Applied Soft Computing Journal 93 (2020) 106339

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

FBI inspired meta-optimization

Jui-Sheng Chou *, Ngoc-Mai Nguyen

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 11 January 2019
Received in revised form 20 April 2020
Accepted 23 April 2020
Available online 28 April 2020

Keywords:
Forensic-based investigation algorithm
Metaheuristic optimization
Swarm intelligence and evolutionary computation
Benchmark functions
Construction engineering and project management

ABSTRACT

This study developed a novel optimization algorithm, called Forensic-Based Investigation (FBI), inspired by the suspect investigation–location–pursuit process that is used by police officers. Although numerous unwieldy optimization algorithms hamper their usability by requiring predefined operating parameters, FBI is a user-friendly algorithm that does not require predefined operating parameters. The performance of parameter-free FBI was validated using four experiments: (1) The robustness and efficiency of FBI were compared with those of 12 representations of the top leading metaphors by using 50 renowned multidimensional benchmark problems. The result indicated that FBI remarkably outperformed all other algorithms. (2) FBI was applied to solve a resource-constrained scheduling problem associated with a highway construction project. The experiment demonstrated that FBI yielded the shortest schedule with a success rate of 100%, indicating its stability and robustness. (3) FBI was utilized to solve 30 benchmark functions that were most recently presented at the IEEE Congress on Evolutionary Computation (CEC) competition on bound-constrained problems. Its performance was compared with those of the three winners in CEC to validate its effectiveness. (4) FBI solved high-dimensional problems, by increasing the number of dimensions of benchmark functions to 1000. FBI is efficient because it requires a relatively short computational time for solving problems, it reaches the optimal solution more rapidly than other algorithms, and it efficaciously solves high-dimensional problems. Given that the experiments demonstrated FBI's robustness, efficiency, stability, and user-friendliness, FBI is promising for solving various complex problems. Finally, this study provided the scientific community with a metaheuristic optimization platform for graphically and logically manipulating optimization algorithms.

1. Introduction

Metaheuristics have become more popular than exact methods for solving optimization problems because of their simplicity and the robustness of the results that they produce [1,2]. They are used in a wide range of fields, including engineering, business, transportation, and even social sciences [3,4]. Metaheuristic optimization algorithms are becoming increasingly popular in engineering applications because they (i) depend on rather simple concepts and are easy to implement; (ii) do not require gradient information; (iii) can avoid local optima; and (iv) can be used in a wide range of problems associated with different disciplines [5].

Hussain et al. (2018) [3] surveyed the literature on metaheuristic research published from 1983 to 2016, which consists of 1,222 publications. They provided the top ten leading metaphors that are preferred by researchers for use in designing new metaheuristic algorithms (Fig. 1). Most metaheuristic methods mimic living and survival systems in the real world.

As illustrated in Fig. 1, insects account for most favored metaphors for mimicking social behavior in the design of efficient optimization methods, and among insects, bees are the most popular (artificial bee colony (ABC) algorithm [6]), followed by ants (ant colony optimization [7] and ant lion optimizer [8]), fireflies (firefly algorithm (FA) [9]), spiders (social spider algorithm [10]), and bacteria (bacteria foraging optimization algorithm [11]). The second most popular metaphor is natural evolution and Darwin's theory of survival (genetic algorithm (GA) [12], and differential evolution (DE) [13]). Some animals, such as bats (bat algorithm [14]), fish (whale optimization algorithm (WOA) [5], and artificial fish swarm algorithm [15]), and monkeys (monkey algorithm [16]) have also attracted metaheuristic designers.

Birds (particle swarm optimization (PSO) [17] and cuckoo search [18]), humans (teaching–learning-based optimization (TLBO) algorithm [19]), plants (flower pollination algorithm (FPA) [20]), water (water cycle algorithm (WCA) [21] and water drop algorithm [22]), ecosystems (symbiotic organisms search (SOS) [23]), the electromagnetic force (electromagnetic field optimization (EFO) [24]), and gravitation (gravitational search algorithm (GSA) [25]) have been increasingly used metaphorically in the design of metaheuristic methods.

* Corresponding author.
E-mail address: jschou@mail.ntust.edu.tw (J.-S. Chou).

https://doi.org/10.1016/j.asoc.2020.106339

Nomenclature

FBI	Forensic-Based Investigation
CEC	Congress on Evolutionary Computation
ABC	Artificial Bee Colony
FA	Firefly Algorithm
GA	Genetic Algorithm
DE	Differential Evolution
WOA	Whale Optimization Algorithm
PSO	Particle Swarm Optimization
TLBO	Teaching–Learning-Based Optimization
FPA	Flower Pollination Algorithm
WCA	Water Cycle Algorithm
SOS	Symbiotic Organisms Search
EFO	Electromagnetic Field Optimization
GSA	Gravitational Search Algorithm
RCPSP	Resource-Constrained Project Scheduling Problem
NP_A	Number of suspected locations that should be investigated
NP_B	Number of police agents in pursuit team
NP	Population size
X_{A_i}	Suspected location i to be investigated (Location X_{A_i})
X_{A1_i} and X_{A2_i}	New potential locations of X_{A_i} in steps A1 and A2
X_{B_i}	Location of police agent i who is pursuing the suspect (Location X_{B_i})
X_{B1_i} and X_{B2_i}	New potential locations of X_{B_i} in steps B1 and B2
p_{A_i}	Possibility that the suspect is at location X_{A_i} (objective value of solution X_{A_i})
p_{A1_i} and p_{A2_i}	Possibilities that the suspect is at locations X_{A1_i} and X_{A2_i} (objective values of locations X_{A1_i} and X_{A2_i}, respectively)
p_{B_i}	Possibility that the suspect is at location X_{B_i} (objective value of location X_{B_i})
p_{B1_i} and p_{B2_i}	Possibilities that the suspect is at locations X_{B1_i} and X_{B2_i} (objective values of locations X_{B1_i} and X_{B2_i}, respectively)
D	Number of dimensions
g	Current iteration
g_{max}	Maximum number of iterations
p_{worst}	Worst possibility (Worst objective value)
p_{best}	Best possibility (Best objective value)
X_{best}	Best location (Best solution)
$Prob(X_{A_i})$	Probability of location X_{A_i}
X_{B_r}	Location of a random police agent r
p_{B_r}	Possibility that the suspect is at location X_{B_r} (objective value of location X_{B_r})

Each metaheuristic algorithm has unique advantages with respect to robustness and performance in noisy environments, under uncertainty, and in various problem spaces [26]. No single metaheuristic algorithm can outperform all others in all possible optimization problems [5,23]. In the early twenty-first century, the field of metaheuristic optimization has seen the development of numerous "novel" metaheuristic algorithms. This fact raises a concern about the quality of any proposed (and supposedly novel) algorithm [27]. Notwithstanding the aforementioned concern, numerous excellent algorithms may exhibit superior performance for solving particular problems. Therefore, new high-performance metaheuristic algorithms are continuously being developed to solve particular optimization problems, especially currently, with the rapid increase in diverse engineering problems [28–32].

The main motivation for developing a new algorithm is its capacity to effectively and efficiently solve various optimization problems. In this paper, a novel optimization method, the forensic-based investigation (FBI) algorithm, is proposed to determine global solutions for continuous nonlinear functions with low computational effort and high accuracy. FBI is inspired by the suspect investigation–location–pursuit process of police officers engaged in criminal investigations. In a search space, investigators analyze and evaluate collected items of evidence to identify and locate a suspect, based on which police agents may make arrests. All investigators and police agents must closely coordinate with each other and comply with headquarters' commands throughout the investigation–location–pursuit process. The output of the process is finding and capturing the suspect.

Attempting to design a perfect test set for metaheuristic optimization that contains all of functions to determine whether an algorithm is better than others, is fruitless. To compare metaheuristic optimization algorithms comprehensively, 50 well-known benchmark problems are considered. This set is sufficiently large to include many types of problems, such as unimodal (17), multimodal (33), separable (14), nonseparable (36), and multidimensional (50) [33]. A real-world case study of a resource-constrained project scheduling problem (RCPSP) was analyzed to demonstrate the effectiveness of FBI. Researchers recognize RCPSP as one of the most crucial and challenging types of operational research problems [34,35].

FBI was validated and compared with 12 representations of the aforementioned top ten metaphors, as follows: ABC and FA (representations of insect metaphor), GA and DE (natural evolution), WOA (animals), PSO (birds), TLBO (humans), FPA (plants), WCA (water), SOS (ecosystem), EFO (electromagnetic force), and GSA (gravitation). Experimental results indicated that FBI outperformed all other algorithms for solving all 50 benchmark problems and the RCPSP in the case studies. Therefore, it is highly competitive with state-of-the-art optimization methods.

A metaheuristic optimization platform was developed, and it provides the following benefits: (i) It greatly facilitates and accelerates the aforementioned comparison, and its users are not required to understand any programming software; (ii) it provides an informative overview of the parameter settings, choice of objective function, and whole optimization process; (iii) it displays performance indicators clearly, logically, and graphically, making the comparison more intuitive, the results easier to read, and conclusions easier to draw.

FBI was further analyzed by using it to solve 30 benchmark functions with a diverse set of characteristics that were presented at the IEEE Congress on Evolutionary Computation (CEC) competition on bound-constrained problems. Its performance levels were compared with those of the winners of that competition. The efficiency of the proposed algorithm was also revealed by its outstanding solutions for high-dimensional problems (D =1,000).

