Expert Systems With Applications 228 (2023) 120242

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Fans Optimizer: A human-inspired optimizer for mechanical design problems optimization☆

Xiaofei Wang [a,1], Jiazhong Xu [a,b,*,1], Cheng Huang [b,1]

[a] Key Laboratory of Advanced Manufacturing and Intelligent Technology Ministry of Education, School of Mechanical and Power Engineering, Harbin University of Science and Technology, Xuefu Road No. 52, Harbin, 150080, China

[b] Heilongjiang Provincial Key Laboratory of Complex Intelligent System and Integration, School of Automation, Harbin University of Science and Technology, Xuefu Road No. 52, Harbin, 150080, China

ARTICLE INFO

Keywords:
Optimization
Fans optimization(FO)
Meta-heuristics
Benchmark
Local optimum
Human-inspired

ABSTRACT

This paper proposes a human-inspired algorithm for optimizing various practical engineering problems, specifically, the Fans Optimization (FO) algorithm that is inspired by the Fans economy fused in the entertainment domain. Opposing current algorithms, the FO algorithm introduces a Multi-groups mechanism (Cooperation–Competition) and a Two-characteristic individual update mechanism to balance the exploration and exploitation (E&E). Additionally, the multi-phase optimization algorithm includes Roles information and Fans-community, Fans-community Switching, Resource Competition, Information Sharing, and K-Update. In the experiments, the FO algorithm is first compared with 12 meta-heuristic algorithms in four groups of benchmark functions (selected from the IEEE CEC and GECCO competitions), demonstrating that the FO algorithm has a superior convergence performance and E&E. Moreover, the Williams test prove the superiority of the FO algorithm over the competitor algorithms. Additionally, eight class practical engineering problems verify the FO's practical engineering applicability. Finally, the FO algorithm solves the inverse kinematic solution problem of the 9-degree of freedom serial robot arm (9-DFSRA) , demonstrating superior results over the competitor schemes. Overall, the test results prove that the FO algorithm is superior to its competitor algorithms for complex engineering problems.

1. Introduction

In recent decades, global optimization has received significant attention from researchers. Traditional approximation algorithms encounter difficulties when dealing with discontinuous and dynamically uncertain problems. Furthermore, as engineering applications become increasingly complex and challenging, swarm intelligence meta-heuristic algorithms offer a highly effective solution to the intractability of optimization problems. Meta-heuristic algorithms are a class of optimization algorithms that model optimization processes found in nature (such as evolution or physical phenomena) to search for optimal solutions. They have a wide range of applications in fields such as engineering design, machine learning, and finance. Examples of meta-heuristic algorithms include Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and simulated annealing algorithms, among others (Al Salami, 2009; Holland, 1992; Kennedy & Eberhart, 1995). These algorithms operate in two stages: exploration and exploitation. During the exploration phase, the algorithm aims to explore the solution space by allowing the agents to spread randomly and wander around the space, thus increasing the possibility of discovering new solutions. This phase determines the algorithm's ability to escape local optima and search for global optima. During the exploitation phase, the agents focus on refining their search in the vicinity of the previously discovered solutions to find more accurate solutions. This phase determines the accuracy of the algorithm's search for an optimum. Therefore, the key to a successful optimization algorithm that can obtain a globally optimal solution lies in striking a balance between exploration and exploitation. The uniqueness of swarm intelligence meta-heuristic algorithms lies in their

☆ This work was supported by : Heilongjiang Province Applied Technology Research and Development Program (Grant No. GA20A401).

* Corresponding author at: Heilongjiang Provincial Key Laboratory of Complex Intelligent System and Integration, School of Automation, Harbin University of Science and Technology, Xuefu Road No. 52, Harbin, 150080, China.

E-mail addresses: 1820510049@stu.hrbust.edu.cn (X. Wang), xujiazhong@hrbust.edu.cn (J. Xu), Hc@hrbust.edu.cn (C. Huang).

URLs: https://sciprofiles.com/profile/XiaofeiWang (X. Wang), https://sciprofiles.com/profile/JiazhongXu (J. Xu), https://sciprofiles.com/profile/ChengHuang (C. Huang).

[1] These authors have contributed equally to this work.

https://doi.org/10.1016/j.eswa.2023.120242
Received 5 January 2023; Received in revised form 5 April 2023; Accepted 22 April 2023
Available online 29 April 2023
0957-4174/© 2023 Elsevier Ltd. All rights reserved.

ability to model rules that enable low-intelligence individuals to exhibit swarm intelligence at the macro level. Hence, in these algorithms, perfecting the rules is more crucial than updating the agent steps. However, current swarm intelligence meta-heuristic algorithms, such as salp swarm algorithm (SSA), PSO, Whale Optimization Algorithm (WOA), etc., mainly focus on the cooperation mechanism among agents within a single swarm, while overlooking the competition within the swarm and inter-group cooperation and competition (Mirjalili et al., 2017; Mirjalili & Lewis, 2016).

Contribution: To overcome this shortcoming, this paper proposes a multi-group game meta-heuristic algorithm that comprehensively considers intra- and inter-group games (cooperation and competition) and emphasizes the design of group rules rather than solely focusing on the design of agent step updates. Timing the balance between algorithmic exploration and exploitation and the individual position updates are critical in determining an algorithm's performance. Thus, the contributing of the proposed FO algorithm in meta-heuristic algorithms is as follows:

- Unlike the current algorithm, the proposed algorithm introduces game theory (Cooperation–Competition) between the communities (**Multi-groups mechanism**). This strategy balances exploration and exploitation at the fundamental level.
- Various attributes of the individuals within the community are considered to enrich the updating method of the individuals' locations and provides more opportunities for individuals.
- Each agent in this article has dual characteristics (**Two-characteristic individuals**), i.e., role and Fans-community. This initiative affords the FO algorithm to consider inter- and intra-games simultaneously.

This study conducts a comparative analysis to evaluate the performance of several existing meta-heuristic algorithms. The experiments include simulations on four sets of benchmark functions and eight typical classical engineering problems. On all trials, the FO algorithm produces appealing results. Additionally, the FO algorithm is applied to solve the inverse kinematic problem of the 9-DFSRA, yielding promising results.

The remainder of this paper is as follows: Section 3 overviews of the inspiration and mathematical principles behind the FO algorithm. Section 4 compares the FO algorithm and several classical meta-heuristic algorithms on four groups of benchmark functions, followed by a discussion of the findings. Section 5 compares experiment between the FO algorithm and several classical meta-heuristic algorithms on eight real-world engineering problems. Additionally, this section investigates the application of the FO algorithm to the inverse kinematic design of a robotic space arm. Finally, Section 6 concludes, analyzes, and summarizes this paper.

2. Related work

Considering the relevance of the research presented in this paper to the field of meta-heuristic algorithms, it is necessary to discuss some previous classical or novel meta-heuristic algorithms. An analysis of the characteristics and limitations of previous meta-heuristic algorithms is presented and compared with the proposed algorithm in this paper.

Among these, meta-heuristics, inspired by natural phenomena, have emerged as a representative categorized into physical-inspired, biological-inspired, and human-inspired algorithms (Siddique & Adeli, 2017). Fig. 1 illustrates the meta-heuristic categories corresponding representative algorithms.

2.1. Biological-inspired algorithms

One major meta-heuristics category is the biological-inspired class inspired by natural foraging behavior. In the past two decades, such algorithms have received significant research attention. Some of the most notable and commonly used algorithms in this category are the GA, PSO, and ACO, among others. It should be noted that the GA algorithm is more suitable for low-dimensional problems, but may converge too quickly. Besides the PSO algorithm is a robust global search algorithm, but it is sensitive to initial values and parameters, and therefore is unsuitable for discrete solution problems. Compared to the GA and PSO algorithms, the ACO algorithm is relatively simpler to implement and debug. Nonetheless, due to the multiple parameters, the ACO algorithm may experience slower convergence and a higher likelihood of being trapped in local optima. Thus, subsequent studies aim to enhance these algorithms, and suggested the Non-Uniform Genetic Algorithm Multi-Objective Genetic Algorithm (Pradhan, Wang, Ali, Yue, & Liaaen, 2021), Seed-Based Genetic Algorithm (Kabir, Xu, Kang, & Zhao, 2017), Hybrid Genetic Algorithm (Misevicius & Verene, 2021), Chaos Adaptive Particle Swarm Optimization (Duan et al., 2022), Improved Weighted Particle Swarm Optimization (Guo, Zhang, Song, & Qian, 2018), Whale Optimization Algorithm and Particle Swarm Optimization (Asghari, Masdari, Gharehchopogh, & Saneifard, 2021), Combination of Multiple Ant Colony Optimization (Zhu, Liu, & Ghosh, 2019), and Improved multi-ant colony theory (Yue, Xi, & Guan, 2019) etc. However, these changes do not modify the core updating mechanism, i.e., the balance between exploration and exploitation. As scientific methodologies continue to advance, new algorithms have been proposed to address complex engineering optimization problems. Recently, several highly effective biological-inspired algorithms emerged, including the social-spider optimization (SSO) algorithm inspired by the social behavior of fireflies and spiders (Hashmi, Goel, Goel, & Gupta, 2013). Moreover, the WOA mathematically modeled the behavior of whales as they use air bubbles to feed on swarms of fish, which reasonably incorporates exploitation (Mirjalili & Lewis, 2016). In 2017, Mirjalili proposed a biological-inspired algorithm called the SSA, which achieves higher arithmetic accuracy at lower iterations inspired by the foraging and exploratory behavior of the salp swarm (Mirjalili et al., 2017). Hisham.S developed a sperm swarm optimization algorithm (SSO2) inspired by the purposeful sperm population movement (Shehadeh, Ahmedy, & Idris, 2018).

Recent biological-inspired algorithms simulate animal swarm behavior by modeling the behavior of individuals within a community to build new algorithms that are essentially similar, differing only in the individual's position update formula. In contrast, the FO algorithm proposed in this paper introduces the concept of a multi-community game at the fundamental logic level, where exploration and exploitation are balanced on the most fundamental update mechanism of the algorithm. As a result, the exploration and exploitation balance is achieved spontaneously by the community's natural competition rather than through manual parameter setting, like in existing algorithms.

2.2. Physically-inspired algorithms

The physically-inspired meta-heuristic algorithms have gained significant attention across various disciplines in the past three decades (Siddique & Adeli, 2016). The concept of physical inspiration was first introduced in 1982 and has since gained increasing attention across several disciplines. Indeed, the simulated annealing algorithm is one of the earliest examples of a physics-inspired algorithm (Wang, Xu, & Liu, 2018). Among the more classic algorithms of the field of physics-inspired algorithms are the attraction–repulsion mechanism algorithm (Birbil & Fang, 2003), adaptive step search (SASS) (Nolle, 2006), quantum genetic algorithm (QGA) (Abraham, Wahyunggoro, & Setiawan, 2019), hysteretic optimization algorithm (HO) (Pál, 2006), COLSHADE (Gurrola-Ramos, Aguirre, & Cedeño, 2020) and central force optimization (CFO) (Formato, 2009). The recent physical-inspired algorithms include the gravitational search algorithm (GSA), harmony search algorithm (HSA), Equilibrium Optimizer (EO) algorithm and water drop algorithm (WDA) (Faramarzi, Heidarinejad, Stephens, &

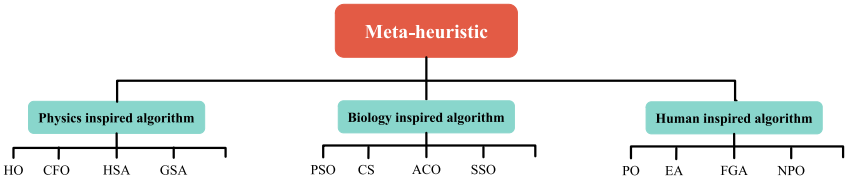

Fig. 1. Categories char of Meta-heuristic and their respective representative algorithms.

Table 1
List of all categories of roles and their corresponding description.

Name		Description
P_1^j	Benefit operator	It refers to benefit-driven managers who distribute guided information downwards. Frequently, these are the first tier of managers with firsthand knowledge, the most significant influence, and the highest loyalty.
P_2^j	Emotional operator	This refers to emotionally driven individuals who have a certain social influence and are mainly responsible for re-processing the messages from Benefit in the same community.
P_3^j	Follower	This category refers to blinkered operators with a certain degree of influence. Typically, in reality, it occupies the largest percentage of managers. The most significant difference from other managers is that it can receive information from other communities.
P_4^j	FansLeader	There is a group of fans with the most fervent enthusiasm, a certain ability to screen information, promote their community feverishly to people around them, simultaneously consciously increase their followers and, to a certain extent, become a leader in a small area.
P_5^j	Courier	This refers to individuals within the fans who receive information without screening it. They can receive information from the same community and other communities, making them somewhat of a deviant group.
P_6^j	Protector	The fans only receive information from the same community, and they ensure the minimum interests of their community to a certain extent.
	Assistantp_{6,n_6}^j	Assistant is a unique role of Protector; each Assistant receives information from all the other communities and finds the most helpful one to receive it. Interestingly, sometimes this community is not the most influential one, but for the Assistant, it allows the Assistant to achieve the most influence.
P_7^j	Deviator	This refers to the one with the lowest influence within the community (simplified loyalty approximates impact in this paper), which means it will shift to the community most similar to Deviator.

Mirjalili, 2020). Besides, numerous scholars have enhanced existing physics-inspired algorithms improving the simulation results. For instance, in 2020, Mittal optimized the parameters of the GSA algorithm, achieving superior simulation results on benchmark functions (Mittal, Tripathi, Pandey, & Pal, 2020). Esra introduced the Taguchi method to improve the parameters of the hybrid harmony search algorithm in 2022, resulting in better simulation results in solving optimization problems (Uray, Carbas, Geem, & Kim, 2022). Moreover, Kumar proposed a novel approach called the Modified Covariance Matrix Adaptation Evolution Strategy (sCMAgES) to solve constrained optimization problems (COPs) by introducing a Variance Covariance Matrix Adaptation Evolution Strategy (CMA-ES) with linear time complexity. The proposed method demonstrated promising results in solving COPs and outperformed several state-of-the-art algorithms (Kumar, Das, & Zelinka, 2020).

Although researchers optimized the previously covered algorithms, where the features of these physics-inspired algorithms can still be summarized as the concentrated amplification of a physical law from physical reality. Researchers have translated these features into algorithms, enhancing regularity during the evolution of these physics-inspired algorithms. However, the balance between the overall exploration and exploitation of algorithms must be completed.

