employed to find the solutions of this problem over 30 runs, and the results are listed in Table 41. EEFO obtains the best results for the 'Best', 'Worst', 'AVE' and 'STD' metrics, followed by the remaining algorithms. The optimal results of all the optimizers and their corresponding design variables are reported in Table 42. According to Table 42, EEFO can find the best designs corresponding to the minimum weight. Fig. 33 illustrates the constraint values and convergence curve variations of EEFO versus the iteration number.

4.8. Results discussion and analysis

This section summarizes all experimental results for different benchmark suites and highlights the main findings. The above experiments can be summarized into three categories. First, for the classic 23 test functions, EEFO achieves the best results at 87 %. From Table 5, EEFO attains the best solutions for all unimodal functions but 1 function. These results are attributed to its excellent exploration capability. Second, from Tables 12–14, for the CEC2017 test suite with 10, 30, and 50 dimensions, EEFO achieves the best results at 66 %, 52 % and 45 %, respectively. For the composition functions, EEFO exhibits superior balancing capability for exploration and exploitation as well as the ability to avoid local optima. Eventually, from Table 19, EEFO obtains the best results for 59 % of the CEC2011 engineering test suite and outperforms other algorithms for most of the 22 engineering cases. In these different engineering cases, there are many problems with numerous constraints and variables. For example, the four-stage gearbox problem has 22 variables and 86 constraints, the dynamic economic dispatch problem has 216 variables and 95 constraints, and the wind farm layout problem has 30 variables and 91 constraints. For these complex constraint problems, EEFO exhibits a superior optimization algorithm. These achievements demonstrate its strong competitiveness in tackling challenging problems with numerous constraints and variables.

Based on the aforementioned analysis, the subsequent findings are obtained. The algorithm demonstrates robust optimization capabilities

Table 29

Comparison of the optimal results from various optimizers for wind farm layout problem in x-axis.

Algorithms	Value of optimal variable						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
EEFO	1148.601305	1679.257546	1687.602214	1909.268285	254.045756	1692.167946	229.466900
LFD	1546.343337	531.503380	1805.002812	40.000012	969.520968	1900.235081	1685.573081
AOA	1839.307168	1425.683432	1859.572955	829.697142	1399.901081	1957.729966	614.591960
WOA	930.470155	1704.559888	905.499153	388.490069	963.558637	289.560913	1881.701003
SCA	1031.583568	42.240824	1182.661912	241.990441	1293.771089	185.622754	1593.298491
HHO	1498.364871	198.139674	937.393959	1555.444396	1898.460240	1150.693548	585.991742
BOA	900.241304	1420.667895	757.492567	1764.180415	107.169801	1571.934446	46.312327
WDO	93.969988	1010.685178	897.657067	54.444197	1164.263372	144.641071	58.997437
CMA-ES	1071.571991	1892.397528	380.009523	1940.069479	104.084907	1610.070639	1542.248998
MFO	1175.697136	1436.440266	1593.823615	204.667109	1598.392392	320.083554	1450.826180
GSA	1789.644313	1956.689538	558.918969	1644.046116	744.587875	1049.148540	485.780327
WDE	86.973981	463.737840	1912.092517	1948.380948	237.246813	1758.440884	192.944793
ASO	184.761026	549.860756	1883.311115	201.557669	446.981749	485.812886	1584.943733
Value of optimal variable							
x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
585.264493	304.874410	233.059165	1268.319395	1375.487220	710.095065	572.521770	1257.162603
1950.538434	272.794786	1641.407505	648.705871	1804.273214	894.282820	1019.868547	1184.721723
594.847399	1779.421102	1033.114605	424.119705	534.493439	137.709261	665.336698	1659.671999
1065.323873	450.832851	1930.662130	225.863050	538.771267	1487.382410	143.558059	626.378025
351.811701	1011.572025	1841.442695	968.316444	1723.664559	928.053727	1638.252236	65.749555
46.295060	1959.814080	239.388563	1649.835446	162.202991	41.600515	192.426979	1414.525827
67.273972	544.124909	1775.655847	1076.072897	353.217847	1538.170519	1557.649660	45.088895
1896.137380	1904.009874	1106.346833	76.150410	1471.768421	560.085030	1916.479410	61.994914
147.598439	1875.032833	143.217990	503.524855	1220.722569	1671.442985	158.036607	638.163983
1782.365231	253.018785	1933.939392	1360.836983	683.623834	938.725194	813.167100	630.072996
1349.108388	1802.792825	273.303905	1488.575576	484.216804	1035.475824	471.405779	1657.252181
749.889457	1771.178027	1930.486931	1585.615190	1025.455095	40.417208	860.645732	411.959485
380.535729	1775.249258	1517.053599	1423.617858	1668.951510	1532.230936	1547.090999	1638.627575

for multimodal and composite functions. These attributes are a result of the algorithm's exceptional exploration capability, its ability to balance exploration and exploitation, and its capability to avoid local optima. These search capabilities fundamentally depend on the algorithm's multi-strategy search mechanism and the energy factor. This multi-strategy search mechanism can effectively help the algorithm explore a wider solution space, maintain solution diversity, and thereby avoid getting stuck in local optima. Meanwhile, the energy factor can adaptively adjust the search behaviors, resulting in a better balance between exploration and exploitation.

5. Application of EEFO in sluice opening control under a tripping accident of a hydropower station

Hydropower, a vital renewable, low-carbon energy source, generates 17 % of global electricity. It is the third largest energy source after coal and natural gas (IEA, 2021). Therefore, the normal operation of hydropower stations is essential (Ma et al., 2022; Kumar & Saini, 2022; Melani et al., 2016). During the operation process of the hydropower station, due to the power grid or other possible accidents, extreme tripping accidents may occur (Wang et al., 2022a). That is, all units are unable to provide output. At this time, it can be considered that the unit flow rate is 0. For hydropower stations with small storage capacity, when the flow rate suddenly drops to 0, if appropriate measures are not taken as soon as possible, the water level behind the dam will quickly exceed the water level limit, resulting in serious accidents. Therefore, the normal operation of the flood sluice is crucial to the safety of the dam and plant equipment as well as the ecological safety downstream of the dam. To ensure the generation efficiency and operation safety of hydropower stations, reasonable water level control through the sluice will be important and complex work once the tripping accident occurs. In this study, the optimization model of the hydropower station by controlling the sluice opening under the condition of a tripping accident is established, and the proposed EEFO is employed to find the optimal sluice opening under extreme operating conditions. A schematic of the hydropower station sluice opening control is depicted in Fig. 34.

According to the principle of water balance, the objective function established to control the sluice water level is as follows (Wang et al., 2022a):

$$F = \min \left(\sqrt{\frac{1}{N} \sum_{i=1}^N (Z_i - Z_{obj})^2} + |Z_N - Z_{obj}| \right) \quad (38)$$

Where Z_{obj} is the objective water level, Z_i is the water level at the end of the i th time, and Z_N is the final water level at the end of the whole regulation time.

When a tripping accident occurs, the change process of the water level is expressed as (Wang et al., 2022a):

$$Z^{t+1} = f(V^{t+1}) = f(V^t + \Delta V^t) \quad (39)$$

Where Z^{t+1} is the water level at the $(t+1)$ th time, V^{t+1} is the reservoir capacity at the $(t+1)$ th time, $f(\cdot)$ is the water level-reservoir capacity function, and ΔV^t is the change in reservoir capacity from the t th time to the $(t+1)$ th time, which is given as:

$$\Delta V^t = (Q_{in}^t - Q_{out}^t) \Delta t \quad (40)$$

Where Δt is the time step, Q_{in}^t and Q_{out}^t is the average inflow and outflow from the t th time to the $(t+1)$ th time, respectively, and Q_{out}^t is calculated as:

$$Q_{out}^t = 0.5(Q_{out,t} + Q_{out,t+1}) \quad (41)$$

The outflow $Q_{out,t}$ at time $t+1$ can be calculated according to the water level-slue opening-flow curve, so the water level Z^{t+1} and the outflow $Q_{out,t+1}$ at time $t+1$ can be solved iteratively.

When a tripping accident occurs, the flow of the generating unit is 0, so the tripping flow is expressed as:

$$Q_{tr} = Q_{in}^t - Q_{out}^t \quad (42)$$

The time required for the reservoir to be raised from its current initial level to its normal level is the limited reaction time, which is defined as:

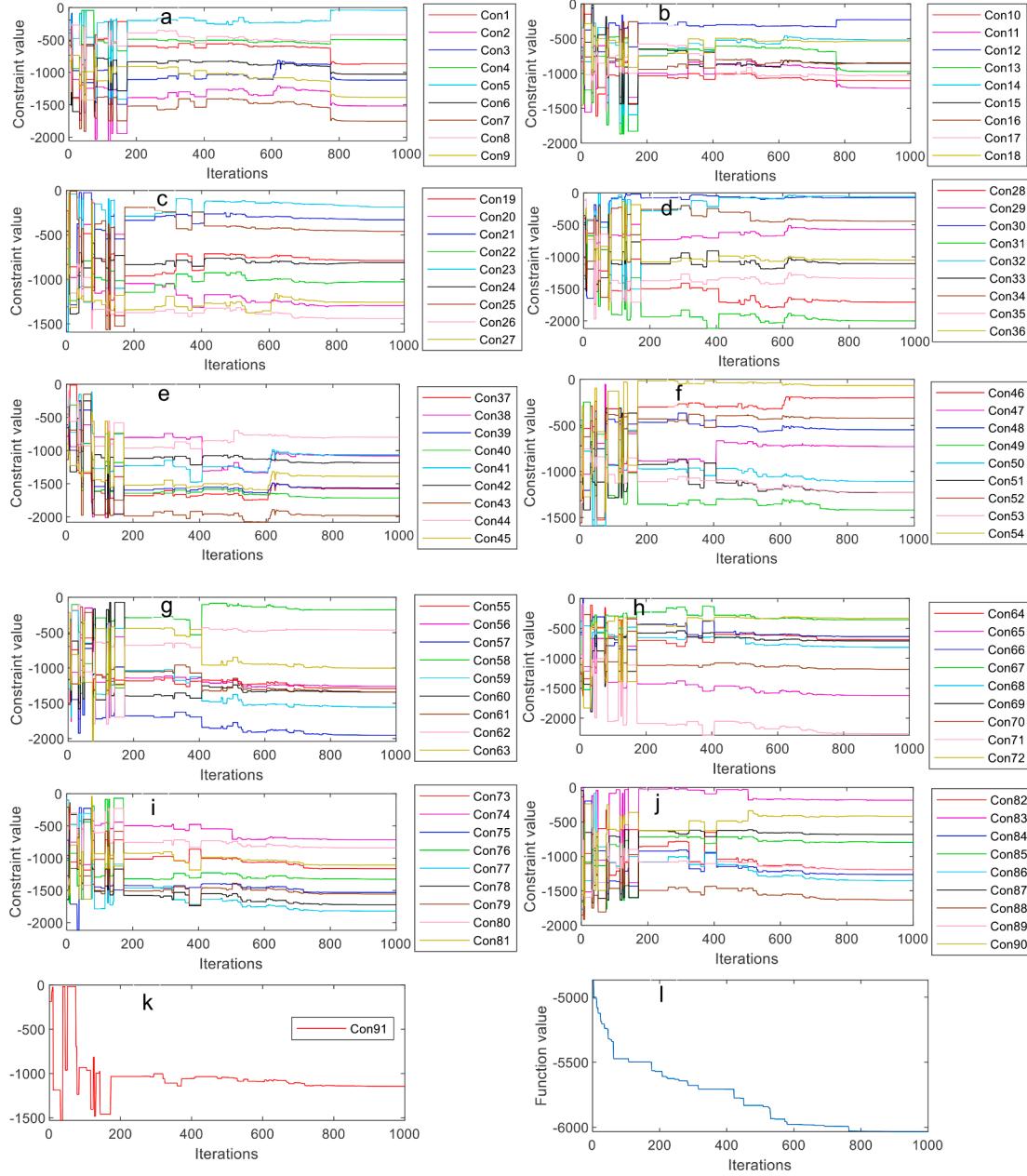


Fig. 21. (a-k) Constraint values versus iteration number, (l) convergence curve.

$$T_{limit} = \frac{V(Z_{st}) - V(Z_0)}{\bar{Q}_{in}} \quad (43)$$

Where $V(Z_{st})$ is the reservoir capacity at the normal water level, $V(Z_0)$ is the reservoir capacity at the initial storage water level at a certain time, and \bar{Q}_{in} is the overall average inflow.