The remainder of the paper is organized as follows: Section 2 introduces the FBI algorithm in detail. Section 3 proposes a comprehensive performance comparison between FBI and the aforementioned algorithms using four experiments, and introduces a metaheuristic optimization platform. Section 4 provides and analyzes the experiment results. Section 5 discusses the novelty and the originality of FBI. Section 6 draws conclusions.

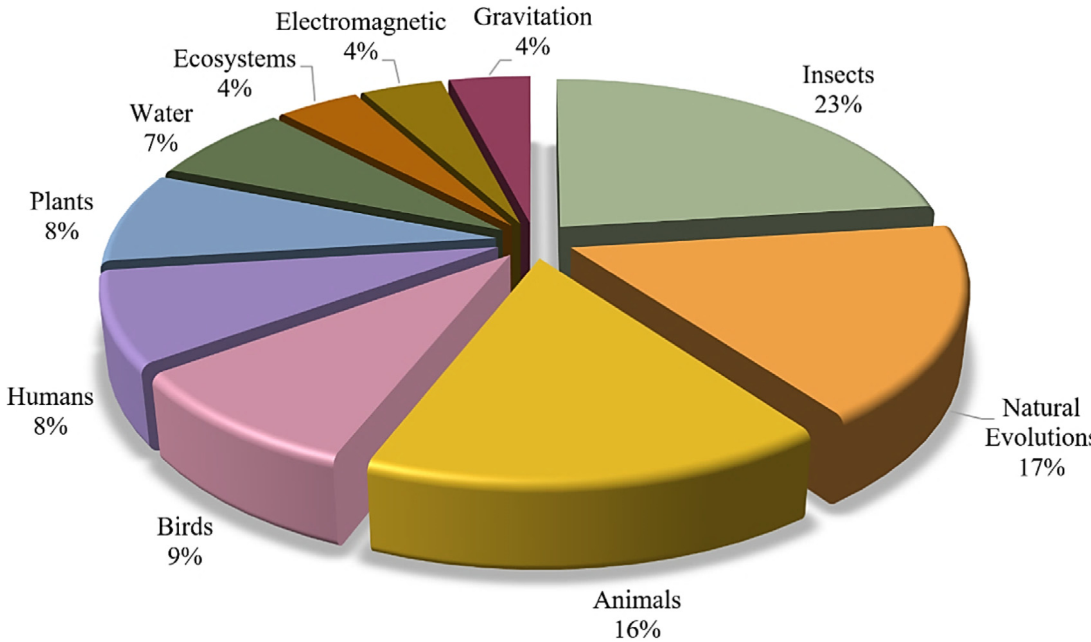

Fig. 1. Metaphors adopted by researchers for designing new metaheuristics.

2. Forensic-based investigation algorithm

This section presents the design of a novel metaheuristic optimization algorithm. The first subsection describes the forensic investigation process. The second subsection elaborates the concept of the FBI algorithm in detail.

2.1. Forensic investigation process

In every country, one of the most frequent but dangerous tasks of law enforcement is forensic investigation. Every investigation is different and may require a different route to be taken. For example, in some cases, the identity of the offender is known from the outset, and as a result, the investigation quickly enters the suspect management phase. In other cases, the identity of the offender may never be known or is discovered only after further investigation. Nevertheless, investigators usually undergo similar stages of investigation activities [36].

Salet (2017) indicated that a large-scale forensic investigation by the police officers comprises five steps; steps 2, 3, and 4 of which can be seen as a cyclical process [37]. Fig. 2 illustrates these five steps.

(1) Open a case: The investigation of the incident begins with information that has been found by the police officers who were the first to arrive on the crime scene. This information forms the main starting point for members of the investigation team, who begin by following several standard procedures to gain a first impression of what might have happened. The team members investigate the crime scene, the victim, possible suspects, and their background information; the team locates witnesses and questions witnesses.

(2) Interpretation of findings: By sharing information in team briefings, team members attempt to gain an overview of all available information. In this second step, the team assesses the information and tries to connect information with the impressions they already have about the case to evaluate possible suspects.

(3) Direction of inquiry: In the third step of the investigation, team members develop several theories (including scenarios, motives for the crime, and lines of inquiry) based on the interpretation of findings. Again, the team assesses the findings, resulting in a new direction or a confirmation, change, or termination of the existing directions of inquiry.

(4) Actions: After lines of inquiry and priorities have been set, the team makes decisions about further actions to be taken. This step is closely related to decisions about priorities, and the most promising research direction is normally pursued first. Once again, the actions taken provide new information. As soon as this information is obtained, the investigation team interprets its meaning or implications based on the information that is already available. The interpretation of new findings may lead to adjustments of directions of inquiry and actions.

(5) Prosecution: This process continues until one dominant and fairly complete story of what "really" happened remains. It comes to an end when a serious suspect has been identified, before a decision is made about their prosecution.

No rule governs the number of police officers who participate in an investigation: the number typically depends on the severity, difficulty, and complexity of the incident. The following three famous cases provide some numbers for reference. (1) On the afternoon of May 17, 1974, elements of the Symbionese Liberation Army (SLA, a well-established paramilitary terrorist organization, according to law enforcement officials) barricaded themselves in a residence on East 54th Street at Compton Avenue in Los Angeles. Sixty Special Weapons and Tactics (SWAT) team members participated in a several-hour gun battle with the SLA [38]. (2) Professor Jose Maria Sison was arrested on August 28, 2007, and charged with ordering the deaths of two security agents in the Philippines in 2003 and 2004. The arrest of Sison was a large police operation that involved approximately 100 police agents [39]. (3) On June 28, 2018, two drug lords, who were wanted for the trafficking of 890 kg of heroin from Laos, were killed in a massive raid in northern Vietnam. More than 200 armed and armored officers were seen at the site [40]. Thus, the number of officers, which may be tens, hundreds, or thousands, depends on the characteristics of the case.

2.2. Development of metaheuristic optimization algorithm

The FBI algorithm is inspired by the forensic investigation process of police officers, specifically the criminal investigation

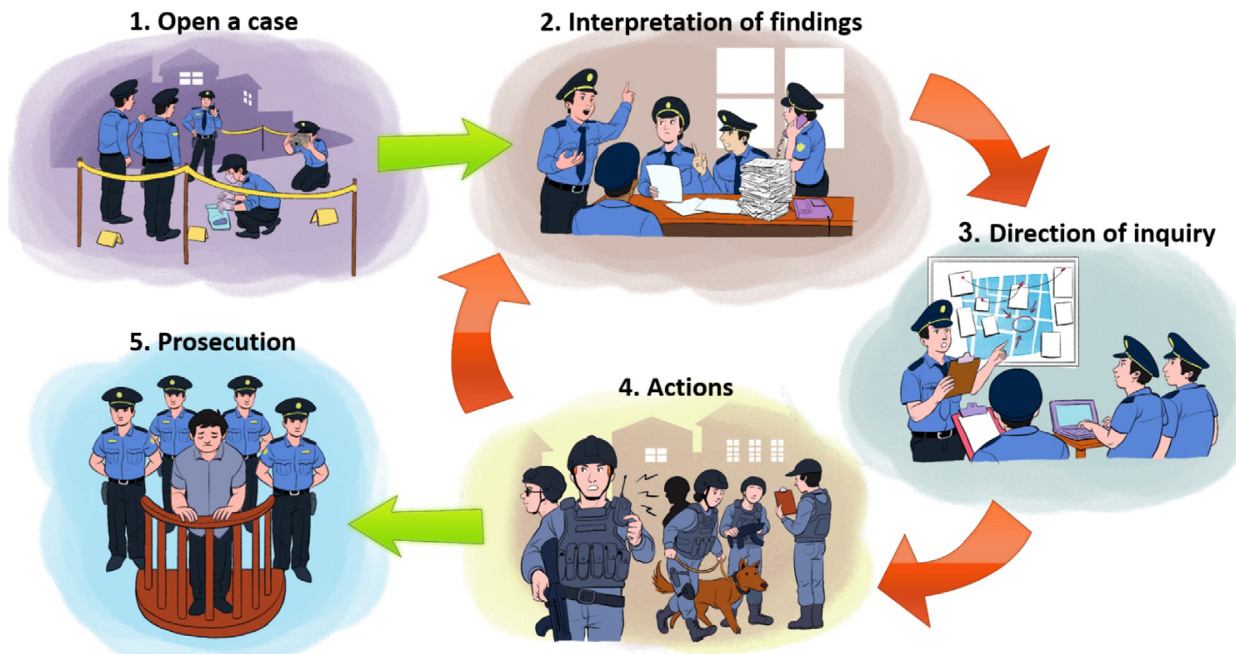

Fig. 2. Process of an investigation.

process. Immediately after criminal activity is reported, an investigation may start. Investigative tasks are aimed at identifying physical evidence, gathering information, collecting and protecting evidence, interviewing witnesses, and interviewing and interrogating suspects [41]. Based on the collected evidence and witnesses' statements, all potential suspects and their locations are identified and their potential locations fixed within a search space. FBI makes two assumptions: that only one most wanted suspect in a case, and that he remains in his hiding place throughout the investigation. The output of the process is locating and capturing the suspect.

A team of investigators is formed to analyze the probable hiding locations of the suspect within the search space, which are called "suspected locations". After the team of investigators has identified the location with the highest probability, a search area is delineated and a pursuit team is organized. All police agents in the pursuit team head toward the designated location, together with team members who can handle and arrest the suspect.