2.3. Human-inspired algorithms

Human-inspired algorithms are a new category of meta-heuristic algorithms that have emerged in recent years, inspired by human social behavior. One of the earliest human-inspired algorithms is the Imperialist Competitive Algorithm (ICA) proposed in 2008 (Atashpaz-Gargari & Lucas, 2008). Several high-performing social behavior algorithms have emerged since then, such as the election algorithm (EA) in 2015, which is the mathematical model of presidential elections (Emami & Derakhshan, 2015). Additionally, Elyas proposed a football game algorithm (FGA) inspired by human teamwork in football training. It should be noted that 2020 is the peak of the meta-heuristics algorithm, as Sinan proposed a nomadic people optimization (NPO) strategy inspired by the movement and searching behavior of nomadic people (Salih & Alsewari, 2020). Moreover, Qamar proposed heap-based optimization (HBO), which is the mathematical model of corporate-rank-hierarchy (CHR) by mapping heap data structures (Askari, Saeed and Younas, 2020). Finally, a political optimizer (PO) is a novel social-inspired algorithm that obtains excellent search performance modeling U.S. election policies (Askari, Younas and Saeed, 2020).

Opposing the previously covered algorithms, human-inspired algorithms simulate mechanisms rather than just how step-wise updates are performed. In human-inspired algorithms, scholars focus more on the rules at a macro level, including the cooperation and competition among individuals, rather than just limiting themselves to changes in individual positions. Also, human-inspired algorithms merely consider cooperation and competition between individuals at the macro level rather than between communities.

3. Comparative analysis of the FO algorithm

3.1. Inspiration

The unique multi-communities, classification, and division of information delivery within the entertainment domain inspire the developed FO algorithm. The widespread adoption of the internet has led to the growth of the entertainment economy, increasing the control of idols over their fan community. As a result, an economically driven fan class structure has emerged. The concept of virtual communities, referring to digital communication-mediated social networks, was first introduced in the literature (Yang & Shim, 2020). By integrating, refining, and extending the concepts related to the fan economy, as covered in the literature (Lyu & Wang, 2021; Qi, Jiang, Yang, & Ying, 2022; Zhou, 2021), the fan community has been divided into a pyramidal hierarchical structure as illustrated in Fig. 4. Based on this structure,

we propose the FO algorithm. Similar to other swarm-based intelligence algorithms, such as ACO and PSO, the FO algorithm relies on resource competition. In entertainment, fans are divided into various camps based on the celebrities they follow. Each camp strives to gain more significant influence (votes) by disseminating information at different levels. Therefore, in the FO algorithm, individual support corresponds to the significance of food in other group intelligence algorithms. Hence the FO algorithm aims to identify individuals with the highest support. To process information in the FO algorithm, the member attributes of each camp are classified as Benefit operator, Emotional operator, Follower, FansLeader, Courier, Protector, Assistant, and Deviator Further details are presented in Table 1 for details.

Interestingly, unlike other swarm intelligence algorithms, a member's attribute (role) in FO algorithm is complex, and the relationships among the different communities are not invariable during the iteration process.

Attribute Complexity: The competition for resources in the entertainment domain divides the communities based on the celebrities (e.g., actors, stars, singers, etc.) they follow. Each community's members' attributes (roles) are complex and ordered. However, human control inevitably becomes involved in competing interests. This paper refers to these controllers as managers, and thus the community can be divided into two main groups: managers and fans. Managers are individuals with a certain level of influence who promote the celebrity's force (power) intentionally and profitably. They can be classified into three categories based on their motivation: benefit operator, emotional operator, and blind (follower), see Tables 1 and 2 for further details. Similarly, individuals within the group of fans are classified according to their function as leader, courier, assistant, deviator, and protector (see Table 1 for details). It is important to note that the relationship between the managers and the fans groups is not of absolute control, and the hierarchical situation is not constant during the resources competition. Specifically, as the influence of an individual within the group increases, there might be a gradual transition from fans to managers. After introducing the concept of **Attribute Complexity** within a community, the following part of the paper will discuss the relationship between communities, including **Cooperation–Competition**.

Cooperation–Competition: Unlike other uni-group (single community) algorithms, the FO algorithm involves the coexistence of multiple communities, which have a dynamically changing relationship among communities. The objective of each community is to become the most influential, leading to fluid competition among communities. However, in the real world, as the influence of communities grows, mere competition is no longer enough to satisfy practical needs. Thus, community relationships transform into cooperation to achieve a more substantial impact. During the **Competition** stage, not all communities have an extensive reach of influence to occupy a significant percentage of the influence space. Therefore, communities work individually to expand their dominance as much as possible and occupy more influence space, which essentially maps to the physical exploration of meta-heuristic algorithms. In the **Cooperation** stage, the dominance of all communities is already large enough to occupy a significant percentage of the influence space. As a result, the efficiency of communities exploring independently during this stage plummets. Therefore the aim of this stage has changed from the expansion of dominance to exploitation, which essentially maps to the physical implications of the exploitation of meta-heuristic algorithms (See Figs. 3 and 2). Hence, communities seek to cooperate during this stage and more efficiently locate the most influential individuals from total dominance and their neighborhoods.

In the FO algorithm, the totally optimized sequences include **Resource Competition**, **Fans-community Switching**, and **Information Sharing**. Fig. 5 illustrates the flowchart of the FO algorithm, which incorporates a territorial erosion mechanism based on a game mechanism with multi-role and a multi-group **Cooperation–Competition** exploration model.

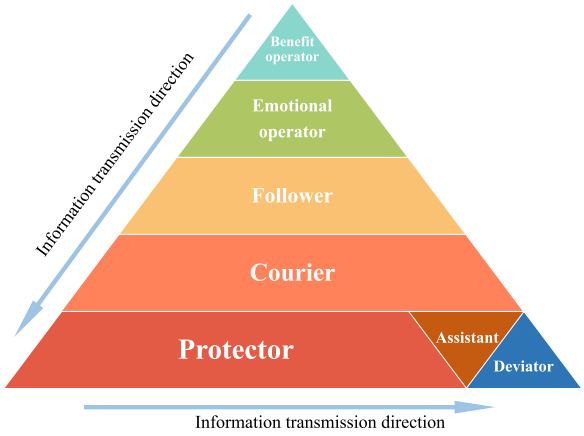

Fig. 2. Diagram of all roles involved in the FO algorithm.

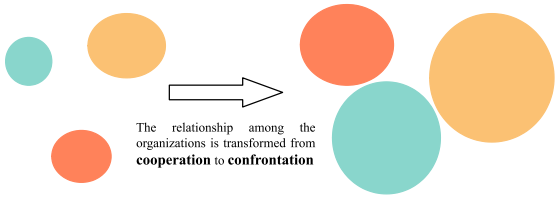

Fig. 3. Relationship among the organizations is transformed from cooperation to confrontation involved in the FO algorithm.

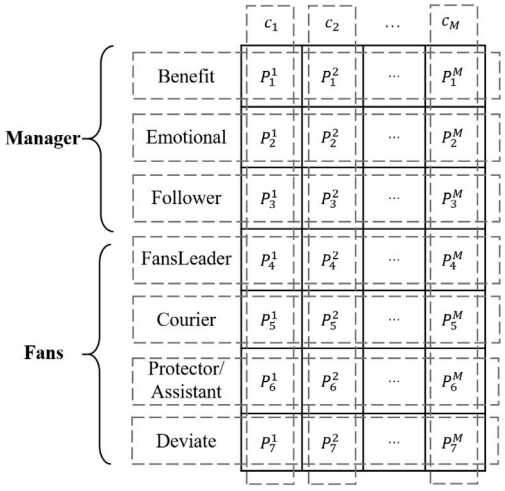

Fig. 4. Diagram of all roles involved in the FO algorithm.

3.2. Mathematical representation of the proposed algorithm

This section mathematically defines the proposed model inspired by the development of the multi-Fans-communities. The proposed algorithm is expressed as Procedure 2. All variables involved and their corresponding description are presented in Table 2.

3.2.1. Roles information and fans-community

According to Eq. (1), the Fans-community C is divided into M communities, and the jth Fans-community comprises seven types of roles P_i^j, as defined in Eq. (2).

$$C = \{c_1, c_2, \ldots, c_M\} \tag{1}$$

$$c_j = \left\{ P_1^j, P_2^j, \ldots, P_r^j \right\} \quad r = 1, 2, \ldots, 7 \tag{2}$$

Table 2
Interpretation of the variables involved in the mathematical model of the FO algorithm.

Variable	Interpretation
C	Set of all Fans-communities.
c_j	jth Fans-community.
P	Set of all roles.
P_i^j	ith roles of the jth Fans-community.
$p_{i,e}^j$	eth member of ith roles of the jth Fans-community.
n	The number of population.
M	The number of Fan-community.
$Ratio_o$	Proportional matrix of roles.
K	$n \times T_{\max}$ matrix is used to store the numbers of the Fans-community to which individuals have belonged during the iteration.
$Tmax$	Total iterations.
t	Current number of iterations.
d	Dimension of the solution.
r	Number of labels in the role category.
n_i	Number of members of ith role.

Procedure 1 Main Framework of the FO algorithm

Input: M; n; Ratio$o$, $Tmax$; K.
Output: P($Tmax$)(the final position of the population); $Convergence_{curve}$(the $Convergence_{curve}$ matrix which stores the optimal fitness values for each iteration).
/***Initialization** */
Initialize P according to Eqs. (1)–(4) and Fig. 4. Assigning the Fans-community of individuals according to Eq. (2).
while t < Tmax **do**
  $C_{temp} = C$; $f(C_{temp}) = f(C)$
  for each $c_j \in C$ **do**
    Update $Ratio(t)$ according to Eq. (5).
    Assigning the role to each individual as expressed in Procedure 2.
    Fans-community Switching(c_j, P_i^j)
    /***K Constraint***/
    Binding matrix K(t) to keep that each Fans-community contains at least one member.
    c_j=***Resource Competition***$(c_j, c_j(t-1))$
  end for
  Information Sharing
  K-Update
end while

The role is another characteristic of the population P, which is divided into seven types, as expressed in Eq. (3). Each role of the population P_i^j contains ni members,where n_i is a random integer greater than 0. Each member (candidate solution) has two labels. I.e., jth Fans-community and ith role, with further details presented in Fig. 4.

$$P = \left\{ P_1^j, P_2^j, \ldots, P_r^j \right\} \quad r = 1, 2, \ldots, 7 \tag{3}$$

$$P_i^j = \{ p_{i,1}^j, p_{i,2}^j, \ldots, p_{i,n_i}^j \} \tag{4}$$

$$\mathrm{Ratio}(t) = \begin{bmatrix} 1 \\ \mathrm{Curve}_{\mathrm{down}} \\ \mathrm{Curve}_{\mathrm{up}} \\ \mathrm{Curve}_{\mathrm{up}} \\ (1 + \mathrm{Curve}_{\mathrm{up}}) \end{bmatrix}^{\mathrm{T}} \times \mathrm{Ratio}_o \tag{5}$$

$$\mathrm{Curve}_{\mathrm{down}} = e^{-\left(\frac{t}{T_{\max}}\right)^2} \tag{6}$$

$$\mathrm{Curve}_{\mathrm{up}} = \frac{e^{\left(\frac{t}{T_{\max}}\right)^2}}{e} \tag{7}$$

At the first iteration, the corresponding roles are assigned to members within the same community according to the probability Ratio_o. However, the probability $Ratio(t)$ of each role is not constant during the iteration. Therefore, throughout the iteration, the ratio of the emotional operator continuously decreases, while the ratio of the follower the FansLeader, Courier, and Protector/Assistant operators continuously increases. The role is assigned to each individual as expressed in Procedure 2. Fig. 6 illustrates the convergence plot for 500 iterations of $\mathrm{Curve}_{\mathrm{down}}$ and $\mathrm{Curve}_{\mathrm{up}}$.

Procedure 2 Assigning a role to individuals

for j=1:M **do**
  $Index_j = sort(f(c_j))$; $c\_temp_j = c_j(Index_j)$
  for i=1:6 **do**
    $P_i^j = c\_temp_j \left((i-1) \times \frac{\mathrm{Ratio}(t) \times n_i}{sum(\mathrm{Ratio}(t))}(i-1) + 1 : \frac{\mathrm{Ratio}(t) \times n_i}{sum(\mathrm{Ratio}(t))}(i) \right)$
  end for
end for

The dynamic change proportion of roles is due to the group member being different during the development of the Fans-community. Specifically, the fans group is gradually growing at the later stage of development, while the emotional and follower gradually decreases at the later development stage. Therefore, one dynamic scale matrix of the roles $Ratio(t)$ is constructed in the FO algorithm, as presented in Eq. (5), better balancing the E&E of the FO algorithm.

3.2.2. Fan-community switching

During the development of the Fans-community, there exists a unique role called the Deviator (betrayer), which has the worst fitness. Thus, transferring the Deviator back to the Fans-community it belongs to is necessary before updating the population. The detailed process is presented in Procedure 3.

Procedure 3 ***Fans-community Switching*** (c_j, P_i^j)

$P_7^j = \arg\max(f(c_j))$
for j_{temp}=1:M **do**
  $q = \arg\min_{j_{temp} \neq j}(\left| f(P_7^{j_{temp}}) - f(p_{1,a}^j) \right|)$
  $a = \arg\min(f(P_1^j))$
  P_7^j becomes a member of the Fans-community to which $p_{1,a}^q$ belongs.
end for

3.2.3. Resource competition

In this section, the members of each Fans-community having different roles are expressed as Procedure 4. Besides, the multi-role update machine provides more possibilities for optimization.

The benefit operator P_1^j dominates the optimization process, affording the best influence and spontaneous exploratory behavior. However, the emotional operator P_2^j contributes a secondary innovation while

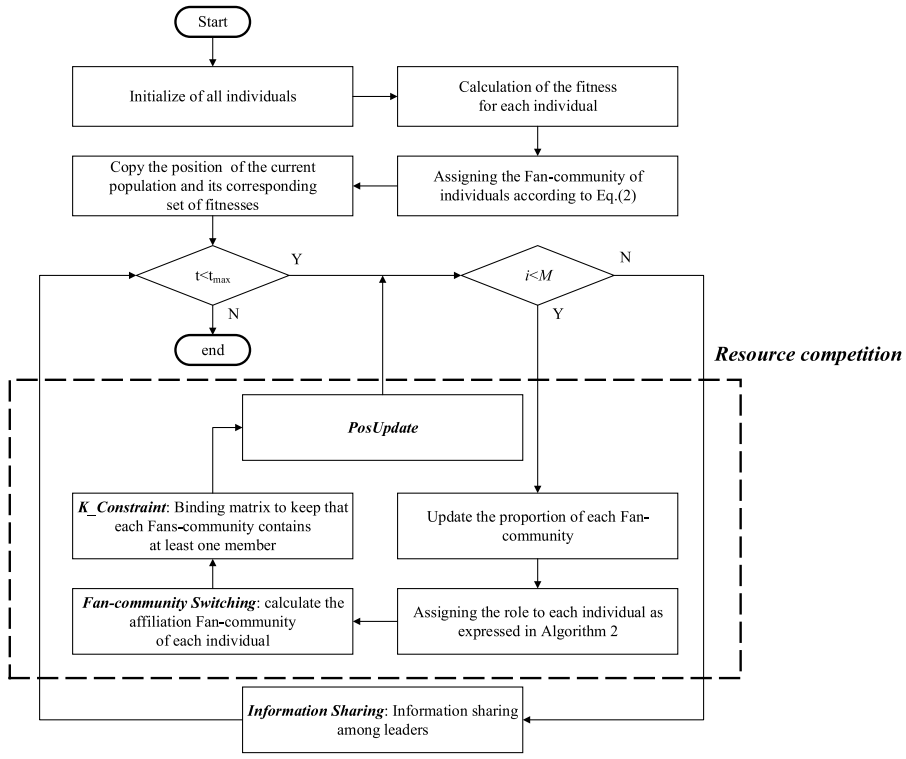

Fig. 5. Flowchart of FO algorithm.