The water level of the reservoir rises from the initial water level to the normal storage water level within a specific reaction time, and the corresponding initial water level is the limit operating water level, which is given as:

$$Z_{limit-op} = Z(V(Z_{nor}) - \bar{Q}_{in}dt) \quad (44)$$

Where $Z_{limit-op}$ is the limit operating water level within a specific reaction time, dt is a specific reaction time and $Z(V)$ is the water level at reservoir capacity V .

The specific reaction time cannot exceed the limited reaction time of the water level, that is:

$$dt \leq T_{limit-re} \quad (45)$$

The opening of each sluice needs to meet the following conditions:

$$K_{min} \leq K_i \leq K_{max}, i = 1, 2, \dots, n \quad (46)$$

$$|K_i - K_j| \leq K_{max-dis}, i \neq j \text{ and } i, j = 1, 2, \dots, n \quad (47)$$

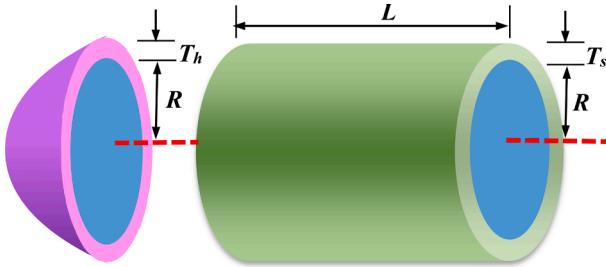
K_{min} and K_{max} are the minimum and maximum sluice openings, respectively, $K_{max-dis}$ is the maximum allowable distance between the sluice openings, and n is the number of sluices.

This study is based on the measured hydrological data of a hydropower station in 2019. The outlet structure of the hydropower station is a five-hole flood discharge sluice with a dead water level of 550.0 m and a normal water level of 554.0 m. Therefore, there are five sluices to control, which means that the dimension of the design variable of the problem is 5. The population size is set as 100, and the maximum number of iterations is set as 200.

Table 30

Comparison of the optimal results from various optimizers for wind farm layout problem in the y-axis.

Algorithms	Value of optimal variable							
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
EEFO	1099.313241	63.184939	328.901663	1092.390463	1934.037569	659.725709	1654.209839	674.272556
LFD	1952.098979	1950.989603	45.209726	170.964598	965.550439	955.165446	1209.786119	1876.801721
AOA	308.177899	130.032159	1435.017404	315.434717	1595.866209	1048.517654	706.448749	1132.994518
WOA	116.346234	1327.878147	899.119555	995.018523	1645.636018	1269.161822	386.935897	504.643504
SCA	1701.288485	811.140807	1003.428594	1498.218893	366.949834	1241.978081	114.616634	970.456972
HHO	1959.993470	1691.263879	41.517062	446.663164	1691.676173	506.696508	1030.988060	54.756090
BOA	1636.299285	1303.239098	1089.681878	186.077094	987.929388	1829.239125	1887.677703	364.132292
WDO	1748.690065	1960.000000	370.892510	124.093594	1109.780477	1952.322461	1420.212467	682.946457
CMA-ES	142.513729	1884.435692	1226.016774	1045.364652	1775.375278	81.611335	1163.456939	1473.953039
MFO	1053.238242	1257.230119	1951.553660	1198.951952	1725.509955	281.128232	1488.968462	292.900901
GSA	792.451087	465.235898	1200.356372	1090.730222	65.498618	725.696130	78.582404	527.104971
WDE	601.741532	1500.338781	1861.133047	1543.786862	1959.473428	350.010719	1759.666596	123.978995
ASO	1785.745408	1122.122633	163.382572	626.935902	839.170373	1398.871918	1424.061299	265.934944
Value of optimal variable								
y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}		
1109.799794	1413.562250	1373.090750	1667.019526	367.077210	44.378659	1959.888433	6033.929894	
1122.199109	1654.513867	612.381851	275.413584	268.321803	477.548705	1459.272122	5515.028012	
552.866723	1862.436106	127.557013	1805.602208	710.919988	1387.765469	392.015400	5089.287758	
1673.753028	47.967289	503.963499	1353.134510	1959.999922	1886.877360	346.571330	5422.624781	
1217.979964	1695.293964	1463.118985	311.541066	1931.561236	671.575108	1680.910575	5103.427557	
958.427291	1317.981590	41.852806	505.380265	855.799216	1956.834583	1267.246502	5800.778593	
797.630370	467.688818	1516.017634	591.748929	1093.256388	273.810127	203.434097	5197.129277	
1867.000630	101.163376	1095.573656	45.777434	647.288288	330.626066	384.192073	5633.798806	
1635.888642	667.108798	416.235717	1724.970222	720.449385	115.746847	932.321416	5111.589811	
1533.288410	629.058101	854.222077	1710.096197	67.496956	594.468055	1959.980440	5745.394216	
1867.446343	1420.868113	1228.853269	753.299514	42.727487	1751.640244	1438.639464	4832.737377	
1146.088287	714.809918	952.894488	349.163697	318.098462	1360.659616	1212.971081	5671.063524	
1816.436043	846.967143	1669.262655	1129.414516	213.124871	477.929854	995.081494	5355.807137	
Optimal output								

**Fig. 22.** Pressure vessel design.

For comparison, the enumeration method is also employed in this study. Fig. 35 depicts the change processes of the water level and the outflow under 161 combinations of different sluice openings. According to Fig. 35, the inflow fluctuates greatly, and no combination of openings can guarantee that the water level is controlled between 550 and 554 m throughout the period, and the outflow is not well matched with the

inflow. To better control the water level, a two-stage optimization process is performed by EEFO. Fig. 36 depicts the convergence curves of the two-stage optimization process for EEFO, and the corresponding combination of sluice openings for each stage is tabulated in Table 43.

From the convergence curves in Fig. 36, EEFO successfully minimizes the objective function during the iteration. It is notable that the solutions do not improve in some successive iterations. This is probably because EEFO fails to find a suitable solution for refinement to avoid local optima, and EEFO keeps searching the new regions to find promising solutions. However, EEFO successfully finds better solutions after each stagnation period of improvement. The change processes of the water level and the outflow for the proposed algorithm are depicted in Fig. 37. From Fig. 37, the sluice openings provided by EEFO can better control the change process of the water level in the reservoir. During the whole regulation period, the water level is almost kept close to the target level of 553 m, which makes the outflow match the inflow well. This study's results demonstrate that EEFO can find global optimum solutions in solving challenging problems with unknown search spaces.

Table 31

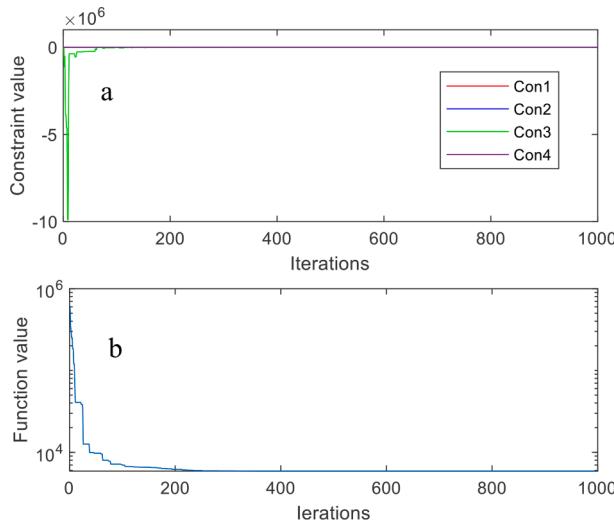
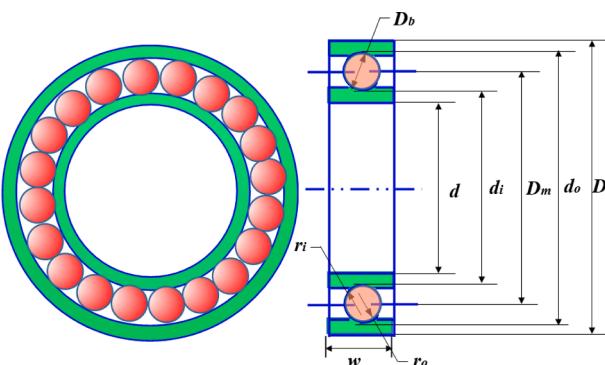
Results from various optimizers for pressure vessel design.

Algorithms	Best	Worst	AVE	STD
EEFO	5885.333900	5885.472291	5885.369464	0.040712
LFD	10016.841025	35256.530996	16563.070530	7519.849337
AOA	7617.934867	12756.105453	9593.178354	1622.645587
WOA	6015.665497	13334.814919	7779.891260	2164.523000
SCA	6464.149566	7832.351502	7118.167574	431.865122
HHO	6210.209898	19909.312798	8844.427559	4291.235784
BOA	7382.196851	8708.611787	8146.910875	544.302121
WDO	6442.559154	48376.624585	12256.128010	12765.923113
CMA-ES	8316.409071	36530.074641	14222.880985	9411.902468
MFO	5931.899493	6898.216484	6230.502772	294.416773
GSA	27556.193810	197936.126764	110720.429782	53548.761447
WDE	5991.424740	6182.773414	6083.698631	64.016551
ASO	6439.320685	10765.024859	8356.674805	1459.539479

Table 32

Comparison of the optimal results from various optimizers for pressure vessel design.

Algorithm	Optimal variable				Optimal cost
	T_s	T_h	R	L	
EEFO	0.778169146	0.384649393	40.319642897	199.999665793	5885.333900
LFD	1.490131	0.502673	52.691062	93.276330	10016.841025
AOA	1.258517	0.614482	63.392359	22.592781	7617.934867
WOA	0.845662	0.418074	43.779774	156.859610	6015.665497
SCA	0.944102	0.460714	46.879464	126.274344	6464.149566
HHO	0.853674	0.473137	44.202216	152.264261	6210.209898
BOA	1.068582	0.578485	49.651806	101.575829	7382.196851
WDO	0.840011	0.417463	41.250823	193.352351	6442.559154
CMA-ES	1.057067	0.637988	52.211268	107.328653	8316.409071
MFO	0.804510	0.397670	41.684437	181.835261	5931.899493
GSA	1.384803	1.862261	71.751464	113.580753	27556.193810
WDE	0.813755	0.399482	41.867113	180.086638	5991.424740
ASO	0.871612	0.430839	45.161269	146.977426	6439.320685

**Fig. 23.** (a) Constraint values versus iteration number, (b) convergence curve.**Fig. 24.** Rolling element bearing design.

6. Conclusions and future research

A novel population-based optimizer termed EEFO is proposed in this study for global optimization. The concept of EEFO originated from the social behaviors of electric eels in the foraging process in nature. Four foraging behaviors of electric eels are simulated to perform exploration

and exploitation when solving optimization problems, and an energy factor is designed to transition from exploration to exploitation. The efficiency of the EEFO algorithm is evaluated using the classical 23 benchmark functions containing unimodal and multimodal functions, the CEC2017 test suite, and the CEC2017 test suite with various dimensions. To highlight the optimization capacity of EEFO, its performance is compared with those of 12 well-designed algorithms such as LFD, AOA, WOA, SCA, HHO, BOA, WDO, CMA-ES, MFO, GSA, WDE and ASO. Meanwhile, several nonparametric statistics, including the Wilcoxon test, Friedman test and Nemenyi test, are employed to test the performance differences. The exceptional exploitation and exploration performances of EEFO are evidenced by its superior efficiency in handling both unimodal and multimodal functions, and the superior results of EEFO over its competitors for the hybrid and composite functions in the CEC2017 discover EEFO can effectively balance global search and local search. These achievements can be attributed to the implementation of both local and global search mechanisms, along with the utilization of the energy factor. To highlight the ability of EEFO to solve real-world problems, EEFO was utilized to solve both the CEC2011 test suite and 10 classic engineering problems. In these practical applications, EEFO exhibits remarkable performance by outperforming many other algorithms in most of the test problems. Eventually, EEFO is utilized to solve a newly emerged engineering problem, sluice opening control under a tripping accident of a hydropower station. The results show that EEFO is effective in obtaining a satisfactory solution.