The pursuit team follows commands from headquarters and moves toward the suspected location to approach, reporting back all information relevant to the suspected location. Throughout the investigation–finding–approaching process, the investigation and pursuit teams closely coordinate with each other: The investigation team instructs the pursuit team to approach the locations, while the pursuit team frequently reports the results of their search so that the investigation team can update the information and maximize the accuracy of its next assessments.

Figs. 3 and 4 present the general principles and mathematical interpretation of FBI, respectively. The algorithm has two phases: the investigation phase (phase *A*), which is implemented by the team of investigators, and the pursuit phase (phase *B*), which is implemented by the team of police agents. In the investigation phase, X_{A_i} is the i^{th} suspected location to be investigated, $i = 1, 2, \ldots, NP_A$; where NP_A is the number of suspected locations that should be investigated. In the pursuit phase, X_{B_i} is the location of police agent i, who is pursuing the suspect, $i = 1, 2, \ldots, NP_B$; where NP_B is the number of police agents in pursuit team. In this study, we assume $NP_A = NP_B = NP$, such that NP is the population size. A forensic investigation is a cyclical process and ends when the number of the current iteration (g) reaches the present maximum number of iterations (g_{max}). FBI is a search method that utilizes NP D-dimensional parameter vectors.

***Step A1*:** This step represents the "interpretation of findings". The team assesses the information and initially identifies possible suspect locations. Each possible location of the suspect is investigated in the context of other findings. First, a new suspected location from X_{A_i}, named X_{A1_i} is inferred on the basis of X_{A_i} and information relevant to other suspected locations. In this work, each individual is assumed to move under the influences of other individuals. The general formula for the movement is as follows:

$$X_{A1_{ij}} = X_{A_{ij}} + ((rand - 0.5)^{*}2)^{*}\left(\sum_{a=1}^{a_1} X_{A_{aj}}\right)/a1 \tag{1}$$

where $j = 1, 2, \ldots, D$; D is the number of dimensions; $((rand - 0.5)^{*}2)$ indicates a random number in the range $[-1, 1]$; $rand$ is a random number in the range $[0, 1]$; $a_1 \in \{1, 2, \ldots, n-1\}$, representing the number of individuals which influence the movement of $X_{A_{ij}}$; $a = 1, 2, \ldots, a_1$. Based on numerical trial-and-error experiments with various forms of Eq. (1), $a_1 = 2$ was determined to produce the best result in a short computational time. Thus, the new suspected location X_{A1_i} is presented in Eq. (2); p_{A_i} is defined as the possibility (objective value) that the suspect is at location X_{A_i}, which means p_{A_i} is the objective value of location X_{A_i} (i.e., $p_{A_i} = f_{objective}(X_{A_i})$). The investigators evaluate the possibility p_{A1_i} of the new suspect's location and compare it with that of the current entry. The location that has better possibility (objectve value) that the suspect is present will be retained, and the other one is abandoned.

$$X_{A1_{ij}} = X_{A_{ij}} + ((rand_1 - 0.5)^{*}2)^{*}(X_{A_{ij}} - (X_{A_{kj}} + X_{A_{hj}})/2) \tag{2}$$

where *k, h,* and *i* are three suspected locations: *{k, h, i}* ∈ *{1, 2, …, NP}*, *k* and *h* are chosen randomly; $j = 1, 2, \ldots, D$; NP is the number of suspected locations; D is the number of dimensions; $((rand_1 - 0.5)^{*}2)$ indicates a random number in the range $[-1, 1]$; $rand_1$ is a random number in the range $[0, 1]$.

***Step A2*:** This step is the "direction of inquiry". Investigators compare the probability of each suspected location to those of others to determine the most likely suspected location that

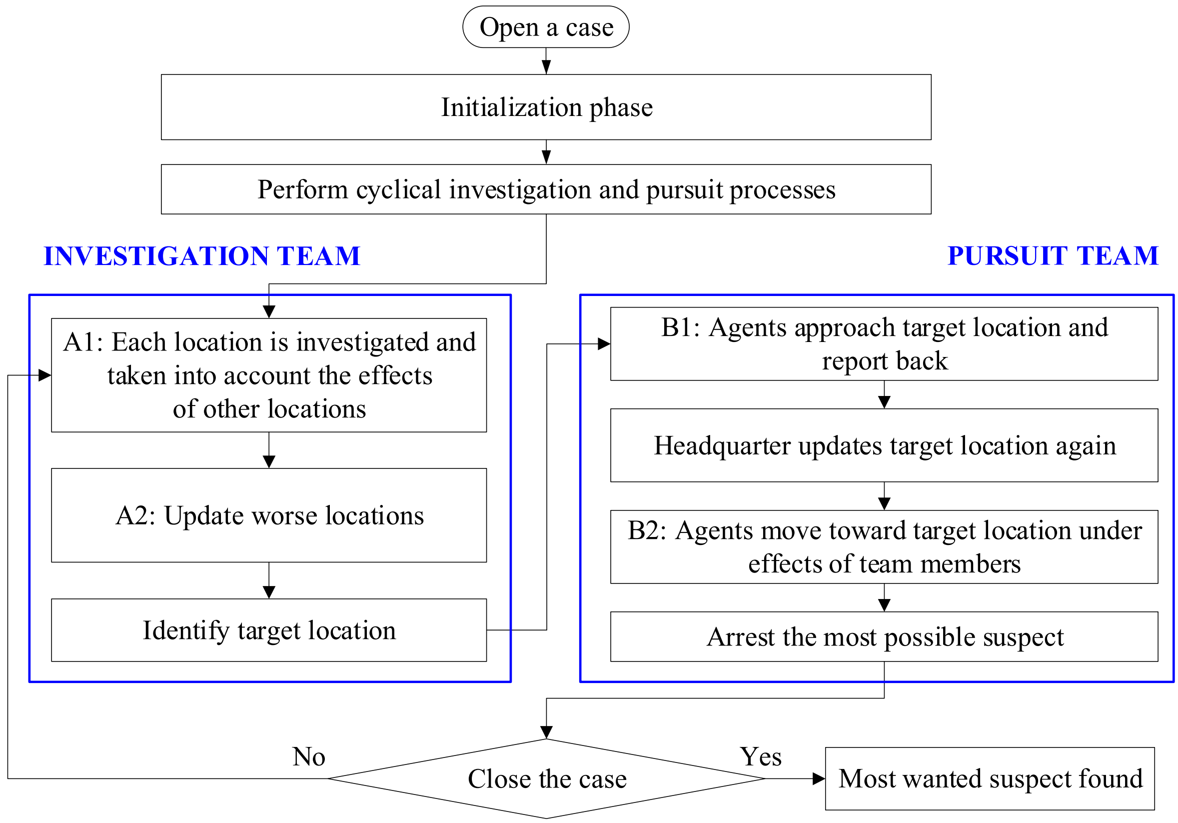

Fig. 3. General principle of FBI algorithm.

should be further investigated. When the optimization is a minimization problem, p_{worst} is the lowest possibility (the worst objective value), p_{best} is the highest possibility (the best objective value) and X_{best} is the best location. Understandably, while p_{worst} is different from p_{best}, any location with a low probability is likely to be given up in favor of another location with a higher probability. The probability of each location, $Prob(X_{A_i})$, is estimated using Eq. (3), and a high value of $Prob(X_{A_i})$ corresponds to a high probability for the location.

$$Prob\ (X_{A_i}) = (p_{worst} - p_{A_i})/(p_{worst} - p_{best}) \tag{3}$$

The update of a searching location is affected by the directions of other suspected locations. However, not all directions are changed; randomly chosen directions in the updated location are changed to increase the diversity of search areas. In this step, the move of X_{A_i} is simply affected by the best individual and the other random individuals. Step A2 is similar to step A1; the general formula for the move is presented in Eq. (4).

$$X_{A2_i} = X_{best} + \sum_{b=1}^{a_2} \alpha_b^* X_{A_{bj}} \tag{4}$$

where X_{best} is the best location; a_2 is the number of individuals that affect the move of X_{A2_i}: $a_2 \in \{1, 2, \ldots, n-1\}$; $b = 1, 2, \ldots, a_2$; α_b is the effectiveness coefficient ($\alpha_b = [-1, 1]$) of the other individuals to the move. Numerical experiments produced $a_2 = 3$. Thus, the new suspected location $X_{A2_{ij}}$ is generated using Eq. (5). The possibility (objective value) is then calculated to determine whether to update the suspected location.

$$X_{A2_{ij}} = X_{best} + X_{A_{dj}} + rand_5^*(X_{A_{ej}} - X_{A_{fj}}) \tag{5}$$

where X_{best} is the best location updated in *Step A1*, $rand_5$ is the random number in the range [0,1]; and *d, e, f*, and *i* are four suspected locations: *{d, e, f, i}* ∈ *{1, 2, …, NP}*, *d, e*, and *f* are chosen randomly; $j = 1, 2, \ldots, D$.

***Step B1*:** This step represents the "actions". After receiving a report of the best location from investigation team, all agents in the pursuit team must approach the target in a coordinated fashion to arrest the suspect. Each agent B_i approaches the location that has the best possibility (objective value) according to Eq. (6). The newly approached location is updated if it yields a better possibility (objective value) than that (p_{B_i}) of the old location.

$$X_{B1_{ij}} = rand_6^* X_{B_{ij}} + rand_7^*(X_{best} - X_{B_{ij}}) \tag{6}$$

where X_{best} is the best location that the investigation team has provided; $rand_6$ and $rand_7$ are two random numbers in the range [0, 1]; $j = 1, 2, \ldots, D$.