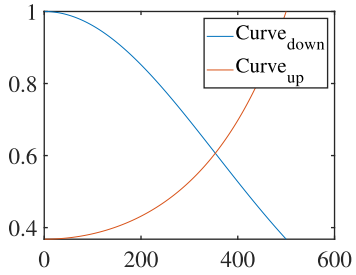

Fig. 6. Balance factor evolution curve of $\text{Curve}_{\text{down}}$ and Curve_{up}.

Procedure 4 ***Resource Competition***$(c_j, c_j(t-1))$

for j=1:M **do**
Update P_1^j according to Eq. (8) and Fig. 7(a).
Update P_2^j according to Eq. (9) and Fig. 7(b).
Update P_3^j according to Eqs. (10), (11) and Fig. 7(c).
Update P_4^j according to Eqs. (12), (13) (14) and Fig. 7(d).
Update P_5^j according to Eq. (15) and Fig. 7(e).
Update P_6^j according to Eq. (16) and Fig. 7(f).
end for

receiving information from the leader $p_{1,1}^j$ of the benefit operator. For example, in Eq.(9), *Leay* is a random matrix generated by the method described in literature (Heidari & Pahlavani, 2017). The rest is follower P_3^j who cannot discern the source of information. Thus, the follower will receive the information from the leader, which fits it, as in Fig. 7(c).

$$P_1^j(t+1) = P_1^j(t) + (2\times r - 1)\times(ub - lb) \tag{8}$$

$$P_2^j(t+1) = P_2^j(t) + (2\times r - 1)\times(p_{1,1}^j(t) - P_2^j(t)) + c_2 \times Leay \times (ub - lb) \tag{9}$$

$$c_2 = e^{-\left(\frac{t}{T_{\max}}\right)^2}/e \tag{10}$$

$$P_3^j(t+1) = P_3^j(t) + r\times(p_{1,1}^q(t) - P_3^j(t)),\quad q = \underset{1\le j\le M}{\arg\min}\left[f(P_3^j(t) - f(p_{1,1}^j(t)))\right] \tag{11}$$

In particular, the FansLeader P_4^j is updated using the mapping transformation between cylindrical and polar coordinates. First, the member of P_4^j moves to the benefit operator's leader $p_{1,1}^j$. Second, the candidate point is obtained by moving a random step from this point to any angle. This phase is divided into two categories according to the number of candidate points, namely, 4-angle and 8-angle update. A detailed schematic is presented in Figs. 7(d) and 8.

$$P_{4temp}^j = P_4^j(t) + r\times(p_{1,1}^j(t) - P_4^j(t)) \tag{12}$$

$$\begin{cases} Rodd(x) = Cylindrical(x) + rand(\rho,\theta,z) \\ Reven(x) = [Polar(x) + rand(\rho,\theta)\ ] \end{cases} \tag{13}$$

Similar to P_3^j, P_5^j receives information from the leader of the benefit operator, while, unlike P_3^j, P_5^j acts as an information Courier who receives information from all leaders of the benefit operators.

$$Ptemp_4^j = \begin{cases} \text{Rodd}(P_{4temp}^j) & if\ d\ is\ odd \& d = 3 \\ \left[Rodd(P_{4temp}^j(1:end-3))\right] & \\ \quad \cup \left[Reven(P_{4temp}^j(end-2:end))\right] & if\ d\ isodd \& d \neq 3 \\ Reven(P_{4temp}^j) & if\ d\ is\ even \end{cases} \tag{14}$$

$$P_5^j(t+1) = P_5^j(t) + r\times\sum_{1\le j\le M}(p_{1,1}^j(t) - P_5^j(t)) \tag{15}$$

$$P_6^j(t+1) = (P_6^j(t) + p_{1,1}^j(t))/2 \tag{16}$$

The Protector P_6^j is the loyal advocate of the benefit operator within the same Fans-community. In the optimization process, it is only responsible for exploitation.

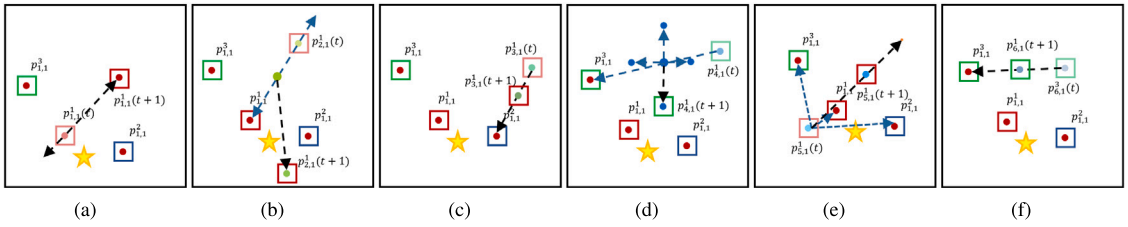

Fig. 7. The updated diagram of each role.

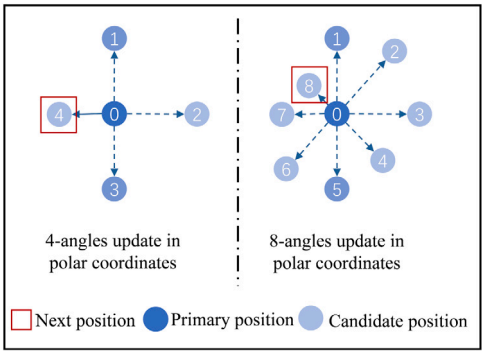

Fig. 8. Schematic diagram of the FansLeader about 4-angle update polar coordinates.

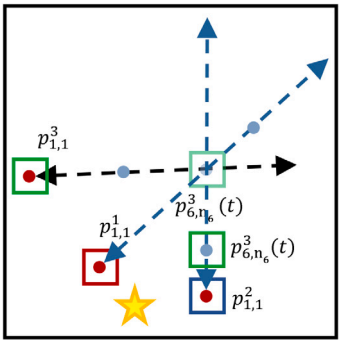

Fig. 9. Updated diagram of the special protector assistant.

3.2.4. Information sharing

The assistant (P_6^j) is a special member of the Protector who obtains better decisions (positions) and assigns a new position to the assistant by communicating with other benefit operators belonging to other Fans-community. Fig. 9 illustrates the updated diagram of the special protector—Assistant.

Procedure 5 ***Information Sharing***$(c_j, c_j(t-1))$

for j=1:M **do**
  $MP_1^j = p_{6,n_6}^j$
  for e=2:M **do**
    $MP_e^j = p_{6,n_6}^j + (2\times r - 1)\times \left|p_{6,n_6}^j - p_{e,1}^j\right|$
    $q = \arg\min_{1\le e\le M}(f(MP_e^j))$
    $p_{6,n_6}^j = MP_q^j$
  end for
end for

3.2.5. K-update

This section introduces the process of updating K, which stores the serial number of the Fans-community where the individual belongs.

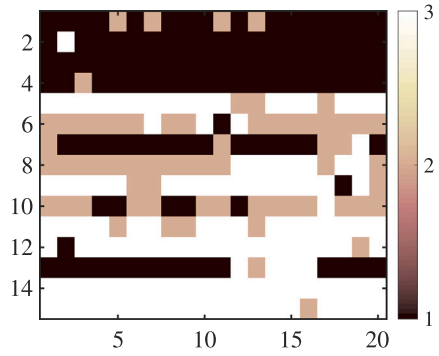

Fig. 10. The changes graph of labels about the individual community over iterations.

For each individual, we calculate the engineering membership in each Fans-community and compare the membership in each community. The Fans-community of the updated individual is the one with the highest membership.

$$Membership_i^j(t+1) = \frac{e^{\frac{t+1}{T_{\max}}} \times number_i^j(t)}{DE(p_i, c_j)\times t \times \text{Mean\_fit\_curve}_i(t)} \tag{17}$$

where $number_i^j(t)$ is the number of times the individual p_i belongs to the jth Fans-community at t iteration, $DE(p_i, c_j)$ is the Euclidean distance between individual p_i and each individual that belongs to the jth Fans-community, and $\text{Mean\_fit\_curve}_i(t)$ is the average optimal fitness of the individual p_i between the 1 and tth iteration. Fig. 10 illustrates the Fans-community transformation graph for each individual with 20 iterations.

Procedure 6 ***K-Update***$(c_j, c_j(t-1))$

for j=1:M **do**
  for i=1:N **do**
    Calculate the membership of each individual according to Eq. (17)
    Compare the membership in each community and update the $K(t)$.
  end for
end for

3.2.6. Mathematical meaning of the major phase

- **Roles information and Fans-community:** In this section, each individual is assigned two characteristics: role and Fans-community. Assigning a Fans-community to each individual facilitates the **Cooperation–Competition** between different forces during the iteration, which contributes to balancing the algorithm's E&E and preventing the algorithm from falling into a local optimum. Moreover, assigning roles to individuals increases the diversity of the update methods and provides more possibilities for the algorithm to explore. This approach enhances the effectiveness of the algorithm in finding optimal solutions.

- **Fan-community Switching:** The exchange object has a specific role, the Deviator, which has the worst fitness value among all individuals. The Deviator is a unique individual designed based on the actual situation, and it is moved to the Fans' community and specifically to an individual with the closest fitness value. Ideally, the Deviator should be the individual with the lowest degree of membership in the Fans community. However, calculating the degree of membership requires additional computation, increasing the algorithm's workload. Therefore, this work considers the Deviator as the individual with the worst fitness since it performs similarly to the Deviator with the lowest membership degree. Exchanging the Deviator improves the algorithm's performance by balancing the E&E and avoiding falling into a local optimum.
- **Resource Competition: Benefit operator:** The benefit operator is updated by randomly exploring within the search space, thus contributing to the algorithm's overall exploration. **Emotional operator:** The emotional operator can re-reprocess outside of receiving information from leaders of the Fans-community. A superior algorithm expects the weights of the exploration capacity to decrease in the later iteration. The mechanism that weight the emotional operator in the FO algorithm is inversely proportional to the number of iterations. **Follower:** The follower is unique as it receives information from the leaders of other Fans-communities and leaders within the same Fans-community. Specifically, it receives information from the leader whose fitness value is closest to theirs, regardless of the Fans-community they belong. This update mechanism increases members' diversity within the Fans community and promotes exploration of the search space. **FansLeader:** It is the leader of Fans, who not only receives information from the leader in the same Fans-community but also recreates the information. Unlike the emotional operator, the FansLeader occupies a gradually increasing proportion of the group during the iteration. **Courier:** The Courier operator differs from the previous operator as it receives information from all leaders and not from the ones within the same Fans-community. This update mechanism is designed to increase information exchange and promote collaboration between Fans-communities. **Protector:** The Protector is the loyal advocate of the benefit operator within the same Fans-community. In the optimization process, it is only responsible for exploitation.
- **Information sharing:** Assistant is a unique member of Protector, who obtains better decisions/positions and assigns the new position to the assistant by communicating with the other benefit operators that belong to other Fans-community.
- **K-Update:** First, calculating the membership of each individual to each Fans-community. Then, comparing the membership of each individual to each Fans-community. Finally, the individual belongs to the Fans-community with the highest membership.

4. Experiments

This section evaluates the performance of the FO algorithm through comparative experiments on well-known benchmark functions, i.e., F, MF, and UF, and against several other classic heuristic algorithms. The simulations are executed in MATLAB-v-2018, and the server parameters include CPU: AMD Ryzen 74800H, 16 GB RAM, and Graphics Card: NVIDIA GeForce RTX 2060. We plan to make the FO algorithm code open-source on MathWorks and GitHub after the paper is accepted. Moreover, this section presents the results of the benchmark function comparison experiments, followed by a discussion on the performance of the FO algorithm. Table 3 summarizes of the parameters of the competitor algorithms. Note that the search agents p of the algorithms not contained in Table 3 is 30.

Table 3
List of parameters for the comparison algorithms involved in this paper.

Algorithm	Parameter
EO	p = 30, V = 1, $a_1 = 2, a_2 = 1, GP = 0.5$
FA	p = 30, $\alpha = 1, \beta_0 = 1, \gamma = 0.01, \theta = 0.97$
PSO	p = 30, $w = 1, c_1 = 1.5, c_2 = 2, w_1 = 0.99$
Others	The optimal parameters given in the paper

4.1. Parameter settings of FO

The parameters of the FO algorithm are simple and flexible. The FO algorithm involves the parameters Ratio_o (1×6 scale matrix for each role), M (the number of the Fans-communities), and p_size (the number of the member of each Fans-community). To balance the E&E, setting the original ratio $Ratio_o = [3, 1, 3, 1, 3, 3]$ of each role. Additionally, a lower rate is assigned to the sentiment operators and FansLeader after multiple simulations. The ratio of the emotional operator continues to decline throughout the iterations.

4.2. Benchmark test functions

The **IEEE CEC and GECCO competitions** are widely recognized annual conferences for evaluating the performance of meta-heuristic algorithms using benchmark functions. Therefore, this paper uses commonly used benchmark functions for meta-heuristic algorithms, as summarized in Digalakis and Margaritis (2001), Liang, Qu, and Suganthan (2013) and Xin and Yong (1999), mainly originating from **IEEE CEC and GECCO**. In this paper, these benchmark functions are grouped into uni-dimensional functions (F) and multidimensional functions (MF) for a more intuitive explanation of the FO algorithm performance.