Based on the above experimental results, it is demonstrated that EEFO is significantly successful compared to other optimization algorithms. This success is attributed to the following main advantages.

- EEFO has four different search strategies, including interacting, resting, hunting, and migrating. These four search strategies can be suitable for solving optimization problems with a wider range of distinctive features such as unimodal, numerous constraints and variables.
- This algorithm uses an energy factor to control the four search strategies. The energy factor adopts an adaptive change strategy, which can better balance exploration and exploitation.
- With the free-parameter feature of EEFO, there is no requirement to set a fixed parameters when solving optimization problems, increasing the ease and simplicity of the algorithm.

Based on the previous analysis, when compared to other methods, EEFO exhibits more pronounced characteristics, such as simplicity, ease of implementation, scalability, and robustness. These qualities also serve as strong motivations for researchers to adopt this algorithm. When

Table 33

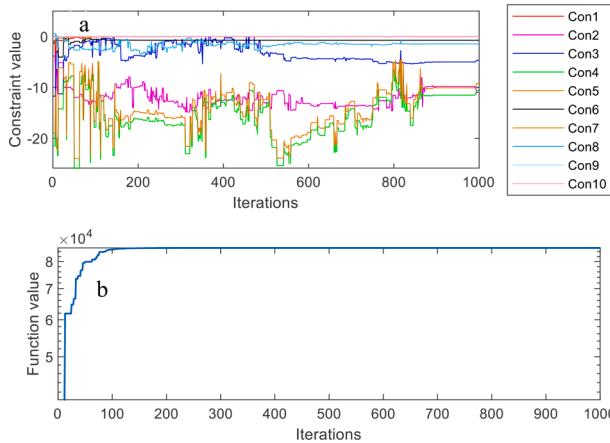
Results from various optimizers for rolling element bearing design.

Algorithm	Best	Worst	AVE	STD
EEFO	85549.239146	85548.502966	85549.165113	0.232656
LFD	75343.855393	50369.015525	62523.969548	8777.594019
AOA	82068.103227	55961.764456	64015.079165	8826.873610
WOA	83600.312249	72043.818857	76436.655320	3065.556860
SCA	70489.744125	46543.430024	59494.299569	6747.795258
HHO	77949.633183	68365.045919	73895.046096	2673.594943
BOA	73170.943185	46569.942559	56868.378187	8216.293273
WDO	85401.862172	84435.430706	84825.820427	449.264694
CMA-ES	78502.655717	55692.973285	70039.802001	8826.033206
MFO	85520.690361	76719.397526	84559.825040	2756.218217
GSA	82646.278409	57629.227168	70408.758519	7472.615817
WDE	85260.915307	84242.320329	84620.797810	306.540975
ASO	76439.452397	68391.953655	74041.540144	2332.199957

Table 34

Comparison of the optimal results from various optimizers for rolling element bearing design.

Algorithm	Optimal variable										Optimal load-carrying capacity
	D_m	D_b	Z	f_i	f_o	$K_{D\min}$	$K_{D\max}$	ε	e	ζ	
EEFO	125.719056	21.425590	10.685095	0.515000	0.515000	0.468500	0.662893	0.300000	0.039010	0.700284	85549.239146
LFD	126.684122	20.329399	11.248031	0.516332	0.530318	0.424146	0.600143	0.301630	0.027940	0.603589	75343.855393
AOA	125.752690	21.046363	10.631241	0.515377	0.546293	0.430523	0.613917	0.313173	0.068139	0.631418	82068.103227
WOA	125.242079	21.151878	10.906384	0.515000	0.586807	0.426069	0.642729	0.321533	0.073257	0.634876	83600.312249
SCA	125.058420	19.931449	11.021414	0.517653	0.599046	0.486988	0.665107	0.314217	0.073243	0.663733	70489.744125
HHO	125.426122	21.075631	10.206080	0.515011	0.517834	0.410758	0.614044	0.318339	0.039933	0.639890	77949.633183
BOA	125.482866	20.157267	10.968797	0.516914	0.545524	0.488465	0.606596	0.300510	0.029923	0.670112	73170.943185
WDO	125.715808	21.405998	10.991733	0.515000	0.557056	0.453010	0.680228	0.300158	0.100000	0.600000	85401.862172
CMA-ES	126.429182	20.427398	11.237624	0.515049	0.547276	0.438705	0.660806	0.305393	0.059206	0.652371	78502.655717
MFO	125.705104	21.421993	10.968576	0.515001	0.522269	0.430688	0.619033	0.300041	0.050351	0.622953	85520.690361
GSA	125.911194	21.112782	10.504103	0.515320	0.574349	0.455095	0.618050	0.304373	0.064810	0.663573	82646.278409
WDE	125.613968	21.392048	11.160646	0.515020	0.536101	0.486367	0.643472	0.303161	0.080177	0.637487	85260.915307
ASO	125.840913	20.111466	10.622865	0.515000	0.546088	0.444825	0.629972	0.312447	0.053554	0.637331	76439.452397

**Fig. 25.** (a) Constraint values versus iteration number, (b) convergence curve.

utilizing this algorithm, practitioners typically start by translating real-world problems into mathematical models, precisely defining the objective function, variable search space, and constraints. Subsequently, the algorithm is employed to optimize the objective function and obtain an approximate optimal solution. Finally, a verification is conducted to ascertain whether the problem requirements and constraints are satisfied, followed by an evaluation to determine the feasibility and effectiveness of the final solution. The algorithm incorporates new ideas and search strategies, which will inspire researchers to explore different ways of thinking and methods in problem-solving. It also provides new

insights for the development of other optimization techniques. The algorithm's high searching efficiency offers an alternative means to address a broader range of practical problems, providing additional possibilities for further application studies.

EEFO represents the standard version of the algorithm, and researchers prefer to improve this algorithm for the following reasons. First, the algorithm employs unique search strategies that are noticeably distinct from the search strategies of any other algorithm. Therefore, it is easy to combine this algorithm with other optimization algorithms or operators to develop hybrid or improved algorithms with significant enhancements. Second, due to the absence of additional parameters in this algorithm, one can better focus on improving the search strategy without considering the impact of parameter value variations on search performance. This makes it easier to apply the improved algorithm to a wider range of engineering problems. Finally, it has been demonstrated that this algorithm has good search capabilities for global optimal solutions, which will significantly contribute to the optimization performance of improved algorithms, such as convergence rate and accuracy of the optimal solutions.

Despite superior performance, EEFO also exhibits several limitations. One of the limitations of EEFO is that it is less powerful in handling certain discrete optimization problems. This may be due to the algorithm's lack of mechanisms for searching in discrete spaces. Another limitation is that it may not be possible for EEFO to achieve a quasi-optimal solution very close to the global optimum when solving certain optimization problems. This limitation, as stated by the NFL theory, is that no single algorithm can effectively solve all optimization problems. The limitations of EEFO, however, create opportunities for promising research directions and breakthroughs in the future. First,

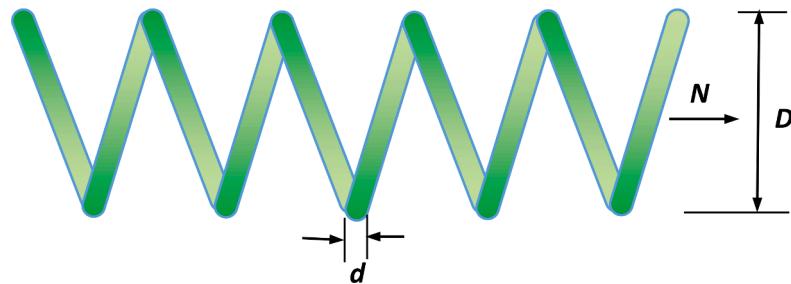


Fig. 26. Tension/compression string design.

Table 35

Results from various optimizers for tension/compression spring design.

Algorithm	Best	Worst	AVE	STD
EEFO	0.012666021	0.012682986	0.012667867	3.5619E-06
LFD	0.012732678	0.017826287	0.015168093	1.0208E-03
AOA	0.014225039	0.014182238	0.012920945	2.8315E-04
WOA	0.012953152	0.017775066	0.013802593	1.2579E-03
SCA	0.013038945	0.013214854	0.01307181	1.5046E-04
HHO	0.012812378	0.015530119	0.013477085	7.6379E-04
BOA	0.012708208	0.014063041	0.013277452	3.3381E-04
WDO	0.013064841	0.016179514	0.013872761	8.3987E-04
CMA-ES	0.012716482	0.023832583	0.017815147	2.4962E-03
MFO	0.012682681	0.01272871	0.012685507	1.7564E-05
GSA	0.015817941	0.027391053	0.018536971	3.2595E-03
WDE	0.012699349	0.013379716	0.013117915	1.4495E-04
ASO	0.014109773	0.020634996	0.016536195	1.7970E-03

Table 36

Comparison of the optimal results from various optimizers for tension/compression spring design.

Algorithm	Optimal variables			Optimal weigh
	d	D	N	
EEFO	0.05189732267	0.36174867132	10.93047450222	0.012666021
LFD	0.051947083	0.362956071	11.42207318	0.012732678
AOA	0.051471901	0.335577673	13.73100108	0.014225039
WOA	0.05576241	0.46286025	6.963042089	0.012953152
SCA	0.052212008	0.367924179	10.52358389	0.013038945
HHO	0.054574793	0.430175383	7.501398737	0.012812378
BOA	0.051247351	0.345631603	12.11774811	0.012708208
WDO	0.053942033	0.408184801	8.682074104	0.013064841
CMA-ES	0.052715164	0.381341981	10.06099289	0.012716482
MFO	0.052675125	0.380906753	10.1313273	0.012682681
GSA	0.055674002	0.425269107	10.2309389	0.015817941
WDE	0.051238875	0.345504924	11.80627308	0.012699349
ASO	0.054585868	0.430493805	8.577709363	0.014109773

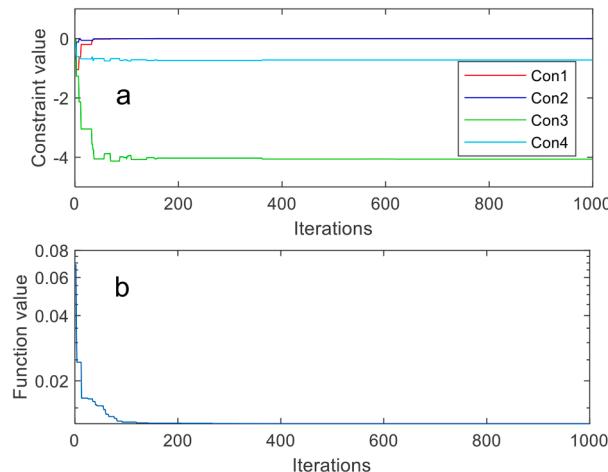
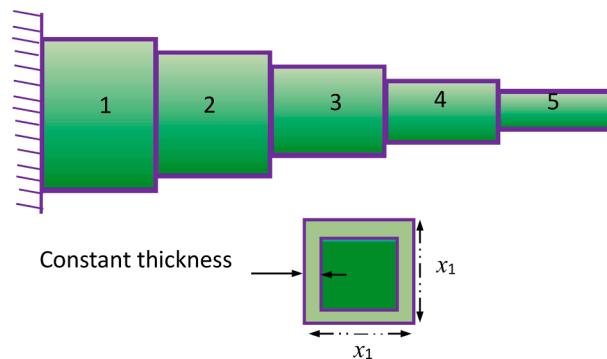


Fig. 27. (a) Constraint values versus iteration number, (b) convergence curve.

**Fig. 28.** Cantilever beam design.**Table 37**

Results from various optimizers for cantilever beam design.