***Step B2*:** This step extends the process of "actions". Whenever they make any move, the police agents report the possibilities (objective values) of the new locations back to headquarters. The headquarters immediately updates the location and commands the pursuit team to approach that position. At that moment, each agent B_i closely coordinates with all the other agents; agent B_i moves toward the best location, and agent B_i is subjected to the influence of other team member (agent B_r, with its possibility p_{B_r}). The new location of agent B_i (which is X_{B2_i}) is formalized as Eq. (7) if p_{B_r} is better than p_{B_i}; otherwise, it is formalized as Eq. (8). The newly found location is updated when it attains a better possibility (objective value) than that of the old location.

$$X_{B2_{ij}} = X_{B_{rj}} + rand_8^*(X_{B_{rj}} - X_{B_{ij}}) + rand_9^*(X_{best} - X_{B_{rj}}) \tag{7}$$

$$X_{B2_{ij}} = X_{B_{ij}} + rand_{10}^*(X_{B_{ij}} - X_{B_{rj}}) + rand_{11}^*(X_{best} - X_{B_{ij}}) \tag{8}$$

where X_{best} is the best location provided in *Step B1*, $rand_8$, $rand_9$, $rand_{10}$, and $rand_{11}$ are random numbers in the range [0, 1]; *r* and *i* are two police agents: *{r, i}* ∈ *{1, 2, …, NP}*, and *r* is chosen randomly; $j = 1, 2, \ldots, D$.

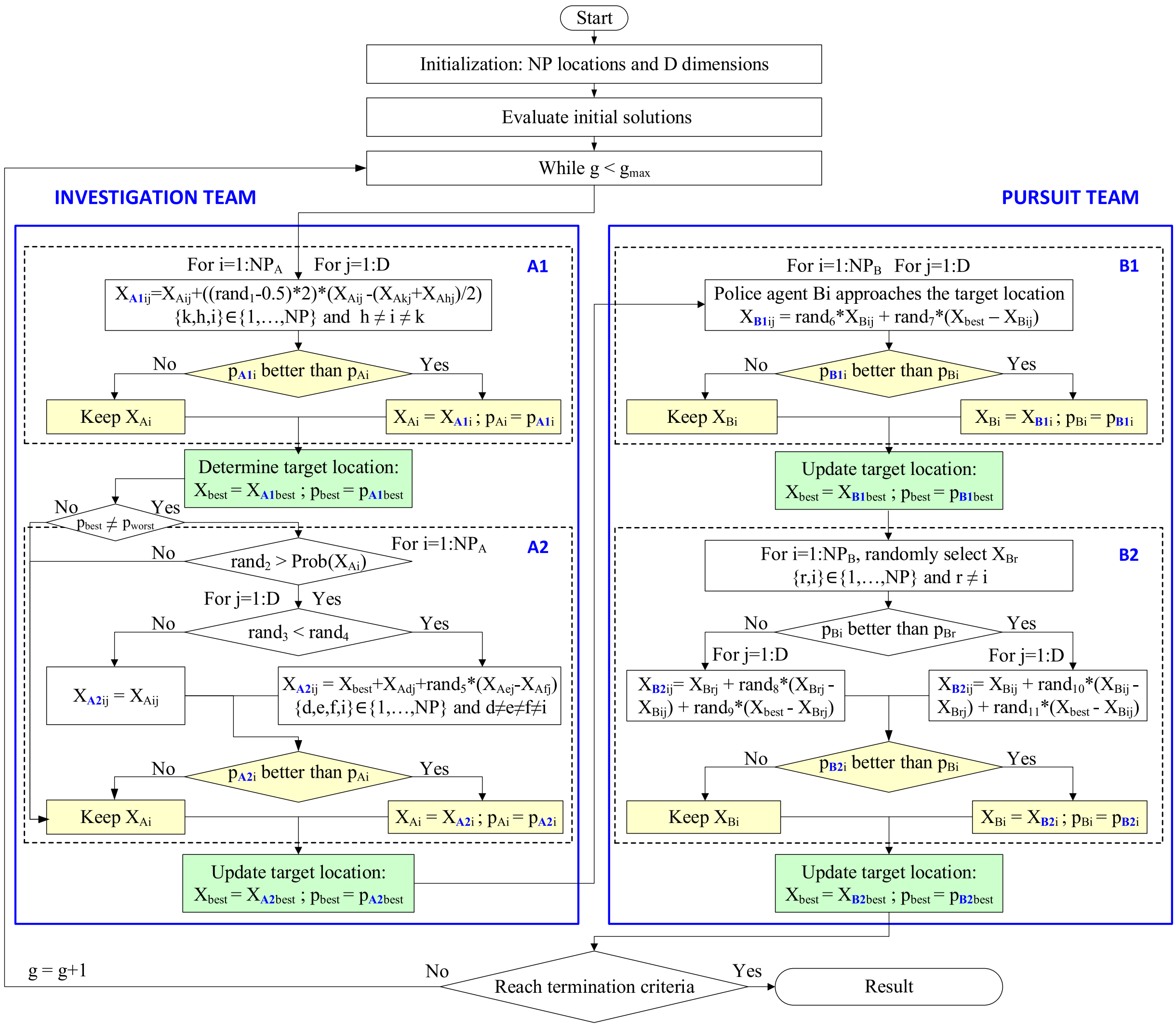

Fig. 4. Mathematical interpretation of FBI algorithm.

The pursuit team reports the best location of suspect to the investigation team to help them increase the accuracy of their analysis and assessments. As Salet (2017) indicated, the process of forensic investigations can be seen as cyclical [37]. The pseudocode for the FBI algorithm (Fig. 5) is provided to illustrate its concepts.

3. Comprehensive validation plan

The FBI algorithm was validated and compared with 12 representations of the aforementioned top ten metaphors that are preferred by most researchers for designing new metaheuristic algorithms, namely ABC and FA (representations for insects), GA and DE (natural evolutions), WOA (animals), PSO (birds), TLBO (humans), FPA (plants), WCA (water), SOS (ecosystem), EFO (electromagnetic force), and GSA (gravitation).

Fifty well-known benchmark functions that contain various complex problems were utilized for validation and comparison. FBI and the aforementioned 12 algorithms were used to solve a RCPSP associated with a real construction project to demonstrate the efficiency of FBI.

FBI was further analyzed by using it to solve 30 benchmark functions that were presented at the IEEE CEC competition on bound-constrained problems. Its performance levels were compared with those of the winners of that competition.

3.1. Benchmark functions with various characteristics

In the field of evolutionary computation, algorithms are commonly compared using a large test set, especially when the test involves function optimization. Karaboga and Akay (2009) provided a set of 50 commonly used benchmark problems to test the performance levels of the GA, DE, PSO, and the ABC algorithms [33]. This set is sufficiently large to include diverse types of problems, including unimodal, multimodal, separable, nonseparable, and multidimensional.

This study took the advantage of the work of Karaboga and Akay [33] by using those 50 benchmark functions for a comprehensive comparison and evaluation of the 13 algorithms' performance levels. Table 1 presents the initial ranges, formulations, characteristics, and numbers of dimensions of these problems.

Unimodal test functions have a single optimum, so they can be used to benchmark the exploitation and convergence of an algorithm. Conversely, multimodal functions are used to test the ability of algorithms to eliminate local minima [42]. Algorithms that seek optimal values encounter difficulties with both unimodal functions with flat surfaces and multimodal functions.

Another group of test problems comprises separable and nonseparable functions. A p-variable separable function can be expressed as the sum of p single-variable functions. Nonseparable functions cannot be expressed in this form. The variables in these

```
/*Initialization phase*/
Initialize the population X_i (i = 1, 2, ..., NP)
X_{A_i} = X_{B_i} = X_i
/*Start the cyclical investigation and pursuit processes*/
while (g < maximum number of iterations)
/*Investigation team – Team A*/
    For i = 1:NP /*Step A1*/
        For j=1:D, where D = dimension of the problem
            Generate new location X_{A1_ij} by using Eq. (2)
        End for
        Calculate p_{A1_i} (objective value of location X_{A1_i})
        Update X_{A_i} and p_{A_i}
    End for
    Update best location X_best = X_{A1_best} and global best p_best = p_{A1_best}
    If p_best ≠ p_worst /*Step A2*/
        For i = 1:NP
        Calculate probability Prob(X_{A_i}) by using Eq. (3)
            If rand_2 > Prob(X_{A_i})
                For j = 1:D
                    If rand_3 < rand_4
                        Generate new location X_{A2_ij} by using Eq. (5)
                    End if
                End for
                Calculate p_{A2_i}
                Update X_{A_i} and p_{A_i}
            End if
        End for
    Update best location X_best = X_{A2_best} and global best p_best = p_{A2_best}
    End if
/*Pursuit team – Team B*/
    For i = 1:NP /*Step B1*/
        For j=1:D
            Generate new location X_{B1_ij} by using Eq. (6)
        End for
        Calculate p_{B1_i}
        Update X_{B_i} and p_{B_i}
    End for
    Update best location X_best = X_{B1_best} and global best p_best = p_{B1_best}
    For i = 1:NP /*Step B2*/
        Randomly select X_{B_r}
        If p_{B_r} better than p_{B_i}
            For j=1:D
                Generate new location X_{B2_ij} by using Eq. (7)
            End for
        Else
            For j=1:D
                Generate new location X_{B2_ij} by using Eq. (8)
            End for
        End if
        Calculate p_{B2_i}
        Update X_{B_i} and p_{B_i}
    End for
    Update best location X_best = X_{B2_best} and global best p_best = p_{B2_best}
End while
/*End the cyclical investigation process*/
```

Fig. 5. Pseudocode for FBI algorithm.