F: This group comprises uni-modal benchmark functions (F1-7, F10-13) and multi-modal benchmark functions (F8-9, F14-19). Furthermore, the multi-modal benchmark function is divided into variable-dimensional benchmark functions (F8-F9) and fixed-dimensional benchmark functions (F14-F19). It is also worth mentioning that variable-dimensional benchmark functions contribute to testing the invariance of function shifting. Differently, fixed-dimensional benchmark functions are challenging by moving the local near-optimal or setting the Bandwidth Boundary as the optimal global solution before iteration.

MF: This is a group of multi-modal benchmark functions that test the algorithm's exploration capability (in particular, MF4,14,15 are uni-modal benchmark functions). In the subsequent experiments, the multi-modal benchmark function tests the algorithm's performance at a high dimension.

UF: It is the group of the uni-modal benchmark functions (UF1-UF3), simply uni-modal benchmark functions (UF4-UF10), and Hybrid benchmark functions (UF11-UF19), derived from **CEC2017** (Lozano & Jose, 2017). Fig. 11 is the spatial reticulation of benchmark mathematical functions.

CEC2020: This is a group of constrained multi-objective optimization with 10 benchmark functions. It is one of CEC's benchmark functions that have more closely approximated real engineering application problems in recent years (Hussain, Polycarpou, & Yao, 2021).

4.3. Results and discussion

4.3.1. Exploration of the FO algorithm

To evaluate exploration, several comparative experiments are performed on four groups of benchmark functions against other representative algorithms from various major categories of meta-heuristic algorithms, as mentioned in the introduction. A comprehensive analysis of the experimental data in Table A.11–Table A.14 indicates that the FO algorithm is overall better than the competition algorithms on the four groups of benchmark functions.

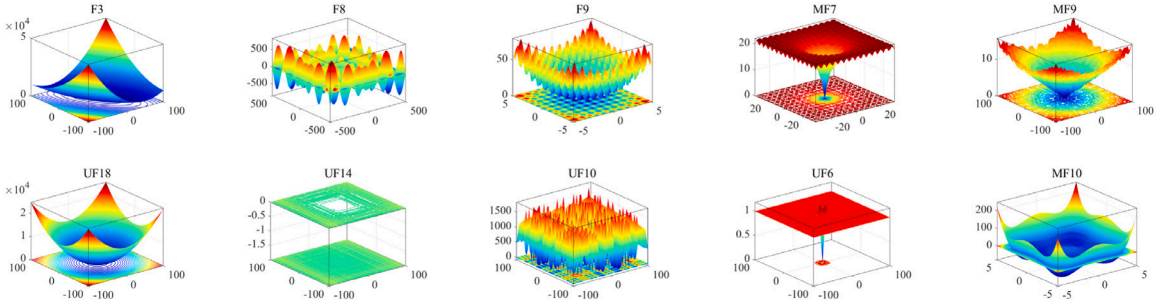

Fig. 11. Spatial reticulation of benchmark mathematical functions.

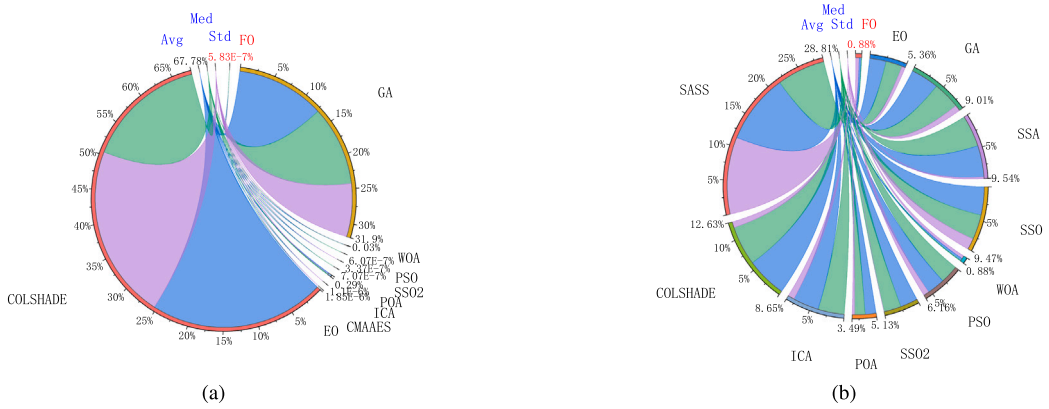

Fig. 12. Chord char of the multi-modal benchmark functions (a) chord char of the fixed-dimensional benchmark functions (b) chord char of the variable-dimensional benchmark functions.

The FO algorithm may not provide the smallest possible solution for a small number of benchmark functions, but it is still equivalent to the best solution. The Fig. 12(b) depicts the chord diagram for the multi-modal benchmark functions, which are further divided into fixed-dimensional and variable-dimensional benchmark functions. The chord diagram visually evaluates the performance of the FO algorithm for each category of the benchmark function. Moreover, the abbreviations MIN, AVG, MED, and STD represent the minimum, average, median, and standard deviation of fitness, respectively. Fig. 12(b) demonstrates that the comprehensive performance of the FO algorithm on the multi-modal benchmark function is much better than the other algorithms, presenting exponential excellence.

Similarly to the unimodel benchmark function, the FO algorithm's performance is very appealing, occupying only a low proportion of the chord chart. Thus the areas of the chord chat are unbalanced and do not facilitate data observation. Therefore, the Fig. 12(b) excludes the worst-performing BA, SSO, and GA algorithms and compares only the top-performing algorithms. The exploration represents the capability of an algorithm to converge to the near-optimal solution in an orderly manner. The initial agent distribution in the FO algorithm is more random than others, and during the iterations, they disperse and gradually converge to the near-optimal solution. This proves that the FO algorithm has a good exploration ability, as illustrated in Fig. 13. To verify this conclusion more objectively, Fig. 14 randomly selects the location trajectories of 6 search agents from the population. The mutational location change of the search agents demonstrates the excellent exploration capability of the FO algorithm and assists the algorithm in breaking out of the local optimum.

4.3.2. Exploitation of the FO algorithm

Similar to exploration, exploitation is an algorithm's necessary evaluation metric. To accurately evaluate this performance, statistical analysis of the data for the uni-modal benchmark functions and explanation with chord chart as presented in Fig. 12(a). Fig. 12(a) is the chord chart containing all algorithms. However, the performance of SASS, SSO, and SSA is poor, forcing the fitness of all uni-modal benchmark functions to occupy a disproportionately large share of the chord chart. Therefore, these worst-performing functions are eliminated in Fig. 12(a). For all test functions considered in this paper, the FO algorithm obtains the optimal fitness value in most cases, proving the excellent exploration capability of the FO algorithm.

4.3.3. The balance of exploration and exploitation

Introducing the balancing factors $Curve_{down}$ and $Curve_{up}$ in the FO algorithm helps balance its E&E capability, as supported by Fig. 15. This figure presents the mean convergence curve of the FO algorithm, which is smooth, stable, and fast compared to the other competitor algorithms. Additionally, the changes in the labels of the individual community over the iterations are illustrated in Fig. 15. In the latter figure, each community is distinguished by a different color. As the iterations progress, the percentage of individuals in each community becomes more concentrated, indicating that most individuals eventually cluster into the same community to optimize the use of resources, influence, and status.

4.3.4. The convergence of the FO algorithm

However, a good algorithm should not converge too early, which leads it to a local optimum, nor should it converge too late, which is not beneficial for real engineering needs. Thus, the algorithm must converge firmly and asymptotically. As presented in Fig. 15, selected several representative functions from the four groups of functions for simulation comparison tests with 300 iterations and 30 search agents. The convergence performance of an algorithm is related to the balance of E&E. Nevertheless, to balance the E&E of algorithms, an adversarial mechanism must be introduced into the FO algorithm, which is the Protector' function. Specifically, the Protector receives information only from those benefit from the populations they belong. The competition and cooperation among the Fans-communities have also

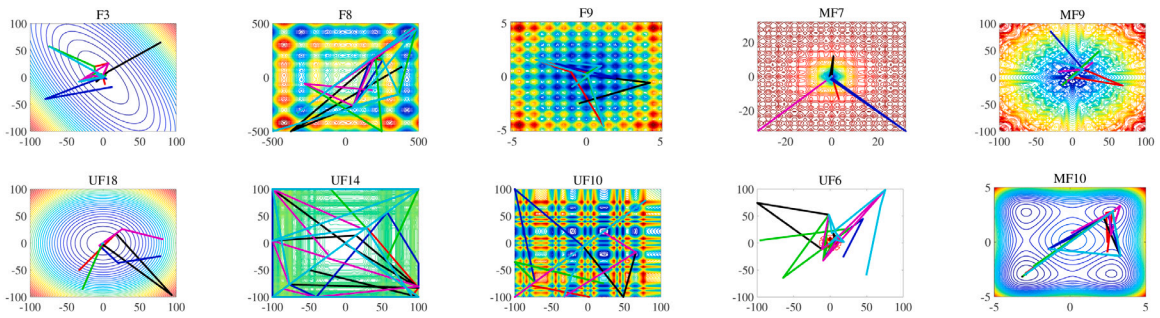

Fig. 13. 2-D trajectory update diagram of the benchmark mathematical functions, presenting first two dimensions of the iterative solution for six randomly selected search agents from the population.

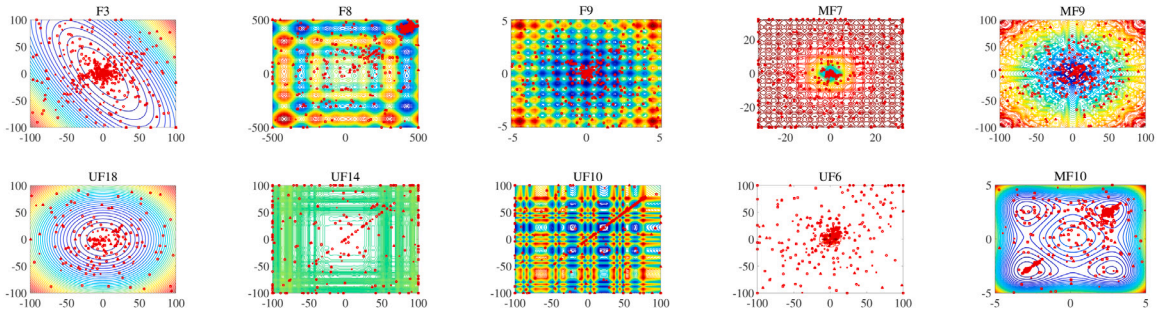

Fig. 14. 2-D position update diagram of the benchmark mathematical functions. To clearly articulate the population's movements, setting a simpler population consisting of three categories with three members.

Table 4
High-dimensional performance test results at F for FO algorithm.

Fn	Dim = 50			Dim = 1000		
	MIN	AVG	STD	MIN	AVG	STD
F1	1.1E−209	4.3E−151	2.3E−150	4.4E−205	9E−132	4.9E−131
F2	1.5E−104	2.38E−70	1.28E−69	0	3.8E−99	2.03E−98
F3	4.3E−146	1.54E−96	8.28E−96	0	6.27E−67	3.38E−66
F4	4E−106	1.45E−73	7.52E−73	0	2.54E−71	1.22E−70
F5	0	3.217649	12.03934	0	0	0
F6	0	0	0	0	0	0
F7	0.000233	0.000246	0.000157	0	3.7E−05	9.53E−05
F8	−20 949.1	−19 567.4	2504.69	0	−13 966.1	31 229.14
F9	0	0	0	0	0	0
F10	8.88E−16	8.88E−16	0	0	1.48E−16	3.31E−16
F11	0	0	0	0	0	0
F12	9.42E−33	9.42E−33	2.74E−48	0	3.93E−34	8.78E−34
F13	1.35E−32	1.35E−32	5.47E−48	0	2.25E−33	5.03E−33
F15	0.000307	0.000316	2.28E−05	0	5.27E−05	0.000118
F16	−1.03163	−1.03158	0.000126	0	−0.17192	0.384419
F17	0.397887	0.397896	2.3E−05	0	0.066315	0.148284
F18	3	3.004259	0.016619	0	0.500022	1.118084

contributed to the balance of E&E. Fig. 15 presents the changes graph of labels over iterations about agents belonging in the community, which visually depicts this conclusion. Throughout the iteration, the power of a Fans-community gradually grows to dominate ultimately. However, it simultaneously ensures the existence of other forces, thus avoiding dominance of one Fans-community, ensuring the diversity of population decisions, and avoiding falling into a local optimum while obtaining the near-optimal solution.

4.3.5. The performance to high-dimension

Based on the experimental data presented in Table A.12–Table A.14, the FO algorithm demonstrates exceptional performance in solving complex multi-dimensional problems. To further verify this observation, conducted a high-dimensional function test with 25 runs, and the results are presented in Tables 4–6. These results confirm that the FO algorithm preserves its excellent high-dimensional solving ability.

4.3.6. Statistical analysis of the FO algorithm

The Williams rank test is a paired test that seeks for significant differences between two samples (Derrac, García, Molina, & Herrera, 2011). In this section, the Williams-rank test (with 0.05 level) is performed on the FO algorithm and other 13 algorithms, to statistically analyze the FO algorithm. The corresponding experimental results are reported in Tables 7–10.

In the Williams rank test, the number of agents in each comparison algorithm is uniform (30) to ensure a strict and relatively stable analysis. This results in the actual number of agents involved in the computation among each algorithm for each iteration. According to the previous convergence data shown in Fig. 15, the characteristics of the algorithm are demonstrated over 300 iterations. Therefore, the iteration number during the Williams rank test is 300 for a fair evaluation. Furthermore, each algorithm is repeated 30 to avoid receiving results by chance. Finally, the ρ-values between the final results of each algorithm and the FO algorithm are calculated (see Tables 7–10).