Algorithm	Best	Worst	AVE	STD
EEFO	1.339957	1.339963	1.339959	1.9448E-06
LFD	1.420081	3.182097	2.296167	0.600313
AOA	1.363395	1.542881	1.448938	6.1979E-02
WOA	1.368453	1.643592	1.432271	8.3574E-02
SCA	1.375715	1.464466	1.417790	2.9772E-02
HHO	1.340146	1.379533	1.360438	1.2659E-02
BOA	1.347303	1.366022	1.356449	7.0475E-03
WDO	1.346289	1.469697	1.375004	3.8302E-02
CMA-ES	1.339996	2.534876	1.498325	0.384103
MFO	1.340047	1.341152	1.340539	4.0889E-04
GSA	1.339987	1.340684	1.340123	2.1228E-04
WDE	1.344258	1.351631	1.347502	2.2478E-03
ASO	1.4051472	1.340097	1.340011	4.2069E-05

Table 38

Comparison of the optimal results from various optimizers for cantilever beam design.

Algorithm	Optimal variables					Optimal weigh
	x_1	x_2	x_3	x_4	x_5	
EEFO	6.014990	5.309936	4.492868	3.502957	2.152917	1.339957
LFD	5.889716	5.658800	3.663844	4.722257	2.823086	1.420081
AOA	5.733315	5.577035	4.190541	3.678126	2.670267	1.363395
WOA	5.99574537	5.26291546	5.22369690	2.98631999	2.46165530	1.368453
SCA	5.384876	6.103360	5.055345	3.566477	1.936657	1.375715
HHO	5.949678	5.340489	4.528291	3.521419	2.136819	1.340146
BOA	6.115911	5.319722	4.673817	3.209578	2.272360	1.347303
WDO	6.191883	5.606316	4.305446	3.336054	2.135438	1.346289
CMA-ES	5.986304	5.308903	4.516318	3.506135	2.156636	1.339996
MFO	6.003523	5.359581	4.490266	3.488541	2.133203	1.340047
GSA	6.027831	5.313718	4.503818	3.494809	2.133973	1.339987
WDE	5.934061	5.169506	4.537754	3.500252	2.401019	1.344258
ASO	6.026975	5.300897	4.477998	3.514038	2.154028	1.4051472

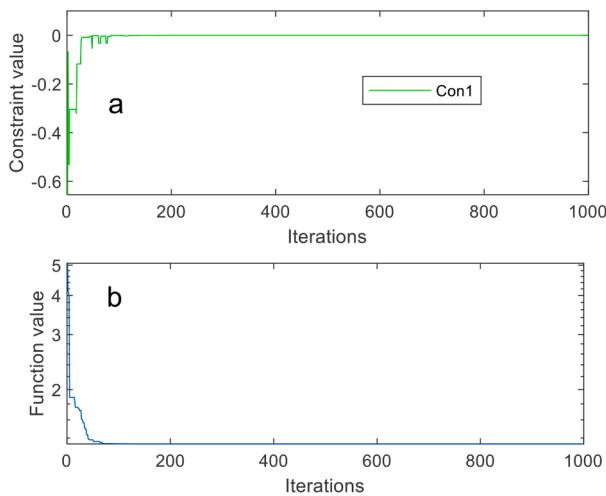


Fig. 29. (a) Constraint values versus iteration number, (b) convergence curve.

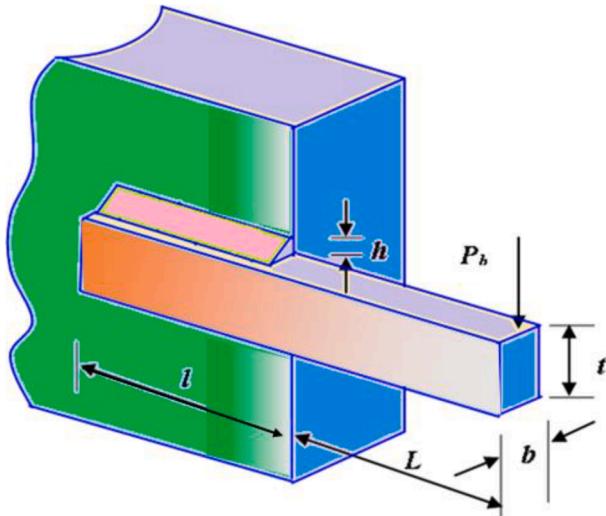


Fig. 30. Welded beam design.

Table 39

Comparison of the optimal results from various optimizers for welded beam design.

Algorithm	Best	Worst	AVE	STD
EEFO	1.724852	1.724852	1.724852	3.0183E-11
LFD	1.998036	4.049777	2.900662	7.9290E-01
AOA	1.905838	2.138004	2.013208	7.5147E-02
WOA	1.779498	2.508008	2.172739	2.4593E-01
SCA	1.854415	2.017647	1.917236	5.4287E-02
HHO	1.744796	2.022719	1.839953	9.4381E-02
BOA	1.818726	2.122507	1.947681	9.1187E-02
WDO	1.738676	2.306834	1.872404	2.0193E-01
CMA-ES	1.735277	1.839211	1.770322	3.3395E-02
MFO	1.724852	1.943198	1.758139	7.0197E-02
GSA	1.941851	2.854070	2.300042	3.2829E-01
WDE	1.754825	1.811350	1.783966	1.9929E-02
ASO	1.841145	2.252507	2.009475	1.3706E-01

some variants of EEFO can be developed by introducing additional effective operators such as selection, crossover, mutation, chaotic mapping and opposite-based learning. Second, some hybridized versions of EEFO can be designed by merging other excellent optimization

Table 40

Comparison of the optimal results from various optimizers for welded beam design.

Algorithm	Optimal variable				Optimal cost
	T_s	T_h	R	L	
EEFO	0.205730	3.470489	9.036624	0.205730	1.724852
LFD	0.220856	3.959110	8.222742	0.251205	1.998036
AOA	0.187063	4.219533	9.304197	0.213687	1.905838
WOA	0.189775	4.075827	9.041007	0.205708	1.779498
SCA	0.209721	3.354941	9.204843	0.220076	1.854415
HHO	0.203809	3.560268	8.977798	0.208503	1.744796
BOA	0.186378	4.260193	8.953971	0.210429	1.818726
WDO	0.205764	3.518282	9.062263	0.206098	1.738676
CMA-ES	0.204501	3.489564	9.064586	0.206376	1.735277
MFO	0.205730	3.470489	9.036624	0.205730	1.724852
GSA	0.203934	3.883961	8.197014	0.250033	1.941851
WDE	0.194452	3.735383	9.038365	0.207313	1.754825
ASO	0.162432	4.791408	8.926360	0.210844	1.841145

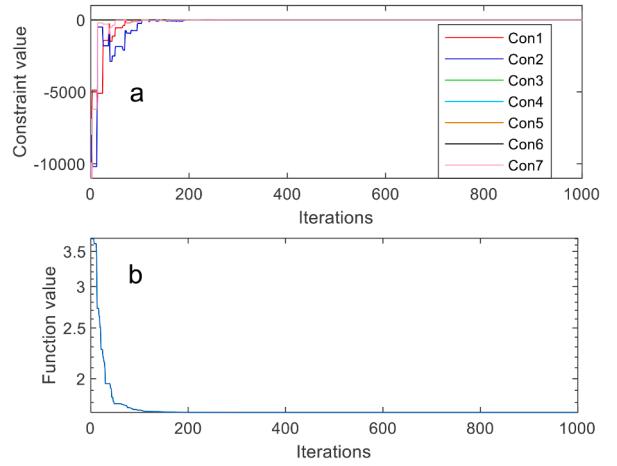


Fig. 31. (a) Constraint values versus iterations, (b) convergence curve.

techniques such as WOA, SCA, TLBO, and ABC. developing binary and multi-objective versions of EEFO for real-world engineering problems is an enticing research task. In addition, it is recommended that EEFO be utilized in various applications, including but not limited to feature selection, distribution of hydraulic units, machine learning, neural networks, fault diagnosis, financial investment, energy utilization, logistics networks, power systems, and image segmentation.

CRediT authorship contribution statement

Weiguo Zhao: Conceptualization, Writing – original draft, Investigation, Funding acquisition, Supervision. **Liying Wang:** Methodology, Writing – original draft, Visualization, Investigation, Funding acquisition, Supervision. **Zhenxing Zhang:** Writing – original draft, Visualization, Investigation, Writing – review & editing. **Honggang Fan:** Validation, Formal analysis, Software, Writing – review & editing. **Jiajie Zhang:** Writing – original draft, Investigation, Software, Visualization. **Seyedali Mirjalili:** Formal analysis, Writing – review & editing, Visualization. **Nima Khodadadi:** Formal analysis, Writing – review & editing. **Qingjiao Cao:** Software, Investigation.

Declaration of Competing Interest

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

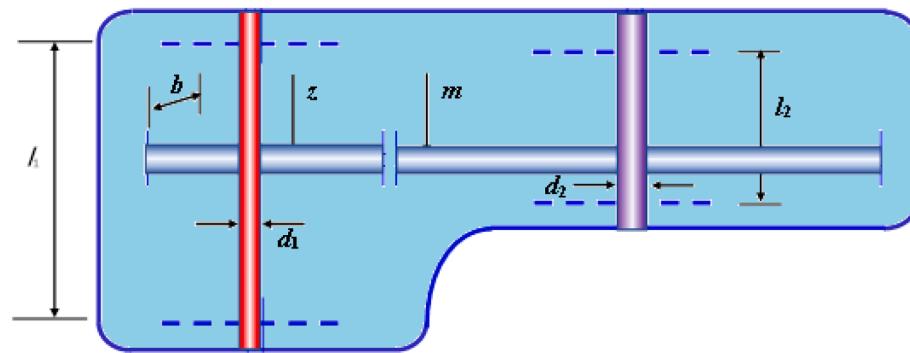


Fig. 32. Speed reducer design.

Table 41

Results from various optimizers for speed reducer design.

Algorithms	Best	Worst	AVE	STD
EEFO	2994.471066	2994.471066	2994.471066	5.2718E-07
LFD	3113.925845	3305.160506	3221.273272	63.802382
AOA	3104.061166	3327.153231	3198.219609	71.057482
WOA	3033.852272	3582.206085	3195.976157	163.585311
SCA	3135.213785	3303.749860	3224.034735	63.904411
HHO	3072.985309	3396.938093	3203.184150	108.055975
BOA	3099.767632	3422.405288	3232.823864	97.311800
WDO	3001.949212	3015.499497	3008.372154	4.232488
CMA-ES	3027.465123	3254.510152	3118.029182	72.909633
MFO	2994.735802	2997.982263	2995.640037	0.930716
GSA	3168.660844	4051.974823	3533.774190	290.333687
WDE	2997.539733	3003.301830	3000.222604	2.050121
ASO	3097.984744	3298.002721	3218.522204	66.488854

Table 42

Comparison of the optimal results from various optimizers for speed reducer design.