Table 1
Benchmark functions used in experiments.

No.	Function	C	D	Range	Formulation
F1	Stepint	US	5	[−5.12, 5.12]	$f(x) = 25 + \sum_{i=1}^{5} \lfloor x_i \rfloor$
F2	Step	US	30	[−100, 100]	$f(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$
F3	Sphere	US	30	[−100, 100]	$f(x) = \sum_{i=1}^{n} x_i^2$
F4	SumSquares	US	30	[−10, 10]	$f(x) = \sum_{i=1}^{n} i x_i^2$
F5	Quartic	US	30	[−1.28, 1.28]	$f(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}[0, 1)$
F6	Beale	UN	2	[−4.5, 4.5]	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$
F7	Easom	UN	2	[−100, 100]	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
F8	Matyas	UN	2	[−10, 10]	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$
F9	Colville	UN	4	[−10, 10]	$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1)$
F10	Trid6	UN	6	[-D^2, D^2]	$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$
F11	Trid10	UN	10	[-D^2, D^2]	$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$
F12	Zakharov	UN	10	[−5, 10]	$f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i^2\right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^4$
F13	Powell	UN	24	[−4, 5]	$f(x) = \sum_{i=1}^{n/k}(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_i)^2 + (x_{4i-2} - x_{i-1})^4 + 10(x_{4i-3} - x_{4i})^4$
F14	Schwefel 2.22	UN	30	[−10, 10]	$f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$
F15	Schwefel 1.2	UN	30	[−100, 100]	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$
F16	Rosenbrock	UN	30	[−30, 30]	$f(x) = \sum_{i=1}^{n-1}[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2]$
F17	Dixon–Price	UN	30	[−10, 10]	$f(x) = (x_i - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$
F18	Foxholes	MS	2	[−65.536, 65.536]	$f(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j} + \sum_{i=1}^{n}(x_j - a_{ij})^6\right]^{-1}$
F19	Branin	MS	2	[−5, 10]; [0, 15]	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$
F20	Bohachevsky1	MS	2	[−100, 100]	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$
F21	Booth	MS	2	[−10, 10]	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
F22	Rastrigin	MS	30	[−5.12, 5.12]	$f(x) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$
F23	Schwefel	MS	30	[−500, 500]	$f(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$
F24	Michalewicz2	MS	2	[0, π]	$f(x) = -\sum_{i=1}^{n} - \sin(x_i)\left(\sin\left(\frac{i x_i^2}{\pi}\right)\right)^{2m}; \ m = 2$
F25	Michalewicz5	MS	5	[0, π]	$f(x) = -\sum_{i=1}^{n} \sin(x_i)\left(\sin\left(\frac{i x_i^2}{\pi}\right)\right)^{2m}; \ m = 5$
F26	Michalewicz10	MS	10	[0, π]	$f(x) = -\sum_{i=1}^{n} \sin(x_i)\left(\sin\left(\frac{i x_i^2}{\pi}\right)\right)^{2m}; \ m = 10$
F27	Schaffer	MN	2	[−100, 100]	$f(x) = 0.5 + \frac{\sin 2\left(\sqrt{(x_1^2 + x_2^2)}\right) - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}$
F28	6 Hump Camel Back	MN	2	[−5, 5]	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$
F29	Bohachevsky2	MN	2	[−100, 100]	$f(x) = x_1^2 - 2x_2^2 - 0.3\cos(3\pi x_1)(4\pi x_2) + 0.3$
F30	Bohachevsky3	MN	2	[−100, 100]	$f(x) = x_1^2 - 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$
F31	Shubert	MN	2	[−10, 10]	$f(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$
F32	GoldStein–Price	MN	2	[−2, 2]	$f(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right]\left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$
F33	Kowalik	MN	4	[−5, 5]	$f(x) = \sum_{i=1}^{11}\left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^2$
F34	Shekel5	MN	4	[0, 10]	$f(x) = \sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$
F35	Shekel7	MN	4	[0, 10]	$f(x) = \sum_{i=1}^{7}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$
F36	Shekel10	MN	4	[0, 10]	$f(x) = \sum_{i=1}^{10}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$
F37	Perm	MN	4	[−D, D]	$f(x) = \sum_{k=1}^{n}\left[\sum_{i=1}^{n}\left(i^k + \beta\right)\left(\left(\frac{x_i}{i}\right)^k - 1\right)\right]^2$

(*continued on next page*)

Table 1 (*continued*).

No.	Function	C	D	Range	Formulation
F38	PowerSum	MN	4	[0, D]	$f(x) = \sum_{k=1}^{n}\left[\left(\sum_{i=1}^{n} x_i^k\right) - b_k\right]^2$
F39	Hartman3	MN	3	[0, 1]	$f(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]$
F40	Hartman6	MN	6	[0, 1]	$f(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right]$
F41	Griewank	MN	30	[−600, 600]	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_2^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F42	Ackley	MN	30	[−32, 32]	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$
F43	Penalized	MN	30	[−50, 50]	$f(x) = \frac{\pi}{n}\left\{10\sin 2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin 2(\pi y_1)\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i\ a \\ 0, & -a\ \ x_i\ \ a \\ k(-x_i - a)^m, & x_i -\ a \end{cases}$
F44	Penalized 2	MN	30	[−50, 50]	$f(x) = 0.1\left\{\sin 2(\pi x_1) + \sum_{i=1}^{n-1}(\ x_i - 1)^2\left[1 + \sin 2(3\pi x_{i+1})\right] + (\ x_n - 1)^2\left[1 + \sin 2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
F45	Langerman 2	MN	2	[0, 10]	$f(x) = -\sum_{i=1}^{m} c_i\left(\exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}(x_j - a_{ij})^2\right)\cos\left(\pi\sum_{j=1}^{n}(x_j - a_{ij})^2\right)\right)$
F46	Langerman 5	MN	5	[0, 10]	$f(x) = -\sum_{i=1}^{m} c_i\left(\exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}(x_j - a_{ij})^2\right)\cos\left(\pi\sum_{j=1}^{n}(x_j - a_{ij})^2\right)\right)$
F47	Langerman 10	MN	10	[0, 10]	$f(x) = -\sum_{i=1}^{m} c_i\left(\exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}(x_j - a_{ij})^2\right)\cos\left(\pi\sum_{j=1}^{n}(x_j - a_{ij})^2\right)\right)$
F48	Fletcher–Powell 2	MN	2	[-π, π]	$f(x) = \sum_{i=1}^{n}(A_i - B_i)^2;\ A_i = \sum_{j=1}^{n}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j);$ $B_i = \sum_{j=1}^{n}(a_{ij}\sin x_j + b_{ij}\cos x_j)$
F49	Fletcher–Powell 5	MN	5	[-π, π]	$f(x) = \sum_{i=1}^{n}(A_i - B_i)^2;\ A_i = \sum_{j=1}^{n}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j);$ $B_i = \sum_{j=1}^{n}(a_{ij}\sin x_j + b_{ij}\cos x_j)$
F50	Fletcher–Powell 10	MN	10	[-π, π]	$f(x) = \sum_{i=1}^{n}(A_i - B_i)^2;\ A_i = \sum_{j=1}^{n}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j);$ $B_i = \sum_{j=1}^{n}(a_{ij}\sin x_j + b_{ij}\cos x_j)$

Note: C: Characteristic, D: Dimension, U: Unimodal, M: Multimodal, S: Separable, N: Nonseparable.

functions are interrelated. In the set herein, as seen in Table 1, 33 functions are multimodal, and 17 are unimodal; furthermore, 14 are separable, and 36 are nonseparable. The difficulty of solution increases with the dimensionality of the function.

3.2. Resource-constrained project scheduling problem

The RCPSP is a classic nondeterministic polynomial-time (NP)-hard optimization problem [43]. The RCPSP has great practical and theoretical importance. Practically, improving project scheduling as a critical part of project management can ensure project completion and significantly reduce related costs. Theoretically, researchers recognize the RCPSP as one of the most vital and challenging types of operational research problems [34,35,44], or to quote one study, "one of the most intractable problems in operations research" [45] .

A project comprises a set of activities $G = \{0, 1, 2, \ldots, N, N+1\}$; dummy activities *0* and $N+1$ represent the start and the end of the project, respectively. A set of renewable resources K exists such that each $k \in K$ has its capacity limit R_k. The duration of an activity i in G is denoted as d_i, and r_{ik} is the amount of resource k that is required by activity i. The makespan is the time spent on the project. The activities are interrelated by two types of constraints: (1) precedence constraints prevent activity i from being begun before all its predecessor activities P_i have been completed, and (2) the sum of the required amount of resource k in any period cannot exceed R_k. The aim of the resource-constrained (RC) problem is to calculate precedence-feasible and resource-feasible completion times for all activities to minimize the project duration.