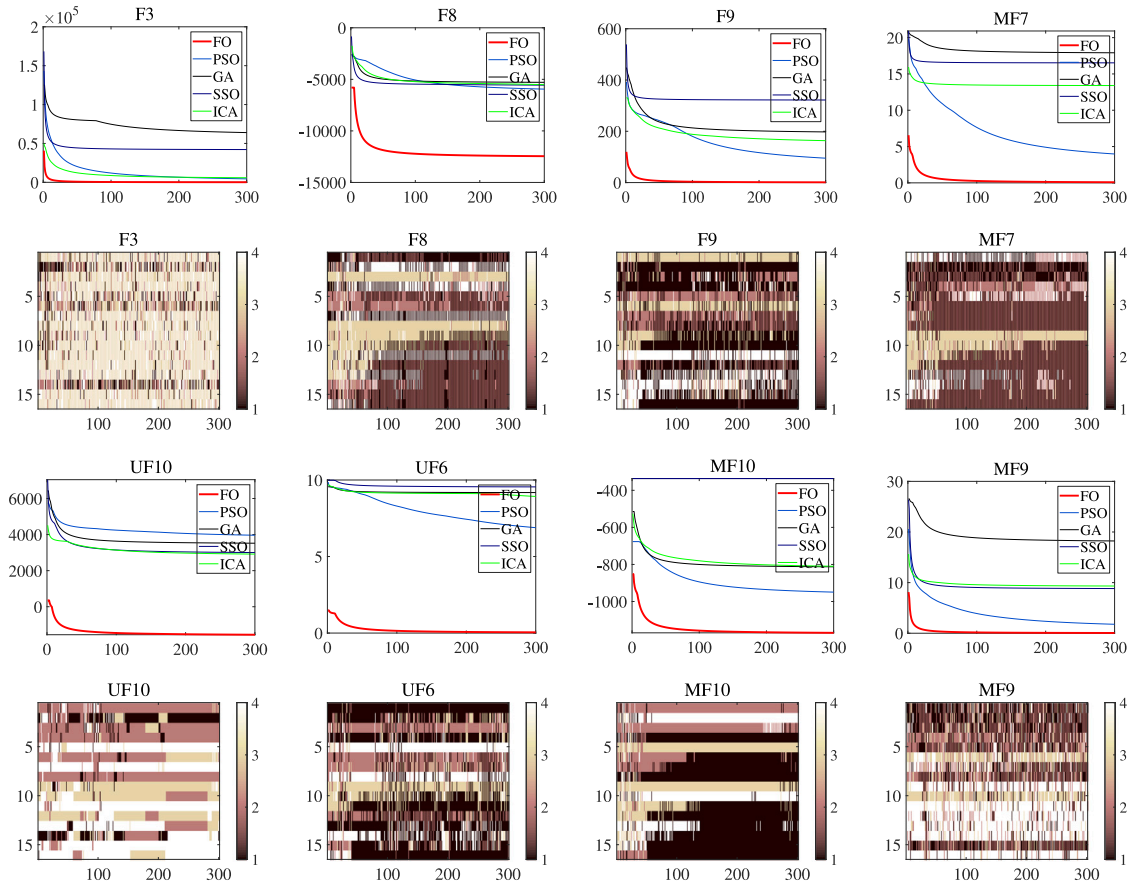

Fig. 15. Convergence of the FO algorithm over 300 iterations and the corresponding changes graph of labels about individual community over iterations.

Table 5
High-dimensional performance test results at MF for FO algorithm.

Fn	Dim = 50			Dim = 1000		
	Min	Avg	Std	Min	Avg	Std
MF1	1.27E−05	31.59621	52.36782	1.27E−05	1.27E−05	0
MF2	0	0	0	0	0	0
MF3	0.9	0.9	4.44E−16	0.9	0.9	0
MF4	5428.81	6170.875	1934.51	615 648.6	613 558.6	39 812.31
MF5	4.5E−106	1.19E−07	6.43E−07	3.4E−107	3.64E−84	7.28E−84
MF6	2.1E−17	8.96E−13	3.37E−12	6.92E−21	5.7E−18	1.14E−17
MF7	−8.9E−16	−8.9E−16	0	−8.9E−16	−8.9E−16	0
MF8	1	1	0	1	1	0
MF9	8.7E−104	8.4E−66	4.52E−65	3.4E−102	3.75E−66	7.49E−66
MF10	−1958.31	−1958.31	9.8E−06	−7833.23	−7833.23	9.09E−13
MF11	0	0	0	0	0	0
MF12	−1	−1	0	−1	−1	0
MF13	1.21E−20	1.13E−20	3.01E−21	3.47E−85	3.47E−85	3.1E−99
MF14	1.35E−32	1.35E−32	5.47E−48	1.35E−32	1.35E−32	0
MF15	9.42E−33	9.42E−33	2.74E−48	2.36E−33	2.36E−33	0
MF20	−1.03163	−1.03138	0.001201	−1.03163	−1.03053	0.002197
MF21	0.397887	0.397913	0.000138	0.397887	0.397887	0
MF22	3	3.002182	0.011573	3	3	9.16E−10

Among which "+" (ρ value < 0.05), "≈" ($\rho \approx 0.05$), and "−" ($\rho > 0.05$) are representatives of the sum number of functions for which the FO algorithm is statistically superior, close, and inferior to other comparison algorithms, respectively.

In summary, the Williams experiment proves that the FO algorithm is statistically superior to the competitor methods.

5. Real-world applications

The proposed FO algorithm aims to solve real engineering optimization problems, and its advantages have been demonstrated in **CEC2017** and **CEC2020**. Hence, this section evaluates the performance of the FO algorithm on eight well-known engineering functions and one real engineering problem. The main focus of this section is to evaluate the FO algorithm's performance in solving actual engineering problems by comparing it against other classic meta-algorithms. Each problem is tested with 300 iterations and 25 runs. In addition to these problems, the FO algorithm is also applied to solve the joint angle optimization problem of 9-DFSRA.

5.1. Speed reducer design(SRD)

The SRD optimization aims to reduce the weight of the reducer while maintaining the necessary performance (Mirjalili et al., 2017).

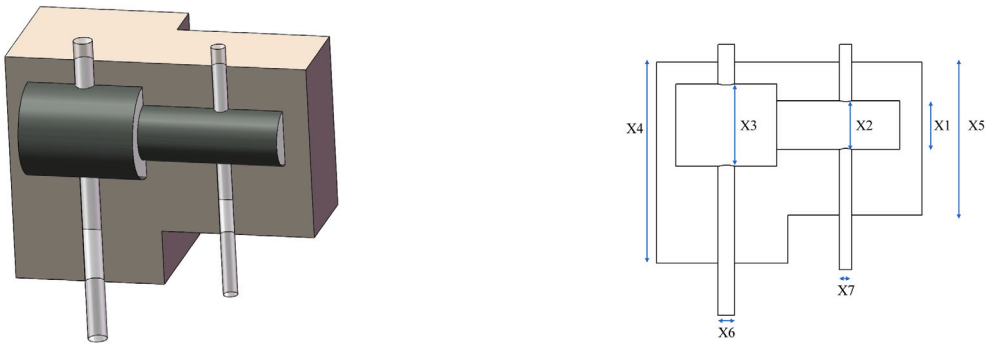

Fig. 16. Schematic diagram of SRD engineering problems.

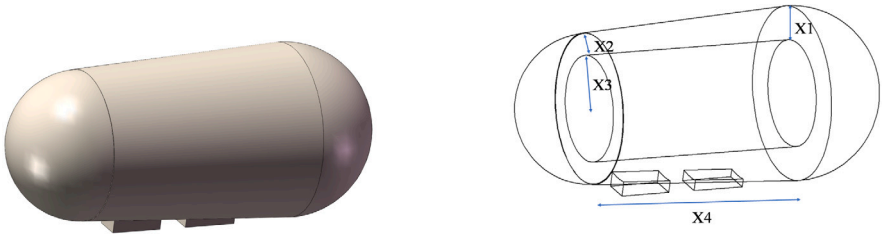

Fig. 17. The schematic diagram of PVD engineering problems.

Table 6
High-dimensional performance test results at UF for FO algorithm.

Fn	Dim = 50			Dim = 1000		
	Min	Avg	Std	Min	Avg	Std
UF1	7.7E−205	1.2E−139	6.6E−139	3.1E−209	9.9E−205	0
UF2	2.59E−51	5.39E−15	2.9E−14	2.58E−63	5.26E−46	1.05E−45
UF3	7E−75	0.643183	3.463644	6525.793	57346.26	69460.93
UF4	0	4.825091	14.47534	0	39.35326	78.70652
UF5	0	0	0	0	0	0
UF6	0	0	0	0	0	0
UF7	0	6.76E−21	3.64E−20	0	0	0
UF8	0	0	0	0	0	0
UF9	1.5E−32	1.5E−32	1.09E−47	1.5E−32	1.5E−32	0
UF10	−4025.82	−4012.84	41.34721	−16085.6	−15910.6	294.7217
UF11	1.1E−204	1.4E−146	7.5E−146	3.6E−205	3.5E−131	7E−131
UF12	1.33E−78	1.15E−14	6.19E−14	2.8E−118	6.32E−65	1.26E−64
UF13	8.88E−16	8.88E−16	0	8.88E−16	8.88E−16	0
UF14	−2450	−2450	0	−39800	−39800	0
UF15	0	1.07E−16	5.78E−16	0	3.55E−16	7.11E−16
UF16	0	0	0	0	0	0
UF17	0.004517	0.030781	0.069743	0.014333	0.024764	0.020624
UF18	0.5	0.5	3.36E−06	0.5	0.5	8.59E−08
UF19	1.2E−51	4.52E−36	1.72E−35	2.46E−50	2.87E−39	5.75E−39

The results of the optimization problem are reported in Table 11 demonstrating the optimal fitness values. Indeed, this figure highlights that the FO algorithm outperforms the other competitor algorithms regarding the MIN, MAX, AVG, and STD metrics, providing evidence of its superiority and stability for this problem. Additionally, the corresponding variable values for the optimal fitness values are presented in Table A.1. The schematic diagram of the SRD engineering problem is depicted in Fig. 16.

5.2. Pressure vessel design(PVD)

The TSD aims to control the weight of the spring while maintaining the shear stress, excitation frequency, and deformation rate of the spring. Table 12 demonstrates that the FO algorithm obtains optimal fitness values for this real engineering problem superior to the competitor algorithms (see Fig. 17). In addition, the values of the variables

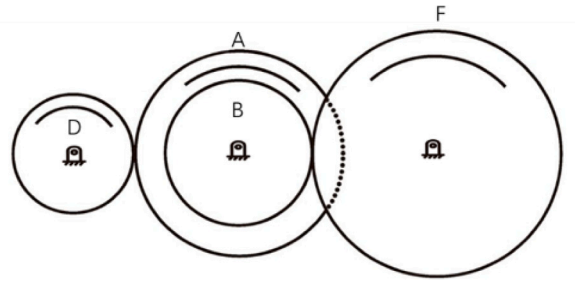

Fig. 18. Schematic diagram of DTD engineering problems.

corresponding to the obtained optimal fitness values are presented in Table A.2.

5.3. Dear train design(DTD)

The DTD is a discrete engineering problem that designed to reduce the gear rotation ratio (see Fig. 18). The result of the FO algorithm and other famous algorithms for DTD are illustrated in Table 13, and the corresponding to the obtained optimal fitness values are shown in Table A.3.

5.4. Cantilever beam design(CB)

As before, the CB design problem aims to control the weight of the workpiece. In particular the CB design problem in this paper contains 5 hollow squares (see Fig. 19). The statistical analysis of the FO algorithm and the competitor algorithms on this problem is shown in Table 14. In addition, the values of the variables corresponding to the obtained optimal fitness values are shown in Table A.4, and the optimal fitness obtained by the FO algorithm is still the best.

5.5. Three-bar truss design(TTD)

The schematic diagram of TTD engineering problems as illustrated in Fig. 20. The result of the FO algorithm and other famous algorithms

Table 7
ρ of Wilcoxon statistical experiment with 0.05 significance level on F.

Fn	EO	GA	SSA	SSO	WOA	PSO	SSO2	POA	ICA	CMAAES	COLSHADE	SASS
F1	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F2	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F3	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	**9.6E−02**	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F4	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F5	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**NAN**
F6	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F7	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.5E−10	3.0E−11	1.2E−12	2.1E−05	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F8	9.3E−12	9.3E−12	9.3E−12	9.3E−12	3.9E−06	9.3E−12	2.6E−13	1.6E−11	9.3E−12	9.3E−12	9.3E−12	9.3E−12
F9	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**8.1E−02**	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.5E−04
F10	1.2E−12	1.2E−12	1.2E−12	1.2E−12	3.9E−10	1.2E−12	1.7E−14	3.8E−10	1.2E−12	1.2E−12	1.2E−12	**6.4E−01**
F11	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**3.3E−01**	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F13	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	5.8E−11	**NAN**	1.7E−14	**NAN**	**NAN**	**NAN**	1.2E−12	1.2E−12
F15	2.6E−11	2.6E−11	2.6E−11	2.6E−11	2.6E−11	**1.0E−01**	1.0E−12	**7.4E−01**	**4.3E−01**	4.2E−04	2.6E−11	2.6E−11
F16	1.9E−11	1.9E−11	1.9E−11	1.9E−11	1.9E−11	**8.5E−02**	6.6E−13	2.4E−04	**1.0E−01**	1.8E−06	1.9E−11	1.9E−11
F17	3.0E−11	3.0E−11	3.0E−11	3.0E−11	4.6E−10	3.0E−11	1.2E−12	1.2E−06	3.0E−11	5.6E−10	3.0E−11	3.0E−11
F18	6.6E−11	3.0E−11	3.0E−11	3.0E−11	2.5E−04	3.0E−11	1.2E−12	4.0E−04	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F19	**2.6E−01**	3.0E−11	3.0E−11	3.0E−11	6.3E−06	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F20	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.3E−09	1.2E−12	1.7E−14	3.8E−10	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F21	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F22	3.0E−11	3.0E−11	3.0E−11	3.0E−11	2.5E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
F23	1.2E−09	3.1E−10	3.0E−12	2.4E−12	1.2E−09	1.2E−09	5.9E−11	1.2E−09	1.2E−09	1.2E−09	2.4E−12	1.2E−09
F24	1.2E−12	1.2E−12	1.2E−12	1.2E−12	4.2E−02	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F25	1.2E−12	1.2E−12	1.2E−12	1.2E−12	8.8E−07	1.2E−12	1.7E−14	6.6E−05	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F26	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F27	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12
F28	5.0E−08	2.0E−11	5.5E−11	2.2E−11	7.6E−08	1.2E−02	4.4E−10	**6.2E−01**	**1.9E−01**	4.5E−06	1.8E−11	3.3E−11
+	27	28	28	28	26	24	28	22	24	27	28	26
≈	0	0	0	0	0	2	0	3	1	1	0	1
−	1	0	0	0	2	2	0	3	3	0	0	1

Table 8
ρ of Wilcoxon statistical experiment with 0.05 significance level on MF.