Algorithms	Optimal values for variables							Optimal weight
	b	m	z	l_1	L_2	d_1	d_2	
EEFO	3.5	0.7	17	7.3	7.71531991	3.35021467	5.28665446	2994.471066
LFD	3.548245	0.702458	17.180243	7.987841	8.002318	3.442205	5.318719	3113.925845
AOA	3.545006	0.702162	17.323320	7.600926	8.229364	3.372217	5.294455	3104.061166
WOA	3.513833	0.701574	17.000001	7.418384	7.972525	3.426935	5.286771	3033.852272
SCA	3.550023	0.707301	17.001157	7.962990	8.079567	3.493672	5.340753	3135.213785
HHO	3.501192	0.700003	17.053083	7.838362	7.995838	3.515435	5.307379	3072.985309
BOA	3.594572	0.700910	17.120789	7.660816	7.794487	3.412198	5.320026	3099.767632
WDO	3.501470	0.700056	17.000000	7.894414	7.753239	3.351411	5.287075	3001.949212
CMA-ES	3.519936	0.700306	17.037710	7.558829	7.898455	3.378864	5.292163	3027.465123
MFO	3.500203	0.700001	17.000069	7.300470	7.719896	3.350405	5.286676	2994.735802
GSA	3.585245	0.700247	17.082715	7.893968	8.186270	3.574860	5.360318	3168.660844
WDE	3.504042	0.700005	17.000965	7.339492	7.743218	3.350656	5.286996	2997.539733
ASO	3.537187	0.700000	17.323061	7.770147	7.994799	3.394492	5.302653	3097.984744

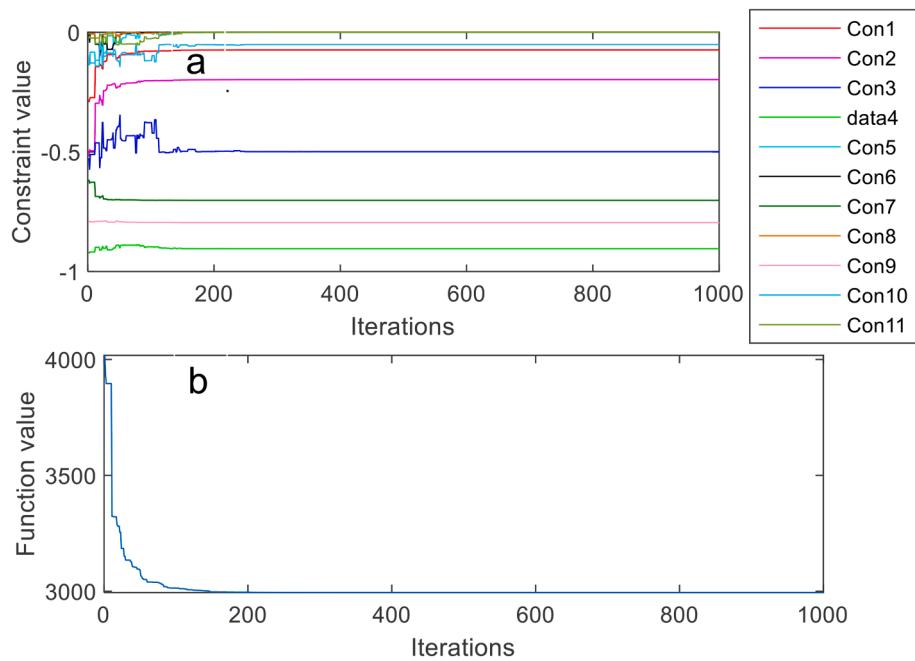


Fig. 33. (a) Constraint values versus iterations, (b) convergence curve.

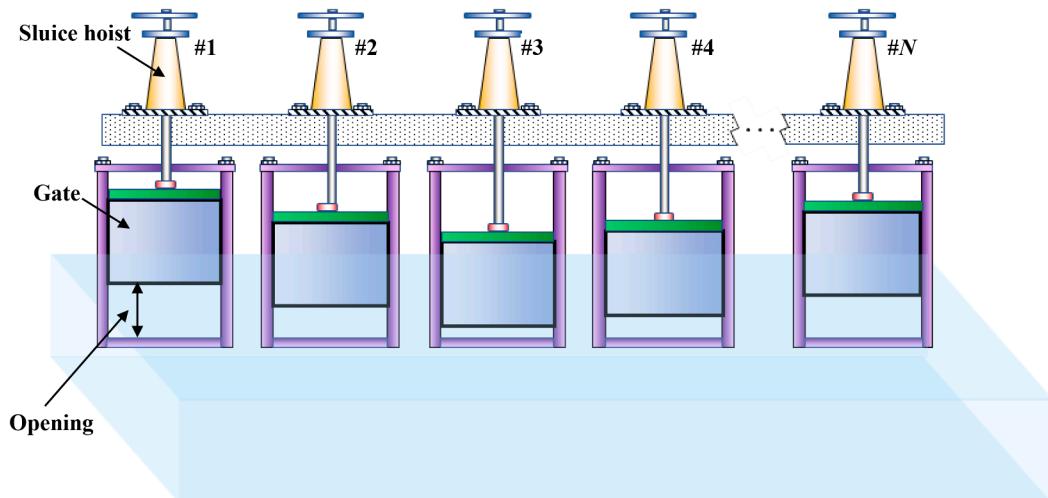


Fig. 34. Schematic of sluice opening control under accident tripping condition.

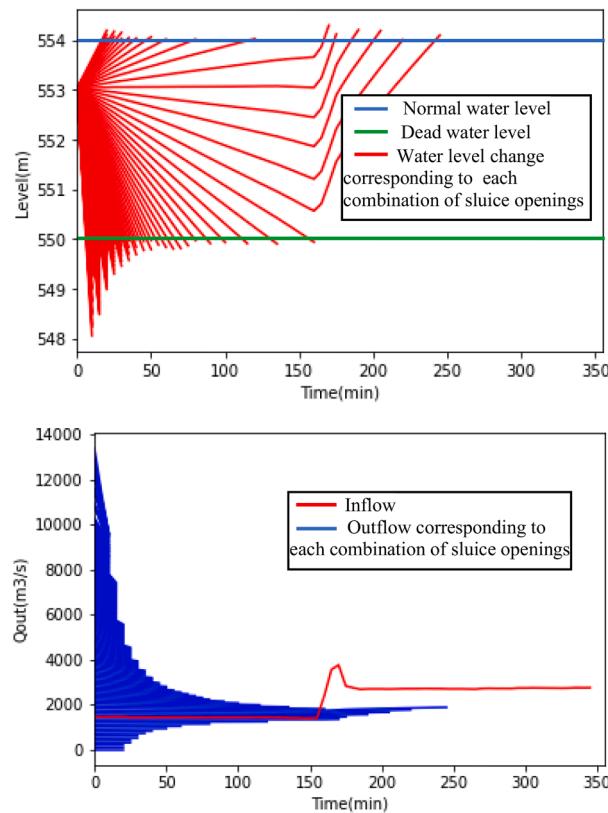


Fig. 35. Outflow change and water level change of the enumeration method.

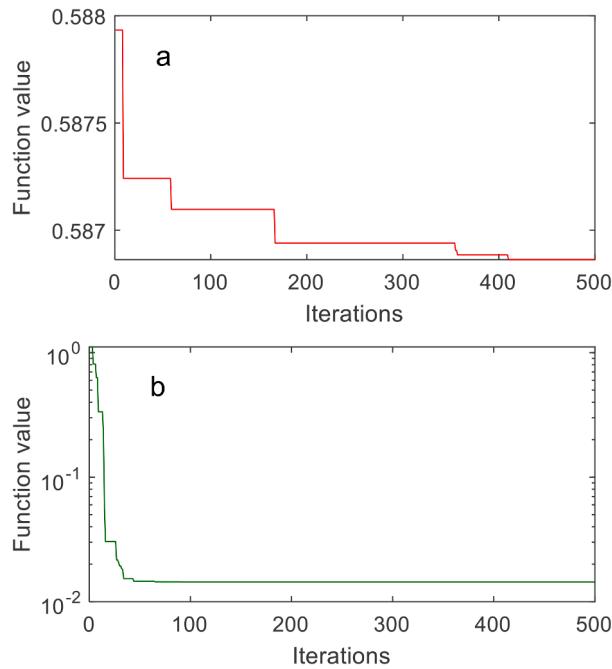
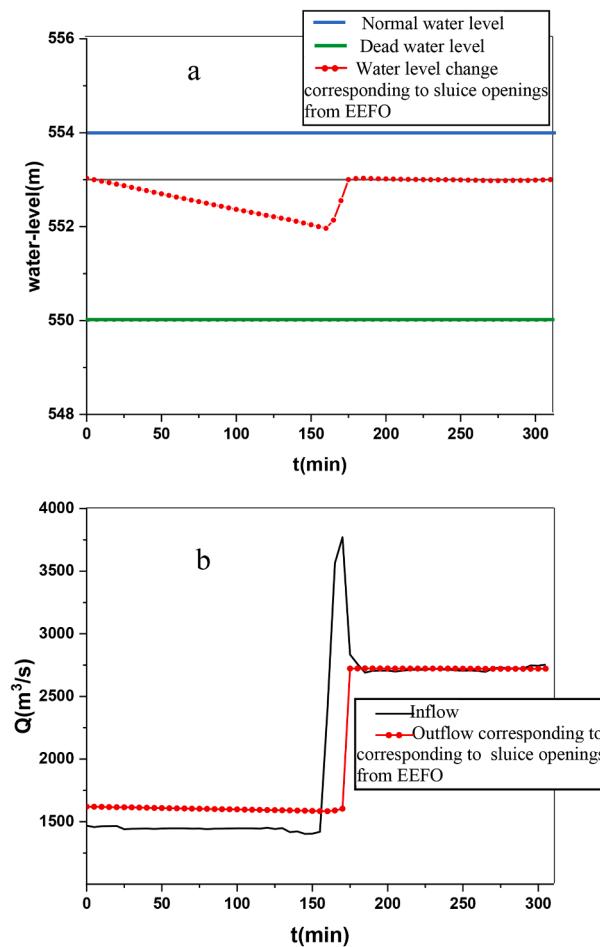


Fig. 36. Two-stage convergence curves of EEFO, (a) the 1st stage, (b) the 2nd stage.

Table 43

Two-stage sluice openings provided by EEFO.

	Sluice openings (m)					Function value
	Sluice #1	Sluice #2	Sluice #3	Sluice #4	Sluice #5	
The 1st stage	1.327444	1.547459	1.30873	1.575192	2.043191	0.586863
The 2nd stage	2.679449	2.722098	2.722626	2.743375	2.781806	0.014413

**Fig. 37.** (a) Outflow change, (b) water level change provided by EEFO.**Data availability**

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

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (11972144 and 12072098) and China Scholarship Council.

Appendix A

See Table A1.

Table A1

23 benchmark functions.

Name	Functions	d	Range	f_{opt}
Sphere	$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100] ^d	0
Schwefel 2.22	$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10] ^d	0
Schwefel 1.2	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100,100] ^d	0
Schwefel 2.21	$F_4(x) = \max_i\{ x_i , 1 \leq i \leq n\}$	30	[-100,100] ^d	0
Rosenbrock	$F_5(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i)^2) + (x_i - 1)^2$	30	[-30,30] ^d	0
Step	$F_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	[-100,100] ^d	0
Quartic	$F_7(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0, 1)$	30	[-1.28,1.28] ^d	0
Schwefel	$F_8(x) = -\sum_{i=1}^n (x_i \sin(\sqrt{ x_i }))$	30	[-500,500] ^d	-12569.5
Rastrigin	$F_9(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)^2$	30	[-5.12,5.12] ^d	0
Ackley	$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$	30	[-32,32] ^d	0
Griewank	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$	30	[-600,600] ^d	0
Penalized	$F_{12}(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)]$ $+ (y_n - 1)^2\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$	30	[-50,50] ^d	0
Penalized 2	$F_{13}(x) = 0.1 \{\sin^2(3\pi x_1) + \sum_{i=1}^{29} (x_i - 1)^2 p [1 + \sin^2(3\pi x_{i+1})]$ $+ (x_n - 1)^2 [1 + \sin^2(2\pi x_{30})]\} + \sum_{i=1}^{30} u(x_i, 5, 10, 4)$	30	[-50,50] ^d	0
Foxholes	$F_{14}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{j=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	2	[-65.536, 65.536] ^d	0.998
Kowalik	$F_{15}(x) = \sum_{i=1}^{11} \left a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right ^2$	4	[-5, 5] ^d	3.075×10^{-4}
Six Hump Camel	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5] ^d	-1.0316
Branin	$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	[-5, 10] \times [0, 15]	0.398
GoldStein–Price	$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)]$ $\times [30 + (2x_1 + 1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	[-2, 2] ^d	3
Hartman 3	$F_{19}(x) = -\sum_{i=1}^4 \exp\left[-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right]$	3	[0, 1] ^d	-3.86
Hartman 6	$F_{20}(x) = -\sum_{i=1}^4 \exp\left[-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right]$	6	[0, 1] ^d	-3.322
Shekel 5	$F_{21}(x) = -\sum_{i=1}^5 \left (x_i - a_i)(x_i - a_i)^T + c_i \right ^{-1}$	4	[0, 10] ^d	-10.1532
Shekel 7	$F_{22}(x) = -\sum_{i=1}^7 \left (x_i - a_i)(x_i - a_i)^T + c_i \right ^{-1}$	4	[0, 10] ^d	-10.4028
Shekel 10	$F_{23}(x) = -\sum_{i=1}^{10} \left (x_i - a_i)(x_i - a_i)^T + c_i \right ^{-1}$	4	[0, 10] ^d	-10.5364