Let the finish time of activity i be denoted as f_i, then the mathematical formula for RC is as follows:

$$\min\{\max f_i | i = 1, 2, \ldots, N\} \tag{9}$$

subject to:

$$f_j - f_i \geq d_i\,; \quad \forall j \in P_i\,; \quad i = 1, 2, \ldots, N \tag{10}$$

$$\sum_{A_t} r_{ik} \leq R_k\,; \quad k = 1, 2, \ldots, K\,; \quad t = s_1, s_2, \ldots, s_N \tag{11}$$

Eq. (9) is the objective. Eqs. (10) and (11) specify precedence constraints and resource constraints, respectively. A_t is the set of ongoing activities at t, and $s_i = (f_i - d_i)$ is the start time of activity i.

In this study, a real highway bridge construction project (Sears et al. 2008) [46] is analyzed to demonstrate the effectiveness of the proposed FBI for solving that RC problem, and the results thus obtained are compared with those obtained using the aforementioned 12 methods (PSO, GA, DE, ABC, FA, WOA, FPA, TLBO, SOS, WCA, EFO, and GSA). The construction in the case study consists of 42 activities with various daily resource demands.

Fig. 6 displays the precedence relationships of the network activities in the construction project. The three types of renewable resources have maximum daily availability values of 12, 8, and 8 units, respectively. In the figure, 44 cycle nodes represent 42 project activities and two dummy activities (start and finish).

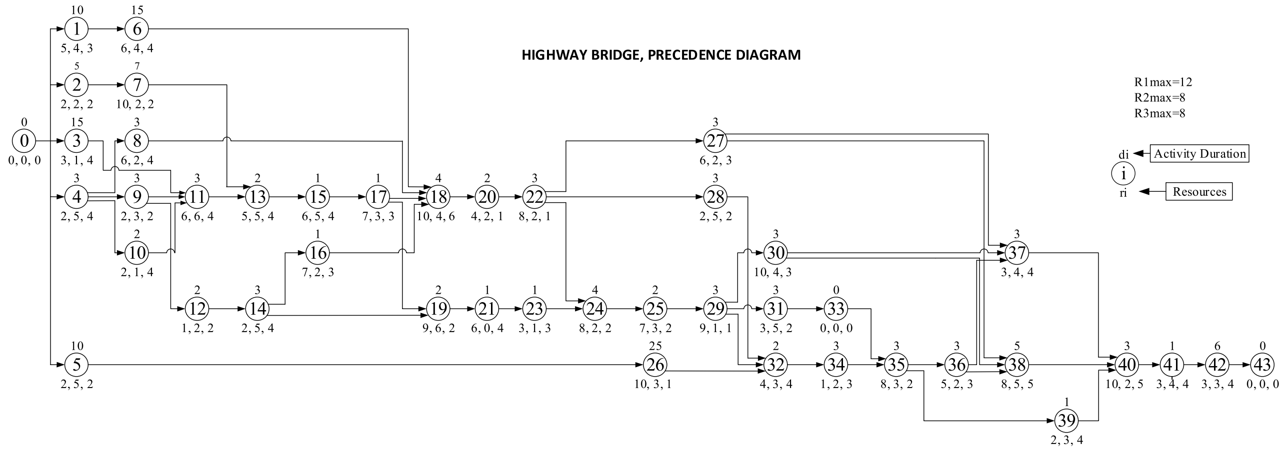

Fig. 6. Network of highway bridge project.

Table 2
Parameter settings of the algorithms.

Algorithm	Abbreviation	Parameter setting
Genetic Algorithm	GA	NP; NE Crossover rate = 0.8 Mutation rate = 0.01 Generation gap = 0.9
Differential Evolution	DE	NP; NE Real constant F = 0.5 Crossover rate = 0.9
Particle Swarm Optimization	PSO	NP; NE Cognitive component = 1.8 Social component = 1.8 Inertia weight = 0.6
Artificial Bee Colony	ABC	NP; NE Trial limit = NP * dimensions
Firefly Algorithm	FA	NP; NE Attractiveness parameter: βmin = 1 Absorption coefficient: $\gamma = 0.1$ Randomness of firefly movement: $\alpha = 0.2$
Whale Optimization Algorithm	WOA	NP; NE Fluctuation range a: decreased from 2 to 0 Coefficient of the logarithmic spiral shape: b = 1
Flower Pollination Algorithm	FPA	NP; NE Proximity probability: p = 0.8
Teaching–Learning-Based Optimization	TLBO	NP; NE
Water Cycle Algorithm	WCA	NP; NE Coefficient of the intensity near the sea: $d_{max} = 1E{-}3$ Number of rivers: $N_{sr} = 8$
Symbiotic Organisms Search	SOS	NP; NE
Electromagnetic Field Optimization	EFO	NP; NE Portion of the population of the positive field: P_field = 0.05 Portion of the population of the negative field: N_field = 0.4 Probability of selecting electromagnets: Ps_rate = 0.1 Possibility of changing one electromagnet: R_rate = 0.1
Gravitation Search Algorithm	GSA	NP; NE Power of R = 0.7 Rnorm = 0.2
Forensic-Based Investigation	FBI	NP; NE

Note: NP: number of populations; NE: number of evaluations.

The figure is marked with the duration of each activity above the corresponding circle node. The amounts of required resources are indicated below the circle node. The precedence constraints among activities are represented by arrows.

3.3. IEEE Congress on Evolutionary Computation

The annual IEEE CEC is one of the leading events in the area of evolutionary computation. In this study, the performance of the proposed FBI is evaluated using a set of problems presented in the IEEE CEC-2017 competition on bound-constrained problems with ten dimensions. This set of benchmarks consists of 30 test functions with a diverse set of characteristics. In summary, functions 1 to 3 are unimodal; functions 4 to 10 are multimodal; functions 11 to 20 are hybrid functions; and functions 21 to 30 are composition functions. More details can be found elsewhere [47].

The performance level of FBI was compared with those of jSO [48] and EBO with CMAR [49], which were the winners of the CEC-2017 competition, and that of LSHADE [50], which was the best-ranked DE-based algorithm in CEC-2014. The results of

Table 3
Results of solving unimodal and separable problems (F1 to F5).

Algorithm	Index	F1 min = 0	F2 min = 0	F3 min = 0	F4 min = 0	F5 min = 0
GA	Mean	**0**	1.17E+03	1.11E+03	1.48E+02	1.81E-01
	Std.	0	7.66E+01	7.42E+01	1.24E+01	2.71E-02
	t(s)	2.99	3.11	3.28	2.95	4.30
PSO	Mean	**0**	**0**	**0**	**0**	1.16E-03
	Std.	0	0	0	0	2.76E-04
	t(s)	2.65	2.76	2.91	2.62	3.81
DE	Mean	**0**	**0**	**0**	**0**	1.36E-03
	Std.	0	0	0	0	4.17E-04
	t(s)	2.75	2.85	3.01	2.71	3.94
ABC	Mean	**0**	**0**	**0**	**0**	3.00E-02
	Std.	0	0	0	0	4.87E-03
	t(s)	3.34	3.09	3.28	3.11	4.31
FA	Mean	2.50E+01	**0**	7.43E-06	4.20E-06	8.99E-03
	Std.	0	0	1.38E-06	1.26E-06	2.21E-03
	t(s)	12.58	13.08	13.80	12.41	18.06
WOA	Mean	**0**	**0**	**0**	**0**	2.05E-04
	Std.	0	0	0	0	2.29E-04
	t(s)	1.48	1.66	1.91	1.75	2.77
FPA	Mean	2.50E+01	**0**	7.16E-11	**0**	1.06E-02
	Std.	0	0	3.92E-10	0	8.55E-03
	t(s)	3.55	3.69	3.89	3.50	5.09
TLBO	Mean	**0**	**0**	**0**	**0**	1.01E-04
	Std.	0	0	0	0	3.56E-05
	t(s)	5.28	5.20	5.83	5.55	7.45
WCA	Mean	**0**	1.12E+01	**0**	1.11E-10	5.65E-01
	Std.	0	5.41E+00	0	1.44E-10	2.75E-01
	t(s)	5.33	5.25	5.88	5.60	7.52
SOS	Mean	**0**	**0**	**0**	**0**	6.72E-05
	Std.	0	0	0	0	2.09E-05
	t(s)	0.11	0.16	0.31	0.28	9.53
EFO	Mean	**0**	**0**	8.23E-06	**0**	4.21E-04
	Std.	0	0	8.35E-05	0	3.59E-04
	t(s)	0.05	0.16	0.22	0.11	0.05
GSA	Mean	**0**	**0**	**0**	**0**	1.15E-02
	Std.	0	0	0	0	2.60E-03
	t(s)	11.19	23.89	24.30	24.58	25.50
FBI	Mean	**0**	**0**	**0**	**0**	**9.29E-07**
	Std.	0	0	0	0	4.88E-07
	t(s)	0.86	1.14	1.33	1.05	1.98

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

those three compared algorithms were taken directly from the literature [48,49].

3.4. Parameter settings in optimization algorithms

The parameters in optimization algorithms significantly affect their performance. FBI has only two parameters, namely the number of populations and number of iterations. Because no rule governs the size of the population, it was experimentally determined to be from 50 up to 5D (for $D > 10$), depending on the search space and the dimensions of the optimization problem.

The results obtained when GA, DE, PSO, and ABC were applied to the 50 benchmark problems were directly taken from Karaboga and Akay (2009) [33]. In their study, the population size was set to 50 and the maximum number of evaluations was set to 500,000 for all algorithms. The maximum number of evaluations, or "the maximum number of fitness function evaluations", is the maximum number of times that the objective function is evaluated in the metaheuristic algorithm [51].