Fn	EO	GA	SSA	SSO	WOA	PSO	SSO2	POA	ICA	CMAAES	COLSHADE	SASS
MF1	1.4E−11	1.4E−11	1.4E−11	1.4E−11	3.4E−04	1.4E−11	4.4E−13	1.4E−11	1.4E−11	1.4E−11	1.4E−11	1.4E−11
MF2	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**8.1E−02**	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.2E−12
MF3	1.7E−12	1.7E−12	1.7E−12	1.7E−12	1.2E−03	1.7E−12	2.7E−14	**3.3E−01**	1.7E−12	1.7E−12	1.7E−12	1.7E−12
MF4	2.9E−11	2.9E−11	2.9E−11	2.9E−11	5.8E−11	2.9E−11	1.1E−12	2.9E−11	2.8E−11	2.9E−11	2.9E−11	2.9E−11
MF5	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
MF6	4.5E−07	1.8E−11	1.4E−10	1.8E−11	9.7E−06	1.1E−05	1.3E−07	**3.5E−01**	1.1E−05	1.1E−05	2.0E−11	1.8E−11
MF7	3.0E−03	1.4E−11	1.4E−11	1.4E−11	1.5E−04	2.9E−04	4.5E−13	1.1E−02	5.6E−05	4.8E−09	1.4E−11	1.4E−11
MF8	3.1E−08	6.0E−08	7.4E−06	3.5E−11	3.7E−04	2.8E−04	1.8E−05	**2.7E−01**	2.3E−02	4.2E−02	1.2E−10	1.9E−07
MF9	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**NAN**	1.7E−14	**NAN**	**NAN**	**NAN**	1.2E−12	1.2E−12
MF10	2.6E−11	2.6E−11	2.6E−11	2.6E−11	2.6E−11	**5.5E−02**	9.9E−13	**3.6E−01**	**1.7E−01**	1.9E−04	2.6E−11	2.6E−11
MF11	2.0E−11	2.0E−11	2.0E−11	2.0E−11	2.0E−11	2.0E−06	7.2E−13	**1.3E−01**	1.3E−02	1.2E−08	2.0E−11	2.0E−11
MF12	3.6E−04	4.2E−10	7.4E−10	3.0E−11	4.1E−05	9.3E−05	**5.9E−02**	3.0E−02	3.0E−04	9.3E−05	5.0E−11	3.0E−03
MF13	4.8E−07	6.5E−12	7.2E−12	6.5E−12	3.8E−06	1.1E−02	3.6E−09	**1.0E−01**	1.1E−02	1.1E−02	6.5E−12	6.5E−12
MF14	1.4E−11	7.8E−11	8.6E−11	3.4E−11	3.4E−10	4.8E−05	6.1E−14	**1.8E−01**	2.2E−07	3.2E−12	3.0E−11	3.9E−12
MF15	4.1E−12	4.1E−12	4.1E−12	4.1E−12	4.1E−12	8.5E−08	8.7E−14	6.4E−03	3.8E−08	4.1E−12	4.1E−12	4.1E−12
MF16	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	5.5E−11	1.2E−12	6.0E−11	9.4E−06	3.0E−11	3.0E−11	3.0E−11
MF17	5.2E−06	1.1E−09	8.2E−08	4.5E−11	7.9E−06	9.2E−04	9.4E−09	**6.0E−01**	4.9E−02	7.3E−07	4.7E−10	3.5E−10
+	17	17	17	17	16	15	16	6	13	15	17	16
≈	0	0	0	0	0	1	0	3	3	2	0	1
−	0	0	0	0	1	1	1	8	1	0	0	0

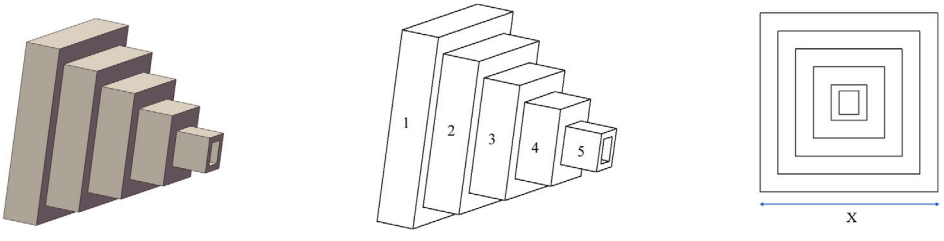

Fig. 19. Schematic diagram of CB engineering problems.

Table 9
ρ of Wilcoxon statistical experiment with 0.05 significance level on UF.

Fn	EO	GA	SSA	SSO	WOA	PSO	SSO2	POA	ICA	CMAAES	COLSHADE	SASS
UF1	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
UF2	5.5E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
UF3	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	8.1E−10	3.0E−11	3.0E−11	3.0E−11	3.0E−11
UF4	1.7E−12	1.7E−12	1.7E−12	1.7E−12	1.7E−12	1.7E−12	2.7E−14	1.7E−12	1.7E−12	1.7E−12	1.7E−12	1.7E−12
UF5	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**1.6E−01**	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.2E−12
UF6	1.2E−12	1.2E−12	1.2E−12	1.2E−12	4.2E−02	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.2E−12
UF7	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**3.3E−01**	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.2E−12
UF8	1.2E−12	1.2E−12	1.2E−12	1.2E−12	2.2E−02	1.2E−12	1.7E−14	**NAN**	1.2E−12	1.2E−12	1.2E−12	1.2E−12
UF9	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.7E−14	1.2E−12	1.2E−12	1.2E−12	1.2E−12	1.2E−12
UF10	3.0E−11	3.0E−11	3.0E−11	3.0E−11	2.0E−10	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
UF11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
UF12	1.2E−05	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.7E−02	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
UF13	1.2E−12	1.2E−12	1.2E−12	1.2E−12	4.5E−09	1.2E−12	1.7E−14	6.1E−11	1.7E−14	1.2E−12	1.2E−12	1.2E−12
UF14	1.7E−12	1.7E−12	1.7E−12	1.2E−13	1.2E−13	1.7E−12	2.7E−14	1.5E−12	1.6E−12	1.7E−12	1.7E−12	1.7E−12
UF15	2.9E−04	1.2E−12	1.2E−12	1.2E−12	**NAN**	**NAN**	1.7E−14	**NAN**	**NAN**	1.2E−12	1.2E−12	1.2E−12
UF16	1.2E−12	1.2E−12	1.2E−12	1.2E−12	**NAN**	1.2E−12	1.7E−14	**NAN**	**NAN**	1.2E−12	1.2E−12	1.2E−12
UF17	3.0E−11	3.0E−11	3.0E−11	3.0E−11	5.0E−11	3.7E−11	1.2E−12	3.0E−11	3.0E−11	4.1E−11	3.0E−11	3.0E−11
UF18	1.7E−07	3.0E−11	3.0E−11	3.0E−11	3.3E−11	**6.6E−01**	1.2E−12	1.0E−06	3.0E−11	1.1E−06	3.0E−11	3.0E−11
UF19	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11	1.2E−12	3.0E−11	3.0E−11	3.0E−11	3.0E−11	3.0E−11
+	19	19	19	19	16	17	19	13	17	19	19	19
≈	0	0	0	0	2	1	0	6	2	0	0	0
−	0	0	0	0	1	1	0	0	0	0	0	0

Table 10
ρ of Wilcoxon statistical experiment with 0.05 significance level on CEC2020.

Fn	EO	GA	SSA	SSO	WOA	PSO	SSO2	POA	ICA	CMAAES	COLSHADE	SASS
CEC1	NAN	1.2E−12	NAN	NAN	NAN	NAN	1.7E−14	NAN	**3.3E−01**	1.1E−12	1.2E−12	NAN
CEC2	NAN	1.2E−12	NAN	NAN	NAN	NAN	1.7E−14	NAN	6.4E−04	1.2E−12	1.2E−12	1.2E−12
CEC3	2.9E−05	1.2E−12	NAN	1.3E−04	**3.3E−01**	NAN	1.7E−14	NAN	3.3E−07	1.7E−14	1.2E−12	4.6E−12
CEC4	NAN	1.2E−12	NAN	NAN	NAN	NAN	1.7E−14	NAN	NAN	1.7E−14	1.2E−12	NAN
CEC5	4.2E−02	1.2E−12	NAN	NAN	NAN	NAN	1.7E−14	NAN	1.2E−07	1.7E−14	1.2E−12	**1.6E−01**
CEC6	1.7E−14	1.2E−12	NAN	NAN	NAN	NAN	1.7E−14	NAN	NAN	1.2E−12	1.2E−12	NAN
CEC7	8.4E−10	2.4E−12	**9.9E−01**	**1.6E−01**	7.6E−05	**1.1E−01**	4.2E−14	**5.7E−01**	1.1E−11	2.4E−12	2.4E−12	**1.6E−01**
CEC8	2.7E−03	1.2E−12	NAN	NAN	NAN	5.4E−09	1.7E−14	NAN	NAN	8.7E−14	1.2E−12	2.9E−13
CEC9	**1.6E−01**	1.2E−12	NAN	NAN	NAN	NAN	1.7E−14	NAN	NAN	1.2E−13	1.2E−12	9.7E−06
CEC10	2.2E−02	1.2E−12	1.2E−12	1.2E−12	**3.3E−01**	NAN	1.7E−14	NAN	NAN	1.2E−12	1.2E−12	1.2E−12
+	7	10	1	2	1	1	0	0	4	10	10	5
≈	2	0	8	7	7	8	0	9	5	0	0	3
−	1	0	1	1	2	1	0	1	1	0	0	2

Table 11
Results of a comparative experiment for SRD.

Algorithms	MIN	MAX	AVG	STD
FO	2994.4245	2994.4245	2994.4245	0
DELC (Wang & Li, 2010)	2994.4711	2994.4711	2994.4711	0
DEDS (Zhang, Luo, & Wang, 2008)	2994.4711	2994.4711	2994.4711	0
EO	2995.0467	3007.1329	2999.7024	3.2718
GA	39 950.9847	40 367.4473	40 106.1684	115.6725
HEAA (Wang, Cai, Zhou, & Fan, 2009)	2994.4991	2994.7523	2994.6134	0.07
ICA	2996.1554	3365.0099	3154.8276	116.1677
POA	2994.4793	3006.3225	2998.3731	4.009
PSO	2994.4247	3007.3899	2996.8133	4.6504
PSO-DE (Liu, Cai, & Wang, 2010)	2996.3482	2996.3482	2996.3482	0
SSA	3040.0525	3222.864	3146.3301	51.9142
SSO	3148.9061	3233.5467	3191.755	19.4217
SSO2	3042.3611	3042.3611	3042.3611	0
WCA (Eskandar, Sadollah, Bahreininejad, & Hamdi, 2012)	2994.4711	2994.5056	2994.4744	0.0074
WOA	3005.7058	3351.1288	3121.3475	116.6116
CMAAES	2994.4793	40 367.4473	2996.3482	0
COLSHADE	3148.9061	3233.5467	3191.7550	1133.84708
SASS	3005.7058	3351.1288	3042.3611	1127.716829

for TTD are summarized in Table 16. In addition, the values of the variables corresponding to the obtained optimal fitness values are summarized in Table A.6. The optimal fitness obtained by the FO algorithm in this section is still superior to the competitor algorithms.

5.6. I-beam design(IBD)

The IBD problem is a typical engineering design problem that aims to minimize vertical deflection (see Fig. 21). The result of the FO

Table 12
Results of a comparative experiment for PVD.

Algorithms	MIN	MAX	AVG	STD
FO algorithm	6059.71434	7635.83363	6590.44258	559.18223
CDE (Huang, Wang, & He, 2007)	6059.734	6371.0455	6085.2303	43.013
CPSO (Krohling & Coelho, 2006)	6061.0777	6363.8041	6147.1332	86.45
EO	6473.72634	8028.54919	7138.79879	394.92875
GA	6881.49529	8930.50685	7956.71808	696.20959
GA3 (Coello & Montes, 2002)	6059.9463	6469.322	6177.2533	130.9297
G-QPSO (dos Santos Coelho, 2010)	6059.7208	7544.4925	6440.3786	448.4711
GSA (Rashedi, Nezamabadi-pour, & Saryazdi, 2009)	8538.8359	NAN	8932.95	683.5475
ICA	6069.58733	18 152.2479	7627.70101	2504.04519
POA	6059.71434	6820.41008	6170.34311	187.39191
PSO	6059.71434	7046.5931	6402.07166	304.77787
SSA	6259.53792	8684.349	7364.07486	552.31436
SSO	7777.67201	11 727.2271	9216.48008	1310.00677
SSO2	6636.36818	6636.36818	6636.36818	0
WOA	6095.11796	9513.50799	7389.07204	845.07313
CMAAES	65 535	6881.49529	6820.41008	845.07313
COLSHADE	6635.64075	8738.877036	6732.64075	3039.674331
SASS	6553.5	85 555.55254	7138.79879	28 529.21095

Table 13
Results of a comparative experiment for DTD.

Algorithms	MIN	MAX	AVG	STD
FO	2.70E−12	1.03E−08	1.65E−09	2.29E−09
EO	5.26E−10	8.75E−09	3.31E−09	2.61E−09
GA	2.36E−09	3.21E−06	4.74E−07	8.01E−07
ICA	2.70E−12	3.10E−03	1.24E−04	6.21E−04
POA	2.70E−12	9.92E−10	1.18E−10	2.52E−10
PSO	2.70E−12	1.12E−08	1.94E−09	2.52E−09
SSA	2.70E−12	5.81E−08	1.16E−08	1.49E−08
SSO	2.36E−09	3.01E−07	5.37E−08	7.53E−08
WOA	2.70E−12	2.73E−08	2.44E−09	5.23E−09
CMAAES	2.73E−08	3.95E−03	5.28E−04	1.07E−03
COLSHADE	2.31E−11	5.64E−07	5.91E−08	1.53E−07
SASS	2.70E−12	9.92E−10	1.16E−10	8.01E−07

Table 14
Results of a comparative experiment for CB.

Algorithms	MIN	MAX	AVG	STD
FO	1.339958	1.340245	1.340019	0.000075
EO	1.340364	1.342731	1.341286	0.000536
GA	1.801506	3.807631	2.720524	0.571762
ICA	1.352656	6.956951	1.707699	1.101951
POA	1.339961	1.340109	1.340021	0.000047
PSO	1.344532	1.503693	1.385448	0.045978
SSA	1.838851	3.446381	2.594040	0.439546
SSO	1.844922	7.363249	3.731013	1.191943
SSO2	1.340104	1.340104	1.340104	0.000000
WOA	1.394056	2.353862	1.701536	0.254881
CMAAES	2.161719	7.686716	5.347781	1.270697
COLSHADE	1.621929	7.276888	4.696556	0.000047
SASS	1.344532	1.508564	1.385564	0.439546

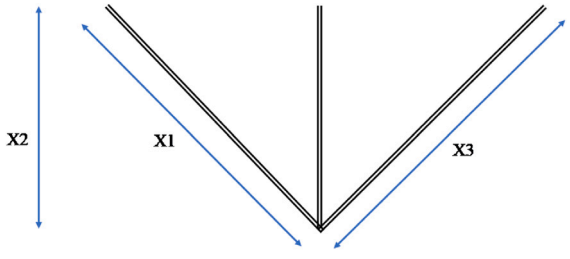

Fig. 20. The schematic diagram of TTD engineering problems.

algorithm and other famous algorithms for IBD is shown in Table 15. In addition, the values of the variables corresponding to the obtained optimal fitness values are summarized in Table A.10. As with the previous real engineering problems, the FO algorithm achieves superior fitness values is the best.