References

- Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). *Metaheuristic algorithms: A comprehensive review* (pp. 185–231). Elsevier Inc..
- Abdel-Basset, M., El-Shahat, D., Jameel, M., & Abouhawwash, M. (2023). Young's double-slit experiment optimizer: A novel metaheuristic optimization algorithm for global and constraint optimization problems. *Computer Methods in Applied Mechanics and Engineering*, 403, Article 115652. <https://doi.org/10.1016/j.cma.2022.115652>
- Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376, Article 113609. <https://doi.org/10.1016/j.cma.2020.113609>
- Agushaka, J. O., & Ezugwu, A. E. (2021). Evaluation of several initialization methods on arithmetic optimization algorithm performance. *Journal of Intelligent Systems*, 31(1), 70–94. <https://doi.org/10.1515/jisys-2021-0164>
- Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2022a). Dwarf mongoose optimization algorithm. *Computer methods in applied mechanics and engineering*, 391, Article 114570. <https://doi.org/10.1016/j.cma.2022.114570>
- Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2022b). Gazelle Optimization Algorithm: A novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, 1–33. <https://doi.org/10.1007/s00521-022-07854-6>
- Ahmadi, S. A. (2017). Human behavior-based optimization: A novel metaheuristic approach to solve complex optimization problems. *Neural Computing and Applications*, 28(1), 233–244. <https://doi.org/10.1007/s00521-016-2334-4>
- Akyol, S., & Alatas, B. (2017). Plant intelligence based metaheuristic optimization algorithms. *Artificial Intelligence Review*, 47, 417–462. <https://doi.org/10.1007/s10462-016-9486-6>
- Alatas, B. (2011). ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Systems with Applications*, 38(10), 13170–13180. <https://doi.org/10.1016/j.eswa.2011.04.126>
- Alatas, B., & Bingol, H. (2020). Comparative Assessment of Light-Based Intelligent Search and Optimization Algorithms. *Light & Engineering*, 28(6), 51–59. <https://doi.org/10.33383/2019-029>
- Alba, E., & Dorronsoro, B. (2005). The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE transactions on evolutionary computation*, 9(2), 126–142. <https://doi.org/10.1109/TEVC.2005.843751>
- Ali, H., Das, S., & Shaikh, A. A. (2023). Investigate an imperfect green production system considering rework policy via Teaching-Learning-Based Optimizer algorithm. *Expert Systems with Applications*, 214, Article 119143. <https://doi.org/10.1016/j.eswa.2022.119143>
- Altinoz, O. T., & Yilmaz, A. E. (2019). Multiobjective Hooke-Jeeves algorithm with a stochastic Newton-Raphson-like step-size method. *Expert Systems with Applications*, 117, 166–175. <https://doi.org/10.1016/j.eswa.2018.09.033>
- Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: A novel approach for global optimization. *Soft Computing*, 23(3), 715–734. <https://doi.org/10.1007/s00500-018-3102-4>
- Ashrafi, S. M., & Dariane, A. B. (2011). A novel and effective algorithm for numerical optimization: Melody search (MS). In *In 2011 11th international conference on hybrid intelligent systems (HIS)* (pp. 109–114). IEEE. <https://doi.org/10.1109/HIS.2011.6122089>
- Askari, Q., Saeed, M., & Younas, I. (2020). Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Systems with Applications*, 161, Article 113702. <https://doi.org/10.1016/j.eswa.2020.113702>
- Askari, Q., Younas, I., & Saeed, M. (2020). Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-based systems*, 195, Article 105709. <https://doi.org/10.1016/j.knosys.2020.105709>
- Atashpaz-Gargari, E., & Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *In 2007 IEEE congress on evolutionary computation* (pp. 4661–4667). IEEE. <https://doi.org/10.1109/CEC.2007.4425083>

- Awad, N. H., Ali, M. Z. & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems. In *2017 IEEE congress on evolutionary computation (cec)* (pp. 372–379), Donostia-San Sebastián, Spain.
- Ayyarao, T. S., Ramakrishna, N. S. S., Elavarasan, R. M., Polumahanthi, N., Rambabu, M., Saini, G., ... Alatas, B. (2022). War strategy optimization algorithm: A new effective metaheuristic algorithm for global optimization. *IEEE Access*, 10, 25073–25105. <https://doi.org/10.1109/ACCESS.2022.3153493>
- Azizi, M., Aickelin, U., A Khorshidi, H., & Baghazadeh Shishehgarkhaneh, M. (2023). Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Scientific Reports*, 13(1), 1–23. <https://doi.org/10.1038/s41598-022-27344-y>.
- Barik, S., & Das, D. (2021). Athletic Run Based Optimization: A novel method for the integration of DGs and shunt capacitors. In *In 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)* (pp. 1–6). IEEE. <https://doi.org/10.1109/CONECCT52877.2021.9622704>.
- Bastos, D. A., Zuanon, J., Rapp Py-Daniel, L., & de Santana, C. D. (2021). Social predation in electric eels. *Ecology and evolution*, 11(3), 1088–1092. <https://doi.org/10.1002/ece3.7121>
- Bayraktar, Z., Komurcu, M., Bossard, J. A., & Werner, D. H. (2013). The wind driven optimization technique and its application in electromagnetics. *IEEE transactions on antennas and propagation*, 61(5), 2745–2757. <https://doi.org/10.1109/TAP.2013.2238654>
- Beyer, H. G., & Schwefel, H. P. (2002). *Evolution strategies-a comprehensive introduction*. *Natural computing*, 1, 3–52. <https://doi.org/10.1023/A:1015059928466>
- Blocho, M. (2020). Heuristics, metaheuristics, and hyperheuristics for rich vehicle routing problems. In *Smart Delivery Systems* (pp. 101–156). Elsevier. <https://doi.org/10.1016/B978-0-12-815715-2.00009-9>
- Borji, A., & Hamidi, M. (2009). A new approach to global optimization motivated by parliamentary political competitions. *International Journal of Innovative Computing, Information and Control*, 5(6), 1643–1653.
- Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, 243, Article 108457. <https://doi.org/10.1016/j.knosys.2022.114616>
- Braik, M., Sheta, A., & Al-Hiary, H. (2021). A novel meta-heuristic search algorithm for solving optimization problems: Capuchin search algorithm. *Neural computing and applications*, 33(7), 2515–2547. <https://doi.org/10.1007/s00521-020-05145-6>
- Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005, August). Learning to rank using gradient descent. Proceedings of the 22nd international conference on Machine learning, Bonn, Germany.
- Castro, L. D., & Timmis, J. I. (2003). Artificial immune systems as a novel soft computing paradigm. *Soft computing*, 7, 526–544. <https://doi.org/10.1109/ICEC.1998.700092>
- Catania, K. C. (2015a). Electric eels concentrate their electric field to induce involuntary fatigue in struggling prey. *Current Biology*, 25(22), 2889–2898. <https://doi.org/10.1016/j.cub.2015.09.036>
- Catania, K. C. (2015b). Electric eels use high-voltage to track fast-moving prey. *Nature Communications*, 6(1), 8638. <https://doi.org/10.1038/ncomms9638>
- Catania, K. C. (2019). The astonishing behavior of electric eels. *Frontiers in Integrative Neuroscience*, 13, 23. <https://doi.org/10.3389/fnint.2019.00023>
- Chickermane, H., & Gea, H. C. (1996). Structural optimization using a new local approximation method. *International journal for numerical methods in engineering*, 39 (5), 829–846. [https://doi.org/10.1002/\(SICI\)1097-0207\(19960315\)39:5<829::AID-NME884>3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-0207(19960315)39:5<829::AID-NME884>3.0.CO;2-U)
- Civicioglu, P., Besirkov, E., Gunen, M. A., & Atasever, U. H. (2020). Weighted differential evolution algorithm for numerical function optimization: A comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms. *Neural Computing and Applications*, 32, 3923–3937. <https://doi.org/10.1007/s00521-018-3822-5>
- Coello Coello, C. A., & Pulido, G. T. (2005). Multiobjective structural optimization using a microgenetic algorithm. *Structural and Multidisciplinary Optimization*, 30, 388–403. <https://doi.org/10.1007/s00158-005-0527-z>
- Coello, C. A. C., & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3), 193–203. [https://doi.org/10.1016/S1474-0346\(02\)00011-3](https://doi.org/10.1016/S1474-0346(02)00011-3)
- Comert, S. E., & Yazgan, H. R. (2023). A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 123, Article 106375. <https://doi.org/10.1016/j.engappai.2023.106375>
- Corno, F., Reorda, M. S., & Squillero, G. (1998). A new evolutionary algorithm inspired by the selfish gene theory. In *In 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence* (pp. 575–580). IEEE. <https://doi.org/10.1109/ICEC.1998.700092>
- Das, B., Mukherjee, V., & Das, D. (2020). Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems. *Advances in Engineering software*, 146, Article 102804. <https://doi.org/10.1016/j.advengsoft.2020.102804>
- Das, S., & Suganthan, P. N. (2010). *Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems* (pp. 341–359). Jadavpur University: Nanyang Technological University, Kolkata.
- Dash, P., Saikia, L. C., & Sinha, N. (2015). Automatic generation control of multi area thermal system using Bat algorithm optimized PD-PID cascade controller. *International Journal of Electrical Power & Energy Systems*, 68, 364–372. <https://doi.org/10.1016/j.ijepes.2014.12.063>
- Dehghani, M., Montazeri, Z., Trojovská, E., & Trojovský, P. (2023). Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowledge-Based Systems*, 259, Article 110011. <https://doi.org/10.1016/j.knosys.2022.110011>
- Dehghani, M., Trojovská, E., & Zuščák, T. (2022). A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training. *Scientific Reports*, 12(1), 1–24. <https://doi.org/10.1038/s41598-022-22458-9>
- Derrac, J., García, S., Hui, S., Suganthan, P. N., & Herrera, F. (2014). Analyzing convergence performance of evolutionary algorithms: A statistical approach. *Information Sciences*, 289, 41–58. <https://doi.org/10.1016/j.ins.2014.06.009>
- Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3–18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- Diop, L., Samadianfar, S., Bodian, A., Yaseen, Z. M., Ghorbani, M. A., & Salimi, H. (2020). Annual rainfall forecasting using hybrid artificial intelligence model: Integration of multilayer perceptron with whale optimization algorithm. *Water Resources Management*, 34, 733–746. <https://doi.org/10.1007/s11269-019-02473-8>
- Doğan, B., & Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information sciences*, 293, 125–145. <https://doi.org/10.1016/j.ins.2014.08.053>
- Dorigo, M., & Stützle, T. (2019). *Ant colony optimization: Overview and recent advances* (pp. 311–351). Springer International Publishing.
- Erol, O. K., & Eksin, I. (2006). A new optimization method: Big bang–big crunch. *Advances in engineering software*, 37(2), 106–111. <https://doi.org/10.1016/j.advengsoft.2005.04.005>
- Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm-A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110, 151–166. <https://doi.org/10.1016/j.compstruc.2012.07.010>
- Eusuff, M. M., & Lansey, K. E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources planning and management*, 129(3), 210–225. [https://doi.org/10.1061/\(ASCE\)0733-9496\(2003\)129:3\(210\)](https://doi.org/10.1061/(ASCE)0733-9496(2003)129:3(210))
- Everett, H., III (1963). Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, 11(3), 399–417. <https://doi.org/10.1287/opre.11.3.399>
- Faramarzi, A., Heidarnejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert systems with applications*, 152, Article 113377. <https://doi.org/10.1016/j.eswa.2020.113377>
- Faramarzi, A., Heidarnejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191, Article 105190. <https://doi.org/10.1016/j.knosys.2019.105190>
- Fiza, S., Kumar, A. K., Devi, V. S., Kumar, C. N., & Kubra, A. (2023). Improved chimp optimization algorithm (ICOA) feature selection and deep neural network framework for internet of things (IOT) based android malware detection. *Measurement: Sensors*, 100785. <https://doi.org/10.1016/j.measen.2023.100785>
- Formato, R. A. (2009). Central force optimization: A new deterministic gradient-like optimization metaheuristic. *Opsearch*, 46(1), 25–51. <https://doi.org/10.1007/s12597-009-0003-4>
- Foroughi, S., Hamidi, J. K., Monjezi, M., & Nehring, M. (2019). The integrated optimization of underground stope layout designing and production scheduling incorporating a non-dominated sorting genetic algorithm (NSGA-II). *Resources Policy*, 63, Article 101408. <https://doi.org/10.1016/j.resourpol.2019.101408>
- Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200), 675–701. <https://doi.org/10.1080/01621459.1937.10503522>
- Galántai, A. (2000). The theory of Newton's method. *Journal of Computational and Applied Mathematics*, 124(1–2), 25–44. [https://doi.org/10.1016/S0377-0427\(00\)00435-0](https://doi.org/10.1016/S0377-0427(00)00435-0)
- Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *SIMULATION*, 76(2), 60–68. <https://doi.org/10.1177/003754970107600201>
- Ghasemian, H., Ghasemian, F., & Vahdat-Nejad, H. (2020). Human urbanization algorithm: A novel metaheuristic approach. *Mathematics and Computers in Simulation*, 178, 1–15. <https://doi.org/10.1016/j.matcom.2020.05.023>
- Givi, H., & Hubalovska, M. (2023). Skill Optimization Algorithm: A New Human-Based Metaheuristic Technique. *Computers, Materials & Continua*, 74(1), 179–202. <https://doi.org/10.32604/cmc.2023.030379>
- Goodarzimehr, V., Talatohari, S., Shojaee, S., & Hamzehei-Javaran, S. (2023). Special relativity search for applied mechanics and engineering. *Computer Methods in Applied Mechanics and Engineering*, 403, Article 115734. <https://doi.org/10.1016/j.cma.2022.115734>
- Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1), 1–18. <https://doi.org/10.1162/106365603321828970>
- Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. (2022). Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, 192, 84–110. <https://doi.org/10.1016/j.matcom.2021.08.013>
- Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222, 175–184. <https://doi.org/10.1016/j.ins.2012.08.023>
- He, P., Almasifar, N., Mehboodiya, A., Javaheri, D., & Webber, J. L. (2022). Towards green smart cities using internet of things and optimization algorithms: A systematic and bibliometric review. *Sustainable Computing: Informatics and Systems*, 36, Article 100822. <https://doi.org/10.1016/j.suscom.2022.100822>

- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97, 849–872. <https://doi.org/10.1016/j.future.2019.02.028>
- Hosseini, H. (2022). Implementation of heuristic search algorithms in the calibration of a river hydraulic model. *Environmental Modelling & Software*, 157, Article 105537. <https://doi.org/10.1016/j.envsoft.2022.105537>
- Houssein, E. H., Oliva, D., Çelik, E., Emam, M. M., & Ghoniem, R. M. (2023). Boosted sooty tern optimization algorithm for global optimization and feature selection. *Expert Systems with Applications*, 213, Article 119015. <https://doi.org/10.1016/j.eswa.2022.119015>
- Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. (2020). Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 94, Article 103731. <https://doi.org/10.1016/j.engappai.2020.103731>
- Hu, G., Du, B., Wang, X., & Wei, G. (2022a). An enhanced black widow optimization algorithm for feature selection. *Knowledge-Based Systems*, 235, Article 107638. <https://doi.org/10.1016/j.knosys.2021.107638>
- Hu, G., Zhong, J., & Wei, G. (2023a). SaCHBA_PDN: Modified honey badger algorithm with multi-strategy for UAV path planning. *Expert Systems with Applications*, 223, Article 119941. <https://doi.org/10.1016/j.eswa.2023.119941>
- Hu, G., Zhong, J., Du, B., & Wei, G. (2022b). An enhanced hybrid arithmetic optimization algorithm for engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 394, Article 114901. <https://doi.org/10.1016/j.cma.2022.114901>
- Hu, G., Zhong, J., Wei, G., & Chang, C. T. (2023b). DTCSMO: An efficient hybrid starling murmurization optimizer for engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 405, Article 115878. <https://doi.org/10.1016/j.cma.2023.115878>
- Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2018). On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications*, 31(11), 7665–7683. <https://doi.org/10.1007/s00521-018-3592-0>
- IEA. (2021). Hydropower Special Market Report. Retrieved from <https://www.iea.org/reports/hydropower-special-market-report>, Accessed January 10, 2022.
- Kannan, B. K., & Kramer, S. N. (1993, September). An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 97690, pp. 103–112), Portland, Oregon, USA.
- Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1), 108–132. <https://doi.org/10.1016/j.amc.2009.03.090>
- Kashan, A. H. (2009). League championship algorithm: A new algorithm for numerical function optimization. In *In 2009 international conference of soft computing and pattern recognition* (pp. 43–48). IEEE. <https://doi.org/10.1109/SoCPaR.2009.21>
- Kashan, A. H. (2015). A new metaheuristic for optimization: optics inspired optimization (OIO). *Computers & Operations Research*, 55, 99–25. <https://doi.org/10.1016/j.cor.2014.10.011>
- Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, 90, Article 103541. <https://doi.org/10.1016/j.engappai.2020.103541>
- Kaveh, A., & Bakhshpoori, T. (2016). Water evaporation optimization: A novel physically inspired optimization algorithm. *Computers & Structures*, 167, 69–85. <https://doi.org/10.1016/j.compstruc.2016.01.008>
- Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Advances in engineering software*, 110, 69–84. <https://doi.org/10.1016/j.advengsoft.2017.03.014>
- Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization method for optimum discrete design of truss structures. *Computers & Structures*, 139, 43–53. <https://doi.org/10.1016/j.compstruc.2014.04.006>
- Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta Mech.*, 213(3–4), 267–289. <https://doi.org/10.1007/s00707-009-0270-4>
- Kaveh, A., & Zolghadr, A. (2016). A novel meta-heuristic algorithm: Tug of war optimization. *International Journal of Civil Engineering*, 6(4), 469–492.
- Kaveh, A., Talatahari, S., & Khodadadi, N. (2020). Stochastic paint optimizer: Theory and application in civil engineering. *Engineering with Computers*, 1–32. <https://doi.org/10.1007/s00366-020-01179-5>
- Kumar, A., Wu, G. H., Ali, M. Z., Mallipeddi, R., Suganthan, P. N., & Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, 56, Article 100693. <https://doi.org/10.1016/j.swevo.2020.100693>
- Kumar, K., & Saini, R. P. (2022). A review on operation and maintenance of hydropower plants. *Sustainable Energy Technologies and Assessments*, 49, Article 101704. <https://doi.org/10.1016/j.seta.2021.101704>
- Kumar, N., Shaikh, A. A., Mahato, S. K., & Bhunia, A. K. (2021). Applications of new hybrid algorithm based on advanced cuckoo search and adaptive Gaussian quantum behaved particle swarm optimization in solving ordinary differential equations. *Expert Systems with Applications*, 172, Article 114646. <https://doi.org/10.1016/j.eswa.2021.114646>
- Kuo, R. J., Luthfiansyah, M. F., Masruroh, N. A., & Zulvia, F. E. (2023). Application of Improved Multi-Objective Particle Swarm Optimization Algorithm to Solve Disruption for the Two-Stage Vehicle Routing Problem with Time Windows. *Expert Systems with Applications*, 120009. <https://doi.org/10.1016/j.eswa.2023.120009>
- Lam, A., & Li, V. O. (2012). Chemical reaction optimization: A tutorial. *Memetic Computing*, 4(1), 3–17. <https://doi.org/10.1007/s12293-012-0075-1>
- Lynn, N., & Suganthan, P. N. (2015). Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm and Evolutionary Computation*, 24, 11–24. <https://doi.org/10.1016/j.swevo.2015.05.002>
- Ma, T., Zhu, Z., Wang, L., Wang, H., & Ma, L. (2022). Anomaly detection for hydropower turbine based on variational modal decomposition and hierarchical temporal memory. *Energy Reports*, 8, 1546–1551. <https://doi.org/10.1016/j.egyr.2022.02.286>
- Mahalakshmi, M., Hariharan, G., & Brindha, G. R. (2020). An efficient wavelet-based optimization algorithm for the solutions of reaction-diffusion equations in biomedicine. *Computer Methods and Programs in Biomedicine*, 186, Article 105218. <https://doi.org/10.1016/j.cmpb.2019.105218>
- Melani, A. H., Silva, J. M., de Souza, G. F., & Silva, J. R. (2016). Fault diagnosis based on Petri Nets: The case study of a hydropower plant. *IFAC-PapersOnLine*, 49(31), 1–6. <https://doi.org/10.1016/j.ifacol.2016.12.152>
- Meng, Z., Li, G., Wang, X., Sait, S. M., & Yıldız, A. R. (2021). A comparative study of metaheuristic algorithms for reliability-based design optimization problems. *Archives of Computational Methods in Engineering*, 28, 1853–1869. <https://doi.org/10.1007/s11831-020-09443-z>
- MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems*, 213, Article 106711. <https://doi.org/10.1016/j.knosys.2020.106711>
- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89, 228–249. <https://doi.org/10.1016/j.knosys.2015.07.006>
- Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96, 120–133. <https://doi.org/10.1016/j.knosys.2015.12.022>
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51–67. <https://doi.org/10.1016/j.amc.2009.03.090>
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 114, 163–191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Moein, S., & Logeswaran, R. (2014). KGMO: A swarm optimization algorithm based on the kinetic energy of gas molecules. *Information Sciences*, 275, 127–144. <https://doi.org/10.1016/j.ins.2014.02.026>
- Mohammed, H., & Rashid, T. (2020). A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. *Neural Computing and Applications*, 32(18), 14701–14718. <https://doi.org/10.1007/s00521-020-04823-9>
- Mousavi, S. H. S., & Bardsiri, V. K. (2019). Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Engineering Applications of Artificial Intelligence*, 86, 165–181. <https://doi.org/10.1016/j.engappai.2019.08.025>
- Moosavian, N., & Roodzari, B. K. (2014). Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm and Evolutionary Computation*, 17, 14–24. <https://doi.org/10.1016/j.swevo.2014.02.002>
- Motevali, M. M., Shanghooshabad, A. M., Aram, R. Z., & Keshavarz, H. (2019). WHO: A new evolutionary algorithm bio-inspired by wildebeests with a case study on bank customer segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(05), 1959017. <https://doi.org/10.3390/math10030351>
- Mousavirad, S. J., & Ebrahimpour-Komleh, H. (2017). Human mental search: A new population-based metaheuristic optimization algorithm. *Applied Intelligence*, 47(3), 850–887. <https://doi.org/10.1007/s10489-017-0903-6>
- National Geographic. (2022). Electric Eel. Retrieved from <https://www.nationalgeographic.com/animals/fish/facts/electric-eel>. Accessed January 10, 2022.
- Nationalzoo. (2022). Retrieved from <https://nationalzoo.si.edu/animals/electric-eel>. Accessed January 20, 2022.
- Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. Princeton University.
- Oszyczka, A., & Krenich, S. (2004). Some methods for multicriteria design optimization using evolutionary algorithms. *Journal of theoretical and applied mechanics*, 42(3), 565–584.
- Pant, M., Thangaraj, R., & Singh, V. P. (2009). Optimization of mechanical design problems using improved differential evolution algorithm. *International Journal of Recent Trends in Engineering*, 1(5), 21.
- Pereira, J. L. J., Francisco, M. B., Diniz, C. A., Oliver, G. A., Cunha, S. S., Jr, & Gomes, G. F. (2021). Lichtenberg algorithm: A novel hybrid physics-based metaheuristic for global optimization. *Expert Systems with Applications*, 170, Article 114522. <https://doi.org/10.1016/j.eswa.2020.114522>
- Price, K. (1996). Differential evolution: A fast and simple numerical optimizer. *Proceedings of North American fuzzy information processing*, Berkeley, CA.. <https://doi.org/10.1109/NAFIPS.1996.534790>
- Qais, M. H., Hasanian, H. M., Turky, R. A., Alghuwainem, S., Tostado-Vélez, M., & Jurado, F. (2022). Circle Search Algorithm: A Geometry-Based Metaheuristic Optimization Algorithm. *Mathematics*, 10(10), 1626. <https://doi.org/10.3390/math10101626>
- Rao, B. R., & Tiwari, R. (2007). Optimum design of rolling element bearings using genetic algorithms. *Mechanism and machine theory*, 42(2), 233–250. <https://doi.org/10.1016/j.mechmachtheory.2006.02.004>
- Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-aided design*, 43(3), 303–315. <https://doi.org/10.1016/j.cad.2010.12.015>
- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information sciences*, 179(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>

- Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, 7(4), 386–396. <https://doi.org/10.1109/TEVC.2003.814902>
- Razmjooy, N., Khalilpour, M., & Ramezani, M. (2016). A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: Theory and its application in PID designing for AVR system. *Journal of Control, Automation and Electrical Systems*, 27, 419–440. <https://doi.org/10.1007/s40313-016-0242-6>
- Ryan, C., Collins, J. J., & Neill, M. O. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In *Genetic Programming: First European Workshop, EuroGP'98* Paris, France, April 14–15, 1998 Proceedings 1 (pp. 83–96). Springer Berlin Heidelberg.
- Salama, Y. (2019). How Do Electric Fish Produce Electricity? Retrieved from <https://www.scienceabc.com/nature/animals/how-do-electric-fish-produce-electricity.html>. Accessed from January 22, 2022.
- Salawudeen, A. T., Mu'azu, M. B., Yusuf, A., & Adedokun, A. E. (2021). A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application. *Knowledge-Based Systems*, 232, 107486. <https://doi.org/10.1016/j.knosys.2021.107486>
- Salem, S. A. (2012). In *BOA: A novel optimization algorithm* (pp. 1–5). IEEE. <https://doi.org/10.1109/ICEngTechnol.2012.6396156>
- Seyyedabbasi, A. (2022). WOASCALE: A new hybrid whale optimization algorithm based on sine cosine algorithm and levy flight to solve global optimization problems. *Advances in Engineering Software*, 173, Article 103272. <https://doi.org/10.1016/j.advengsoft.2022.103272>
- Shi, Y. (2011). *Brain storm optimization algorithm*. In *International conference in swarm intelligence* (pp. 303–309). Berlin, Heidelberg: Springer.
- Shunmugapriya, P., & Kanmani, S. (2017). A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). *Swarm and evolutionary computation*, 36, 27–36. <https://doi.org/10.1016/j.swevo.2017.04.002>
- Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702–713. <https://doi.org/10.1109/TEVC.2008.919004>
- Singh, P., Chaudhury, S., & Panigrahi, B. K. (2021). Hybrid MPSO-CNN: Multi-level particle swarm optimized hyperparameters of convolutional neural network. *Swarm and Evolutionary Computation*, 63, Article 100863. <https://doi.org/10.1016/j.swevo.2021.100863>
- Smithsonian. (2021). Scientists discover electric eels hunting in a group. Retrieved from <https://www.eurekalert.org/news-releases/879792>. 14 January 2021.
- Tahani, M., & Babayan, N. (2019). Flow Regime Algorithm (FRA): A physics-based meta-heuristics algorithm. *Knowledge and Information Systems*, 60(2), 1001–1038. <https://doi.org/10.1007/s10115-018-1253-3>
- Tanyildizi, E., & Demir, G. (2017). Golden Sine Algorithm: A Novel Math-Inspired Algorithm. *Advances in Electrical & Computer Engineering*, 17(2), 71–78. <https://doi.org/10.4316/aece.2017.02010>
- Tikhamarine, Y., Souag-Gamane, D., Ahmed, A. N., Kisi, O., & El-Shafie, A. (2020). Improving artificial intelligence models accuracy for monthly streamflow forecasting using grey Wolf optimization (GWO) algorithm. *Journal of Hydrology*, 582, Article 124435. <https://doi.org/10.1016/j.jhydrol.2019.124435>
- Too, J., & Abdullah, A. R. (2020). Chaotic atom search optimization for feature selection. *Arabian Journal for Science and Engineering*, 45(8), 6063–6079. <https://doi.org/10.1007/s13369-020-04486-7>
- Veysari, E. F. (2022). A new optimization algorithm inspired by the quest for the evolution of human society: Human felicity algorithm. *Expert Systems with Applications*, 193, Article 116468. <https://doi.org/10.1016/j.eswa.2021.116468>
- Viswanathan, G. M., Afanasyev, V., Buldyrev, S. V., Murphy, E. J., Prince, P. A., & Stanley, H. E. (1996). Lévy flight search patterns of wandering albatrosses. *Nature*, 381(6581), 413–415. <https://doi.org/10.1038/381413a0>
- Wagan, A. I., & Shaikh, M. M. (2020). A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem. *Applied Soft Computing*, 90, Article 106176. <https://doi.org/10.1016/j.asoc.2020.106176>
- Wang, L., Zhang, J., Cao, Q., Wang, G., & Wang, X. (2022). Study on Emergency Control Strategy of Gates under accident tripping condition of hydropower Station. *Water Power*, 48(9), 92–97. <https://doi.org/10.3969/j.issn.0559-9342.2022.09.018>
- Wang, G. G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural computing and applications*, 31(7), 1995–2014. <https://doi.org/10.1007/s00521-015-1923-y>
- Wang, J. F., Periaux, J., & Sefrioui, M. (2002). Parallel evolutionary algorithms for optimization problems in aerospace engineering. *Journal of Computational and Applied Mathematics*, 149(1), 155–169. [https://doi.org/10.1016/S0377-0427\(02\)00527-7](https://doi.org/10.1016/S0377-0427(02)00527-7)
- Wang, J. S., & Li, S. X. (2019). An improved grey wolf optimizer based on differential evolution and elimination mechanism. *Scientific reports*, 9(1), 1–21. <https://doi.org/10.1038/s41598-019-43546-3>
- Wang, J., Zhan, C., Li, S., Zhao, Q., Liu, J., & Xie, Z. (2022b). Adaptive variational mode decomposition based on Archimedes optimization algorithm and its application to bearing fault diagnosis. *Measurement*, 191, Article 110798. <https://doi.org/10.1016/j.measurement.2022.110798>
- Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 114, Article 105082. <https://doi.org/10.1016/j.engappai.2022.105082>
- Wang, P. C., & Shoup, T. E. (2011). Parameter sensitivity study of the Nelder-Mead simplex method. *Advances in Engineering Software*, 42(7), 529–533. <https://doi.org/10.1016/j.advengsoft.2011.04.004>
- Wang, Y., Liu, H., Long, H., Zhang, Z., & Yang, S. (2017). Differential evolution with a new encoding mechanism for optimizing wind farm layout. *IEEE Transactions on Industrial Informatics*, 14(3), 1040–1054. <https://doi.org/10.1109/TII.2017.2743761>
- Wikipedia. (2022). Electric Eel. Retrieved from https://en.wikipedia.org/w/index.php?title=Electric_eel&oldid=107109099. Accessed January 10, 2022.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- Wu, G., Mallipeddi, R., & Suganthan, P. N. (2017). Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report.
- Yadav, A., & Kumar, N. (2020). Artificial electric field algorithm for engineering optimization problems. *Expert Systems with Applications*, 149, Article 113308. <https://doi.org/10.1016/j.eswa.2020.113308>
- Yan, Z., Duan, X., Chang, Y., Xu, Z., & Sobhani, B. (2023). Optimal energy management in smart buildings with electric vehicles based on economic and risk aspects using developed whale optimization algorithm. *Journal of Cleaner Production*, 137710. <https://doi.org/10.1016/j.jclepro.2023.137710>
- Yang, S., Gu, X., Liu, Y., Hao, R., & Li, S. (2020). A general multi-objective optimized wavelet filter and its applications in fault diagnosis of wheelset bearings. *Mechanical Systems and Signal Processing*, 145, Article 106914. <https://doi.org/10.1016/j.ymssp.2020.106914>
- Yang, X. S., & Deb, S. (2014). Cuckoo search: Recent advances and applications. *Neural Computing and Applications*, 24(1), 169–174. <https://doi.org/10.1007/s00521-013-1367-1>
- Yao-sheng, S., Zhang-can, H., Yu, C., & Chao, Y. (2014, April). A new discrete eel swarm intelligence algorithm. The 4th IEEE International Conference on Information Science and Technology, Shenzhen, China. <https://doi.org/10.1109/ICIST.2014.6920601>
- Yilmaz, S., & Sen, S. (2020). Electric fish optimization: A new heuristic algorithm inspired by electrolocation. *Neural Computing and Applications*, 32(15), 11543–11578. <https://doi.org/10.1007/s00521-019-04641-8>
- Yousri, D., Abd Elaziz, M., Oliva, D., Abualigah, L., Al-Qaness, M. A., & Ewees, A. A. (2020). Reliable adaptive objective for identifying simple and detailed photovoltaic models using modern metaheuristics: Comparative study. *Energy Conversion and Management*, 223, Article 113279. <https://doi.org/10.1016/j.enconman.2020.113279>
- Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2022). Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Computer Methods in Applied Mechanics and Engineering*, 392, Article 114616. <https://doi.org/10.1016/j.cma.2022.114616>
- Zhao, S., Zhang, T., Ma, S., & Wang, M. (2022). Sea-horse optimizer: A novel nature-inspired meta-heuristic for global optimization problems. *Applied Intelligence*, 1–28. <https://doi.org/10.1007/s10489-022-03994-3>
- Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 388, Article 114194. <https://doi.org/10.1016/j.cma.2021.114194>
- Zhao, W., Wang, L., & Zhang, Z. (2019). Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems*, 163, 283–304. <https://doi.org/10.1016/j.knosys.2018.08.030>
- Zhao, W., Zhang, Z., & Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence*, 87, Article 103300. <https://doi.org/10.1016/j.engappai.2019.103300>
- Zhong, C., Li, G., & Meng, Z. (2022). Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-Based Systems*, 109215. <https://doi.org/10.1016/j.knosys.2022.109215>
- Angeline, P. J. (1994). *Genetic programming: On the programming of computers by means of natural selection*: John R. Koza. A Bradford Book, MIT Press.
- Back, T. (1996). *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*. New York: Oxford University Press.
- Brunetti, G., Stumpf, C., & Šimůnek, J. (2022). Balancing exploitation and exploration: A novel hybrid global-local optimization strategy for hydrological model calibration. *Environmental Modelling & Software*, 150, Article 105341. <https://doi.org/10.1016/j.envsoft.2022.105341>
- Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., & Woodward, J. R. (2010). *A classification of hyper-heuristic approaches*. *Handbook of metaheuristics* (pp. 449–468). Springer.
- CARSON M. (2021). Retrieved from <https://www.courthousenews.com/electric-eels-more-creative-social-hunters-than-previous-thought/>. 16 January 2021.
- Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267(1), 66–73. <http://www.jstor.org/stable/24939139>
- Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN '95-international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
- Melman, A., & Evtusin, O. (2023). Comparative study of metaheuristic optimization algorithms for image steganography based on discrete Fourier transform domain. *Applied Soft Computing*, 132, Article 109847. <https://doi.org/10.1016/j.asoc.2022.109847>
- Mora-Gutiérrez, R. A., Ramírez-Rodríguez, J., & Rincón-García, E. A. (2014). An optimization algorithm inspired by musical composition. *Artificial Intelligence Review*, 41, 301–315. <https://doi.org/10.1007/s10462-011-9309-8>
- Nikki, M. (2016). Energy Storing Fiber. Retrieved from <https://www.eeweb.com/energy-storing-fiber>. January 22, 2022.