Clearly, the values of the parameters that were common to all algorithms, such as population size and number of evaluations, must be equal to ensure a fair comparison. For solving 50 benchmark problems, because the results obtained using the four aforementioned algorithms were taken directly from Karaboga and Akay (2009) [33], the population size was set to 50 and the maximum number of evaluations was set to 500,000 for all 13 algorithms (Table 2). All algorithms were run 30 times on each of the 50 benchmark functions.

In Tran et al. (2015) [26], where GA, DE, and PSO solved the RCPSP, both the size of population and the number of iterations were set to 100. Because using the same number of iterations is not as fair as using the same number of evaluations, in this experiment, the population size was set to 100 and the maximum number of evaluations was set to 10,000 for all 13 algorithms. The results were obtained after 30 independent runs.

Table 2 presents the set values of the other parameters of FA, WOA, FPA, TLBO, WCA, SOS, EFO, and GSA algorithms for solving 50 benchmark problems and the RCPSP in the context of previously published studies: Chou and Ngo (2017) [52], Mirjalili and Lewis (2016) [5], Yang 2012 [20], Rao et al. (2011) [53], Eskandar et al. (2012) [21], Cheng and Prayogo (2014) [23], Abedinpourshotorban et al. (2016) [24], and Rashedi et al. (2009) [25].

For solving CEC-2017 benchmark functions, the parameter settings for FBI were the same as those used in all three compared algorithms: 51 runs were performed for each function and the maximum number of evaluations was 100,000 (for a problem with 10 dimensions).

3.5. Metaheuristic optimization platform

A metaheuristic optimization platform was developed to make the comparison of algorithms much easier and faster and to provide performance indicators clearly, logically, and graphically. Fig. 7 displays the main user interface of this platform. Users can simply click "Benchmark test" or "Practical applications" in the "Objective Functions" box to choose to run any of the 13 algorithms on 50 benchmark functions or a practical application (RCPSP). Fig. 7a illustrates the default interface, after it has been opened by a user.

As depicted in Fig. 7, a user can choose to run one algorithm. Then, its avatar image and all of its tuning parameters pop up with default values, but the user is allowed to modify them (Fig. 7b). To run a benchmark test, the user can choose a benchmark function, its lower bound, its upper bound, and the number of its dimensions (Fig. 7c). The input values will pop up with default values after the user has entered the number of the benchmark function; the user can modify these also. Similarly, to run a practical application, a user is only required to enter its name (Fig. 7d). After the input parameters have been defined, as soon as the user clicks the "run" button, the problem is optimized (regardless of whether it is a benchmark problem or a practical application).

After the optimization process has been executed, the software plots the convergence curve, displays the computation time (unit: second), updates the obtained best optimum (mean and standard variation of *n* run times), and presents the variable values in the "result" section on the interface. Here, the run time is the number of trials; $\{X_1, X_2, \ldots, X_i\}$ are solutions and Y is the optimal solution of the objective function of each trial. All results can be exported to Excel files by clicking on the "save" button on the interface (Fig. 7e, f).

4. Experimental results

The first part of this section presents the analysis of the experimental results obtained using 50 benchmark problems, divided into four subsets, namely unimodal-and-separable problems, unimodal-and-nonseparable problems, multimodal-and-separable problems, and multimodal-and-nonseparable problems. Then, the overall performance of FBI in solving these 50 benchmark problems is discussed. Any value less than 1E−12 is assumed to be 0 [33].

The second part of this section provides an analysis of the effectiveness of FBI and other algorithms for solving the RCPSP for a real construction project. In the third part, FBI is evaluated using a set of problems presented in the IEEE CEC-2017's competition on bound-constrained problems. The final part addresses the ability of FBI and its components for solving high-dimensional problems.

4.1. 50 benchmark functions

4.1.1. Unimodal and separable problems (functions 1 to 5)

Functions F1 to F5 are unimodal because each of them has only one global optimum. They are also separable. A p-variable separable function can be expressed as the sum of p single-variable functions. These functions serve as tests of the exploitation abilities of the investigated metaheuristic algorithms. Table 3 reveals that FBI outperformed all other metaheuristic algorithms. In particular, it was the most efficient optimizer for all unimodal and separable functions (F1 to F5). FBI found the global minima of four out of five functions; this result is excellent, because no algorithm could discover the global minimum of F5.

4.1.2. Unimodal and nonseparable problems (functions 6 to 17)

A p-variable separable function can be expressed as the sum of p single-variable functions, whereas nonseparable functions cannot be expressed in this form. The variables of nonseparable functions are interrelated. Therefore, discovering the global minimum of a nonseparable function is more difficult than discovering the global minimum of a separable function [54].

F6 to F17 are nonseparable and unimodal functions. Table 4 presents the performance levels of the algorithms when applied to those functions. FBI, SOS, and TLBO performed the best overall and most efficiently identified the global minima of 10 of 12 functions.

4.1.3. Multimodal and separable problems (functions 18 to 26)

Unlike unimodal functions, multimodal functions have many local optima whose number increases exponentially with the problem size (number of design variables). Therefore, such problems are notably useful for evaluating the exploratory capacity of an optimization algorithm [8,55]. If the exploratory process of an algorithm is so poor that it cannot search the whole space efficiently, then the algorithm will become stuck at a local minimum [56]. Functions 18 to 26 represent multimodal and separable problems. The results in Table 5 for functions F18 to F26 indicate that FBI has excellent exploratory capacity. FBI and ABC were the most efficient algorithms in the majority of the test problems; they also found the global minima of all nine functions.

4.1.4. Multimodal and nonseparable problems (functions 27 to 50)

These multimodal and nonseparable functions present the hardest tests for optimization algorithms. If an algorithm cannot explore the search space properly and cannot adapt to the changes in direction of functions that have a narrow curving valley, then it will fail at those challenges in these types of problems. Moreover, as mentioned in the previous section, nonseparable functions are more difficult than separable functions.

Table 6 provides the performance levels of 13 algorithms applied to F27–F50. The results indicate that FBI outperformed all other metaheuristic algorithms. FBI yielded the best results for all 24 multimodal and nonseparable functions because it achieved the global minima of 22 functions, whereas the SOS returned the best results for only 18 of 24 functions. The comparative result (Table 6) clearly reveals a remarkable gap of performance levels between FBI and other algorithms in solving the multimodal and nonseparable problems.

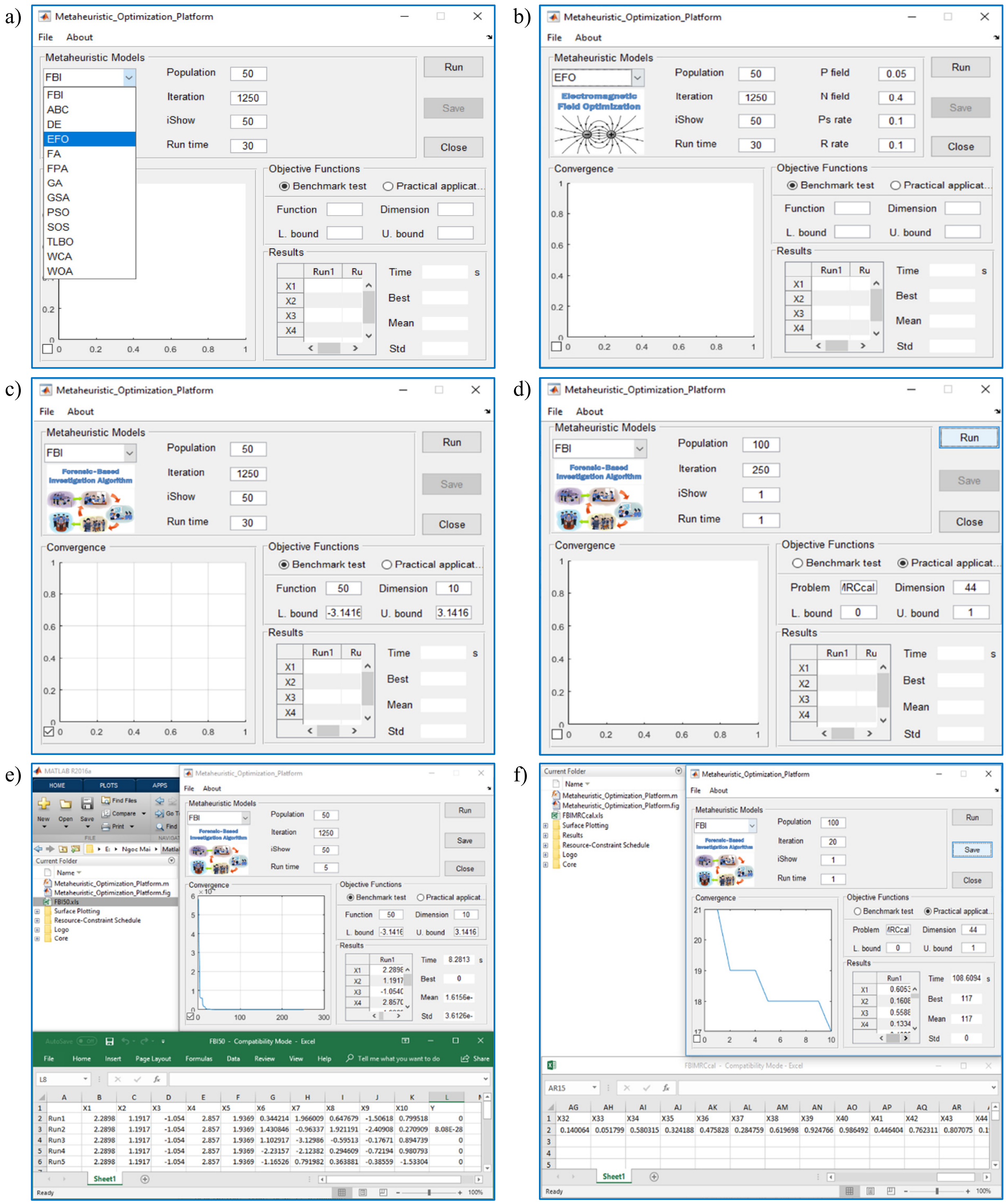

Fig. 7. Metaheuristic optimization platform.