Table 15
Results of a comparative experiment for IBD.

Algorithms	MIN	MAX	AVG	STD
FO	0.01307412	0.01307418	0.01307412	0.00000001
EO	0.01307567	0.01327566	0.01313752	0.00004737
ICA	0.01307412	0.01465404	0.01320946	0.00043625
POA	0.01307412	0.01307412	0.01307412	0.00000000
PSO	0.01307412	0.01328854	0.01308723	0.00004331
SSA	0.01307989	0.01371589	0.01332992	0.00017994
SSO	0.01307452	0.01329508	0.01313340	0.00005786
SSO2	0.01342326	0.01342326	0.01342326	0.00000000
WOA	0.01310094	0.01861894	0.01409752	0.00121272
WOA	1.394056	2.353862	1.701536	0.254881
CMAAES	0.01307802	0.21253698	0.05124319	0.05806017
COLSHADE	0.01307412	0.01348562	0.01258475	0.05806415
SASS	0.01308559	0.01586741	0.01318569	0.00043625

Table 16
Results of a comparative experiment for TTD.

Algorithms	MIN	MAX	AVG	STD
FO	263.89584	263.89584	263.89584	0.00000
EO	263.89589	263.90487	263.89739	0.00213
GA	263.90544	264.09936	263.98198	0.05189
ICA	263.89584	263.89585	263.89584	0.00000
POA	263.89584	263.89585	263.89584	0.00000
PSO	263.89584	263.89630	263.89587	0.00009
SSA	263.89584	263.96750	263.91523	0.01554
SSO	263.90586	282.84271	266.49388	6.17213
SSO2	263.89942	263.89942	263.89942	0.00000
WOA	263.89586	264.39264	264.01899	0.11016
CMAAES	263.89584	263.96750	263.91523	0
COLSHADE	263.89584	263.89585	263.89584	0.00009
SASS	263.89589	263.90487	263.89739	6.17213

5.7. Tension/compression spring design(TSD)

The TSD is a classic engineering problem that aims to control the weight of the spring while maintaining the shear stress, excitation frequency, and deformation rate of the spring (see Fig. 22). Table 17 demonstrates that the FO algorithm obtains optimal fitness values for this real engineering problem superior to other good algorithms. In addition, the values of the variables corresponding to the obtained optimal fitness values are summarized in Table A.7.

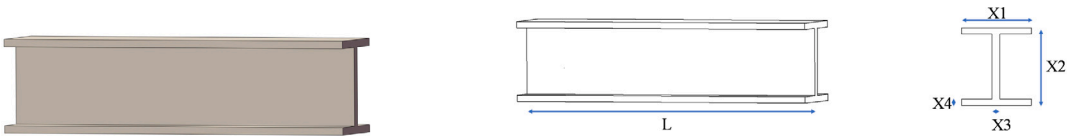

Fig. 21. Schematic diagram of IBD engineering problems.

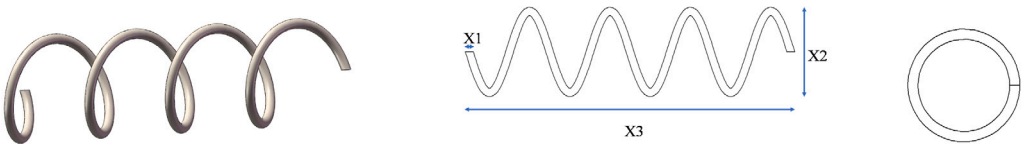

Fig. 22. Schematic diagram of TSD engineering problems.

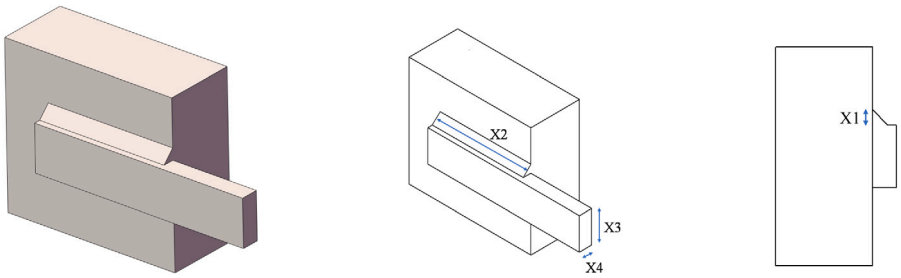

Fig. 23. Schematic diagram of WBD engineering problems.

Table 17
Results of a comparative experiment for TSD.

Algorithms	MIN	MAX	AVG	STD
FO	0.012665	0.012788	0.012672	0.000024
CPSO (Krohling & Coelho, 2006)	0.012700	0.012900	0.012700	0.000500
DE (Lampinen & Storn, 2004)	0.012700	0.012800	0.012700	0.000000
DEDS (Zhang et al., 2008)	0.012700	0.012700	0.012700	0.000000
DELC (Wang & Li, 2010)	0.012700	0.012700	0.012700	0.000000
EO	0.012833	0.016840	0.014180	0.000956
GA	0.021585	0.056329	0.031873	0.008767
GA3 (Coello & Montes, 2002)	0.012700	0.013000	0.012700	0.000100
GSA (Rashedi et al., 2009)	NAN	NAN	0.013600	0.002600
HEAA (Wang et al., 2009)	0.012700	0.012700	0.012700	0.000000
ICA	0.012673	2.793543	0.331999	0.873704
MADE (Hamza, Abderazek, Lakhdar, Ferhat, & Yıldız, 2018)	0.012700	0.012700	0.012700	0.000000
PO	0.012700	0.012800	0.012700	0.000000
POA	0.012665	0.013191	0.012796	0.000161
PSO	0.012665	0.017147	0.013593	0.001267
SSA	0.012676	0.025806	0.015181	0.003249
SSO	0.013116	0.022299	0.015107	0.003061
SSO2	0.013732	0.013732	0.013732	0.000000
WCA (Eskandar et al., 2012)	0.012700	0.013000	0.012700	0.000100
WOA	0.012670	0.017780	0.014015	0.001645
CMAAES	0.013116	0.022299	0.013116	0.003249
COLSHADE	0.012700	0.021752	0.014012	0.006654
SASS	0.012665	1.837801	0.247470	0.456710

5.8. Welded beam design(WBD)

This section considers the WBD engineering problem, which involves designing a robust welded beam frame with four variables. The schematic diagram of the WBD engineering problem is illustrated in Fig. 23. Results obtained by applying the FO algorithm and the competitor algorithms to the WBD problem are summarized in Table 18, and the corresponding variable values are presented in Table A.8. The analysis of the data in Table 18 shows that the FO algorithm outperforms the other algorithms. In contrast, the GA algorithm, on the other hand, converges to a poor local optimum when solving this problem.

5.9. Inverse kinematic design of space robotic arm by FO

This section investigates the joint angle optimization problem of a 9-DFSRA in the Denavit–Hartenberg (DH) coordinate system. The inverse kinematic solution of a 9-DFSRA considers solving a set of joint angles of the arm $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9]$, while the set of 6-degree-of-freedom coordinates is known. This section defines the 6-degree-of-freedom coordinates as $X_d = [0.9168, 0.8816, 0.6864, -0.8127, 0.6331, 2.5851]$ where the first three are end positions $[p_x, \ p_y, \ p_z]$ and the last three are end poses $[r_x, \ r_y, \ r_z]$. The conversion relationship

Table 18
Results of a comparative experiment for WBD.

Algorithms	MIN	MAX	AVG	STD
FO	1.724852	1.805447	1.729501	0.016956
CAEP (Coello, 2004)	1.724852	3.179709	1.971809	0.443
CGWO (Kohli & Arora, 2018)	1.72545	2.4357	2.4289	1.3578
CPSO (Krohling & Coelho, 2006)	1.728024	1.782143	1.748831	0.0129
EO	1.744786	2.168084	1.841524	0.115083
GA	33 435.8042	38 472.983	35 343.3436	1343.24421
GA3 (Coello & Montes, 2002)	1.728226	1.993408	1.792654	0.0747
ICA	1.755222	3.991568	2.208728	0.46298
MADE (Hamza et al., 2018)	1.724852	1.724852	1.724852	9.6 E−16
PO	1.724851	1.724852	1.724851	2.53 E−07
POA	1.724852	1.730017	1.725561	0.001208
PSO	1.728264	2.245762	1.891471	0.139316
SSA	1.86904	2.434105	2.09035	0.165961
SSO	2.030702	2.412085	2.248039	0.130643
SSO2	1.849693	1.849693	1.849693	0
WCA (Eskandar et al., 2012)	1.724856	1.744697	1.726427	0.00429
WOA	1.836094	5.423111	2.667605	0.938704
CMAAES	1.869040	190.655350	39.625840	65 535
COLSHADE	2.224291	5.107907	2.283711	1.265740
SASS	3.045023	8.503198	3.575820	2.315139

between θ and X_d for the DH coordinate system is as follows:

$$T_i^{i-1} = Rot_{z_i-1}(\theta_i)Trans_{z_i-1}(d_i)Trans_{x_i}(a_i)Trans_{x_i-1}(\alpha_i)$$
$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & \cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$T_n^0 = T_1^0 T_2^1 T_3^2 T_4^3 \cdots T_n^{n-1} = \begin{bmatrix} [R] & \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \\ [0] & 1 \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \mathrm{EUL}(R) \tag{20}$$

where, R is the rotation matrix. Table A.5 reports the details on the 9-DFSRA. Thus the definition of the fitness function is as follows :

$$f = \arg\min(X_\theta - X\_d) \tag{21}$$

$$X_\theta = [p_x, p_y, p_z, r_x, r_y, r_z] \tag{22}$$

For the design problem of a 9-DFSRA in the DH coordinate system, the experiment conducted 25 times per algorithm and for 300 iterations. The corresponding minimum, maximum, intermediate solution, and standard deviation values per algorithm are presented in Table 19. To provide a better visual understanding of the problem, present a 3-D simulation of a 9-DFSRA in Fig. 24. The results indicate that the FO algorithm outperforms the competitor algorithms in solving this complex real-world application problem of the 9-DFSRA, highlighting the superiority and competitiveness of the proposed method.

Table 19
Comparison with the previous meta-heuristic algorithms for Inverse kinematic design of space robotic arm.

Algorithm	MIN	MAX	AVG	STD
FO	9.05E−13	0.395621	0.395621	0.325872
EO	0.785842	1.099452	1.099452	0.117448
FA	0.057138	0.462029	0.462029	0.186534
GA	1.282408	1.711929	1.711929	0.209019
ICA	0.927603	1.530292	1.530292	0.319898
POA	0.024126	0.449627	0.449627	0.290893
PSO	0.066125	0.922649	0.922649	0.384243
SSA	0.705819	1.34762	1.34762	0.297984
SSO	2.040608	2.246815	2.246815	0.119891
SSO2	0.177369	0.177369	0.177369	2.85E−17
WOA	0.624604	1.378495	1.378495	0.328872

6. Conclusion

This paper is inspired by the resource competition in the entertainment domain and proposes a novel FO algorithm that integrates a **Cooperation–Competition Mechanism** into the FO algorithm to simulate the game between communities in the real world, which leads to a non-zero-sum game outcome. Moreover, we increase the diversity of roles in the FO algorithm to enhance the individuals' flexibility and avoid local optima during iterations.

To evaluate the algorithm's performance, we compare the FO algorithm with 12 meta-heuristics methods on four groups of benchmark functions: uni-modal, multi-modal, and fixed-dimensional. The experimental results demonstrate that the FO algorithm outperforms the competitor algorithms, especially on multi-modal and fixed-dimensional benchmark functions. Moreover, the Williams test confirms that the FO algorithm is statistically superior to competitor algorithms.

Furthermore, we test the FO algorithm's real-world applicability by solving eight engineering problems. In this trial, the FO algorithm demonstrates the best overall accuracy and stability compared to the 12 comparison algorithms. Finally, we also apply the FO algorithm to solve the inverse kinematic solution of the 9-DFSRA. Overall, all trials reveal that the FO algorithm outperforms the competitor algorithms in solving real-world problems.

In conclusion, the FO algorithm is a promising meta-heuristic algorithm that effectively handles complex real-world optimization problems. Indeed, this study demonstrates the superiority and stability of the FO algorithm over current algorithms, indicating its potential in practical applications.

Limitations: From an academic perspective, it is essential to acknowledge the limitations of the FO algorithm. One limitation is the refined definition of the Deviator, which sacrifices some physical meaning to reduce the computational burden. Additionally, the multiple roles of the FO algorithm increase the algorithm's complexity while improving its updates. Another limitation is the initial scale matrix of the roles, which requires seven adjustable parameters and makes the FO algorithm more complex.

Future research will investigate the initial role proportion matrix to improve the FO algorithm's performance. We aim to refine and simplify the input parameters of the algorithm and investigate variants that may be more applicable to complex practical engineering problems. These findings are inspired by the excellent performance of the FO algorithm on complex problems and will be essential to advance its practical use.

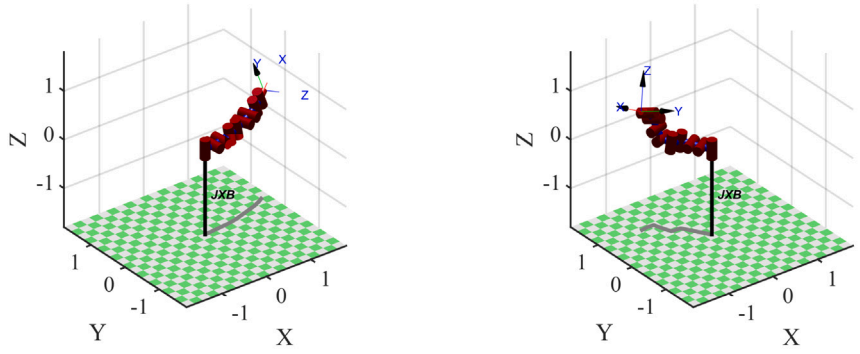

Fig. 24. 3-D simulation of a 9-DFSRA (a) is the initial state of the robot arm with each joint angle is 0.1 rad; (b) is the final state of the robot arm.

Funding

This research was funded by: Heilongjiang Province Applied Technology Research and Development Program (Grant No. GA20A401).

CRediT authorship contribution statement

Xiaofei Wang: Conceptualization, Methodology, Software, Validation, Data curation, Writing – original draft. **Jiazhong Xu:** Visualization, Investigation, Writing – review & editing, Supervision. **Cheng Huang:** Writing – review & editing.

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.

Data availability

No data was used for the research described in the article

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2023.120242.

References

Abraham, J. B., Wahyunggoro, O., & Setiawan, N. A. (2019). An effective quantum inspired genetic algorithm for continuous multiobjective optimization. In *2019 5th international conference on science in information technology (ICSITech* (pp. 161–166). Ieee, http://dx.doi.org/10.1109/ICSITech46713.2019.8987578, URL: https://www.webofscience.com/wos/wosccc/full-record/WOS:000542989200028.

Al Salami, N. M. (2009). Ant colony optimization algorithm. *UbiCC Journal*, *4*(3), 823–826. http://dx.doi.org/10.3724/SP.J.1238.2010.00585.

Asghari, K., Masdari, M., Gharehchopogh, F. S., & Saneifard, R. (2021). Multi-swarm and chaotic whale-particle swarm optimization algorithm with a selection method based on roulette wheel. *Expert Systems*, *38*(8), Article e12779. http://dx.doi.org/10.1111/exsy.12779.

Askari, Q., Saeed, M., & Younas, I. (2020). Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Systems with Applications*, *161*, http://dx.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. http://dx.doi.org/10.1016/j.knosys.2020.105709, URL: https://www.sciencedirect.com/science/article/pii/S0950705120301350.

Atashpaz-Gargari, E., & Lucas, C. (2008). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *IEEE congress on evolutionary computation* (pp. 4661–4667). http://dx.doi.org/10.1109/CEC.2007.4425083.

Birbil, S. I., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, *25*(3), p.263–282. http://dx.doi.org/10.1023/A:1022452626305.

Coello, C. (2004). Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization*, *36*(2), 219–236. http://dx.doi.org/10.1080/03052150410001647966, URL: https://www.tandfonline.com/doi/abs/10.1080/03052150410001647966.

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. http://dx.doi.org/10.1016/S1474-0346(02)00011-3.

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. http://dx.doi.org/10.1016/j.swevo.2011.02.002, URL: https://www.sciencedirect.com/science/article/pii/S2210650211000034.

Digalakis, J. G., & Margaritis, K. G. (2001). On benchmarking functions for genetic algorithms. *International Journal of Computational Methods*, *77*(4), 481–506. http://dx.doi.org/10.1080/00207160108805080.

Duan, Y., Chen, N., Chang, L., Ni, Y., Kumar, S. V. N. S., & Zhang, P. (2022). CAPSO: Chaos Adaptive Particle Swarm Optimization Algorithm. *Ieee Access*, *10*, 29393–29405. http://dx.doi.org/10.1109/ACCESS.2022.3158666.

Emami, H., & Derakhshan, F. (2015). Election algorithm: A new socio-politically inspired strategy. *AI Communications*, *28*, 591–603. http://dx.doi.org/10.3233/AIC-140652, URL: https://content.iospress.com/articles/ai-communications/aic652.

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*, *110–111*, 151–166. http://dx.doi.org/10.1016/j.compstruc.2012.07.010, URL: https://www.sciencedirect.com/science/article/pii/S0045794912001770.

Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, *191*, Article 105190. http://dx.doi.org/10.1016/j.knosys.2019.105190, URL: https://www.sciencedirect.com/science/article/pii/S0950705119305295.

Formato, R. A. (2009). Central force optimization: A new deterministic gradient-like optimization metaheuristic. *OPSEARCH*, *46*(1), 25–51. http://dx.doi.org/10.1007/s12597-009-0003-4.

Guo, S., Zhang, T., Song, Y., & Qian, F. (2018). Color feature-based object tracking through particle swarm optimization with improved inertia weight. *Sensors*, *18*(4), 1292. http://dx.doi.org/10.3390/s18041292.

Gurrola-Ramos, J., Aguirre, A. H., & Cedeño, O. D. (2020). COLSHADE for real-world single-objective constrained optimization problems. (pp. 1–8). http://dx.doi.org/10.1109/CEC48606.2020.9185583.

Hamza, F., Abderazek, H., Lakhdar, S., Ferhat, D., & Yıldız, A. R. (2018). Optimum design of cam-roller follower mechanism using a new evolutionary algorithm. *International Journal of Advanced Manufacturing Technology*, *99*(5), 1267–1282. http://dx.doi.org/10.1007/s00170-018-2543-3, URL: https://link.springer.com/content/pdf/10.1007/s00170-018-2543-3.pdf.

Hashmi, A., Goel, N., Goel, S., & Gupta, D. (2013). Firefly algorithm for unconstrained optimization. *IOSR Journal of Computer Engineering (IOSR-JCE)*, http://dx.doi.org/10.9790/0661-1117578, e-ISSN, 2278–0661.

Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Applied Soft Computing*, *60*, 115–134. http://dx.doi.org/10.1016/j.asoc.2017.06.044.

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73. http://dx.doi.org/10.1038/scientificamerican0792-66.

Huang, F.-z., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, *186*(1), 340–356. http://dx.doi.org/10.1016/j.amc.2006.07.105.

Hussain, A., Polycarpou, M. M., & Yao, X. (2021). Conference report on 2020 IEEE world congress on computational intelligence (IEEE WCCI 2020) [Conference reports]. *IEEE Computational Intelligence Magazine*, *16*(1), 15–98. http://dx.doi.org/10.1109/MCI.2020.3039042.

Kabir, M. M. J., Xu, S., Kang, B. H., & Zhao, Z. (2017). A new multiple seeds based genetic algorithm for discovering a set of interesting Boolean association rules. *Expert Systems with Applications*, *74*, 55–69. http://dx.doi.org/10.1016/j.eswa.2017.01.001.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International conference on neural networks, Vol. 4* (pp. 1942–1948). IEEE, http://dx.doi.org/10.1007/978-0-387-30164-8_630.

Kohli, M., & Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*, *5*(4), 458–472. http://dx.doi.org/10.1016/j.jcde.2017.02.005, URL: https://www.sciencedirect.com/science/article/pii/S2288430016301142.

Krohling, R. A., & Coelho, L. d. S. (2006). Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *36*(6), 1407–1416. http://dx.doi.org/10.1109/TSMCB.2006.873185.

Kumar, A., Das, S., & Zelinka, I. (2020). A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems. In *GECCO '20: Genetic and evolutionary computation conference* (p. 80). http://dx.doi.org/10.1145/3377929.3398185.

Lampinen, J., & Storn, R. (2004). Differential evolution. In *Studies in fuzziness and soft computing*: *vol. 141*, *New optimization techniques in engineering* (pp. 123–166). Berlin, Heidelberg: Springer, http://dx.doi.org/10.1007/978-3-540-39930-8_6.

Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2013). *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, Vol. 635* (p. 490). Singapore: Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, http://dx.doi.org/10.1109/cec.2013.6557797.

Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*, *10*(2), 629–640. http://dx.doi.org/10.1016/j.asoc.2009.08.031, URL: https://www.sciencedirect.com/science/article/pii/S1568494609001550 https://www.sciencedirect.com/science/article/abs/pii/S1568494609001550?via%3Dihub.

Lozano, & Jose, A. (2017). Conference report on 2017 IEEE congress on evolutionary computation (IEEE CEC 2017) [Conference reports]. *IEEE Computational Intelligence Magazine*, *12*(4), 5–6. http://dx.doi.org/10.1109/MCI.2017.2742840.

Lyu, D., & Wang, J. (2021). Innovation of Bazaar harper's BAZAAR in the new media era: Fan economic application. In *2021 5th international seminar on education, management and social sciences (ISEMSS 2021)* (pp. 348–352). Atlantis Press, http://dx.doi.org/10.2991/assehr.k.210806.065.

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*, http://dx.doi.org/10.1016/j.advengsoft.2017.07.002.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67. http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

Misevicius, A., & Verene, D. (2021). A hybrid genetic-hierarchical algorithm for the quadratic assignment problem. *Entropy*, *23*(1), 108. http://dx.doi.org/10.3390/e23010108.

Mittal, H., Tripathi, A., Pandey, A. C., & Pal, R. (2020). Gravitational search algorithm: a comprehensive analysis of recent variants. *Multimedia Tools and Applications*, *80*(5), 7581–7608. http://dx.doi.org/10.1007/s11042-020-09831-4, URL: https://link.springer.com/content/pdf/10.1007/s11042-020-09831-4.pdf.

Nolle, L. (2006). On a hill-climbing algorithm with adaptive step size: Towards a control parameter-less black-box optimisation algorithm. *Computational Intelligence Theory and Applications*, http://dx.doi.org/10.1007/3-540-34783-6_56.

Pál, K. F. (2006). Hysteretic optimization for the Sherrington–Kirkpatrick spin glass. *Physica A: Statistical Mechanics and its Applications*, *367*, 261–268. http://dx.doi.org/10.1016/j.physa.2005.11.013.

Pradhan, D., Wang, S., Ali, S., Yue, T., & Liaaen, M. (2021). CBGA-ES+: A cluster-based genetic algorithm with non-dominated elitist selection for supporting multi-objective test optimization. *Ieee Transactions on Software Engineering*, *47*(1), 86–107. http://dx.doi.org/10.1109/TSE.2018.2882176.

Qi, S., Jiang, Y., Yang, J., & Ying, Y. (2022). The integrated development of fan economy and social welfare. *International Journal of Social Science and Education Research*, *5*(1), 528–531. http://dx.doi.org/10.6918/IJOSSER.202201_5(1).0082.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248. http://dx.doi.org/10.1016/j.ins.2009.03.004.

Salih, S. Q., & Alsewari, A. A. (2020). A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer. *Neural Computing and Applications*, *32*(14), 10359–10386. http://dx.doi.org/10.1007/s00521-019-04575-1, URL: https://link.springer.com/content/pdf/10.1007/s00521-019-04575-1.pdf.

dos Santos Coelho, L. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications*, *37*(2), 1676–1683. http://dx.doi.org/10.1016/j.eswa.2009.06.044.

Shehadeh, H. A., Ahmedy, I., & Idris, M. Y. I. (2018). Sperm swarm optimization algorithm for optimizing wireless sensor network challenges. (pp. 53–59). http://dx.doi.org/10.1145/3193092.3193100.

Siddique, N., & Adeli, H. (2016). Physics-based search and optimization: Inspirations from nature. *Expert Systems*, *33*(6), 607–623. http://dx.doi.org/10.1111/exsy.12185.

Siddique, N., & Adeli, H. (2017). Nature-inspired chemical reaction optimisation algorithms. *Cognitive Computation*, *9*(3), 1–12. http://dx.doi.org/10.1007/s12559-017-9485-1.

Uray, E., Carbas, S., Geem, Z. W., & Kim, S. (2022). Parameters optimization of taguchi method integrated hybrid harmony search algorithm for engineering design problems. *Mathematics*, *10*(3), http://dx.doi.org/10.3390/math10030327.

Wang, Y., Cai, Z., Zhou, Y., & Fan, Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization*, *37*(4), 395–413. http://dx.doi.org/10.1007/s00158-008-0238-3.

Wang, L., & Li, L.-p. (2010). An effective differential evolution with level comparison for constrained engineering design. *Structural and Multidisciplinary Optimization*, *41*(6), 947–963. http://dx.doi.org/10.1007/s00158-009-0454-5.

Wang, G., Xu, C., & Liu, G. (2018). The transient electromagnetic inversion based on the simplex-simulated annealing algorithm. In X. Chen, & Q. C. Zhao (Eds.), *2018 37th Chinese control conference (Ccc)* (pp. 4321–4324). New York: Ieee, http://dx.doi.org/10.23919/ChiCC.2018.8484067.

Xin, Y., & Yong, L. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, *3*(2), P.82–102. http://dx.doi.org/10.1109/4235.771163.

Yang, A., & Shim, K. (2020). Antecedents of microblogging users' purchase intention toward celebrities' merchandise: Perspectives of virtual community and fan economy. *Journal of Psychological Research*, *2*(2), 11–26. http://dx.doi.org/10.30564/jpr.v2i2.1646.

Yue, W., Xi, Y., & Guan, X. (2019). A new searching approach using improved multi-ant colony scheme for multi-UAVs in unknown environments. *Ieee Access*, *7*, 161094–161102. http://dx.doi.org/10.1109/ACCESS.2019.2949249.

Zhang, M., Luo, W., & Wang, X. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, *178*(15), 3043–3074. http://dx.doi.org/10.1016/j.ins.2008.02.014, URL: https://www.sciencedirect.com/science/article/pii/S0020025508000510 https://www.sciencedirect.com/science/article/abs/pii/S0020025508000510?via%3Dihub.

Zhou, Z. (2021). Community order and spontaneous economic behavior: Fan community, fan economy and psychology of Cai Xukun. In *2nd international conference on language, art and cultural exchange (ICLACE 2021)* (pp. 416–419). Atlantis Press, http://dx.doi.org/10.2991/assehr.k.210609.082.

Zhu, J., Liu, S., & Ghosh, S. (2019). Model and algorithm of routes planning for emergency relief distribution in disaster management with disaster information update. *Journal of Combinatorial Optimization*, *38*(1), 208–223. http://dx.doi.org/10.1007/s10878-018-00377-8.

Xiaofei Wang was born in Shuangyashan, Heilongjiang Province, China in 1996. She is a doctoral student in the School of Mechanical and Power Engineering, Harbin University of Science and Technology, Harbin, China.

Her research interests include digital imaging processing, deep learning algorithm, control theory, heterogeneous multi-intelligence, robotic technology and intelligent algorithms.

She presented "A New Fracture Image Segmentation Method Based on MSA-k Clustering Algorithm" at the 39th Chinese control conference (CCC2020) accept. She published "Multi-feature Detection of Micro-aneurysms Based on Improved SSA" at Symmetry and "Particle Swarm Optimization and Salp Swarm Algorithm for the Segmentation of Diabetic Retinal Blood Vessel Images "at Computational Intelligence and Neuroscience.

Jiazhong Xu was born in Harbin, Heilongjiang Province, China in 1977. He received his Ph.D. degree from Harbin University of Science and Technology.

His main research interests include image processing, nonlinear control, neural network and composite material molding equipment.

He is doctoral supervisor of the School of Mechanical Power Engineering Supervisor at Harbin University of Science and Technology. Presided over a number of national key research and development projects. He has published more than 70 papers, including 20 papers indexed by SCI and 22 papers indexed by EI.

Cheng Huang was born in China in 1986. He received his master's degree from Harbin Engineering University. He received his Ph.D. from Harbin Institute of Technology.

His research interests include spacecraft attitude orbit control, autonomous spacecraft decision making in rendezvous and approach, autonomous space grabbing in orbit, and virtual simulation of hardware-in-the-loop.

He is a master's supervisor in the School of Automation of Harbin University of Science and Technology. He presided over and participated in several key national R&D projects. In recent five years, he has published eight papers, including 3 SCI and 5 EI, 1 authorized invention patent, 1 utility model patent, and 2 software copyrights.