4.1.5. Performance and convergence comparison of algorithms

A set of 50 well-known benchmark functions was used to test comprehensively the efficiency of FBI. As listed in Tables 3–6, FBI found the global minima for 45 out of 50 functions. Besides, there are still three functions that are relatively difficult to find the optimal minimum values, but FBI's values were smaller than those of other optimizers. Those functions were Quartic (F5), Perm (F37), and Power Sum (F38). A box plot analysis was conducted to compare the efficiency of FBI with those of four top-performing algorithms (Fig. 8).

Box plots constitute a standardized approach to displaying the distributions of data [57,58]. It can be observed that FBI's box plots were remarkably narrow for F5, F37, and F38. Narrow box plots suggest that the obtained solutions possessed high levels of agreement with each other. Therefore, for the aforementioned three functions, solution quality from FBI was superior to those from the other algorithms. The interquartile range and median of the proposed FBI were also superior to those of the others. We thus conclude that the FBI remarkably outperformed the

Table 4a
Results of solving unimodal and nonseparable problems (F6 to F11).

Algorithm	Index	F6 min = 0	F7 min = -1	F8 min = 0	F9 min = 0	F10 min = -50	F11 min=-210
GA	Mean	**0**	**-1**	**0**	1.49E-02	-49.9999	-209.476
	Std.	0	0	0	7.36E-03	2.25E-05	1.93E-01
	t(s)	2.95	2.70	2.47	2.53	2.61	2.70
PSO	Mean	**0**	**-1**	**0**	**0**	**-50**	**-210**
	Std.	0	0	0	0	0	0
	t(s)	2.62	2.39	2.18	2.24	2.31	2.39
DE	Mean	**0**	**-1**	**0**	4.09E-02	**-50**	**-210**
	Std.	0	0	0	8.20E-02	0	0
	t(s)	2.71	2.48	2.26	2.32	2.40	2.48
ABC	Mean	**0**	**-1**	**0**	9.30E-02	**-50**	**-210**
	Std.	0	0	0	6.63E-02	0	0
	t(s)	2.95	3.02	2.91	3.05	3.19	3.17
FA	Mean	**0**	**-1**	**0**	2.53E-08	**-50**	**-210**
	Std.	0	1.24E-10	0	9.73E-09	6.48E-09	8.02E-07
	t(s)	12.41	11.36	10.36	10.64	10.97	11.36
WOA	Mean	**0**	**-1**	**0**	1.24E-01	**-50**	**-210**
	Std.	0	6.41E-11	0	8.90E-02	3.84E-08	2.30E-05
	t(s)	1.16	1.17	1.09	1.13	1.20	1.30
FPA	Mean	**0**	**-1**	**0**	**0**	**-50**	**-210**
	Std.	0	0	0	0	0	0
	t(s)	3.50	3.20	2.92	3.00	3.09	3.20
TLBO	Mean	**0**	**-1**	**0**	**0**	**-50**	**-210**
	Std.	0	0	0	0	0	0
	t(s)	4.69	4.67	4.58	4.61	4.75	4.83
WCA	Mean	**0**	**-1**	**0**	1.34E-06	**-50**	**-210**
	Std.	0	0	0	1.08E-06	0	3.03E-10
	t(s)	4.73	4.71	4.62	4.65	4.79	4.87
SOS	Mean	**0**	**-1**	**0**	**0**	**-50**	**-210**
	Std.	0	0	0	0	0	0
	t(s)	0.23	0.09	0.16	3.25	0.13	0.11
EFO	Mean	2.21E-02	**-1**	**0**	9.11E-02	**-50**	-209.9998
	Std.	1.28E-01	0	0	4.62E-01	0	5.25E-04
	t(s)	0.03	0.06	0.06	0.08	0.09	0.11
GSA	Mean	**0**	-0.9608	**0**	8.37E-03	**-50**	**-210**
	Std.	0	1.26E-01	0	3.27E-02	0	0
	t(s)	9.34	9.25	9.23	10.41	11.44	13.56
FBI	Mean	**0**	**-1**	**0**	**0**	**-50**	**-210**
	Std.	0	0	0	0	0	0
	t(s)	0.48	0.64	0.55	0.47	0.55	0.59

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

other algorithms even for benchmark functions that are relatively difficult to optimize.

Table 7 presents a summary of all comparative models' performance levels on 50 benchmark functions. The results demonstrate that FBI was the most efficient optimizer, outperforming the other algorithms in all domains when applied to unimodal, multimodal, separable, and nonseparable problems. FBI performed best for 48 out of 50 functions (96%) and found the global minima of 45 of 50 functions (90%). The second-best algorithm (SOS) performed best for 39 out of 50 functions (68%), and found the global minima for 39 out of 50 functions (68%).

Table 7 indicates that FBI had short computational times. The total computational time of FBI with 50 benchmark functions was 65 seconds, making it the second-fastest algorithm. FBI thus not only delivers excellent solutions but also is fast and effective.

Twelve benchmark functions (Step, Sphere, Sum Squares, Quartic, Rastrigin, Schwefel, Michalewicz10, Schaffer, 6 Hump Camel Back, Griewank, Ackley, and Penalized) were used to compare the convergence levels of the algorithms because they are the most common benchmark functions for evaluating metaheuristics [59]. On the basis of Table 7, five best algorithms were

Table 4b
Results of solving unimodal and nonseparable problems (F12 to F17).

Algorithm	Index	F12 min = 0	F13 min = 0	F14 min = 0	F15 min = 0	F16 min = 0	F17 min=0
GA	Mean	1.34E-02	9.70E+00	1.10E+01	7.40E+03	1.96E+05	1.22E+03
	Std.	4.53E-03	1.55E+00	1.39E+00	1.14E+03	3.85E+04	2.66E+02
	t(s)	2.85	3.44	3.05	3.64	2.93	2.90
PSO	Mean	**0**	1.10E-04	**0**	**0**	1.51E+01	6.67E-01
	Std.	0	1.60E-04	0	0	2.42E+01	10E-8
	t(s)	2.52	3.05	2.70	3.22	2.59	2.57
DE	Mean	**0**	2.17E-07	**0**	**0**	1.82E+01	6.67E-01
	Std.	0	1.36E-07	0	0	5.04E+00	10E-9
	t(s)	2.61	3.16	2.79	3.34	2.69	2.66
ABC	Mean	2.48E-04	3.13E-03	**0**	**0**	8.88E-02	**0**
	Std.	1.83E-04	5.03E-04	0	0	7.74E-02	0
	t(s)	3.13	3.59	3.13	3.94	3.13	3.14
FA	Mean	**0**	2.19E-01	1.19E-03	2.54E-04	1.74E+01	8.39E-01
	Std.	0	5.31E-02	1.81E-04	6.93E-05	7.48E+00	8.54E-02
	t(s)	11.97	14.46	12.80	15.29	12.30	12.19
WOA	Mean	**0**	6.10E-07	**0**	**0**	2.46E+01	6.67E-01
	Std.	0	1.28E-06	0	0	3.94E-01	5.12E-06
	t(s)	1.38	2.03	1.64	4.61	1.66	1.63
FPA	Mean	**0**	**0**	1.17E-01	**0**	1.90E+01	6.67E-01
	Std.	0	0	1.51E-01	0	1.10E+01	0
	t(s)	3.38	4.08	3.61	4.31	3.47	3.44
TLBO	Mean	**0**	1.03E-06	**0**	**0**	**6.18E-04**	6.67E-01
	Std.	0	2.69E-06	0	0	1.79E-03	0
	t(s)	5.17	6.16	5.33	13.45	5.27	5.20
WCA	Mean	2.31E-10	4.05E-04	3.33E+00	6.08E-11	4.11E+00	6.44E-01
	Std.	3.21E-10	7.69E-05	5.46E+00	7.66E-11	4.93E+00	1.22E-01
	t(s)	5.22	6.21	5.38	13.57	5.31	5.25
SOS	Mean	**0**	**0**	**0**	**0**	3.34E-01	6.67E-01
	Std.	0	0	0	0	1.52E-01	0
	t(s)	0.33	8.52	0.42	0.34	7.58	7.64
EFO	Mean	**0**	2.16E-02	**0**	**0**	7.79E+01	**0**
	Std.	0	1.04E-02	0	0	6.72E+01	0
	t(s)	0.14	0.17	0.19	0.16	0.09	0.11
GSA	Mean	**0**	1.94E-04	1.51E-08	**0**	2.04E+01	6.67E-01
	Std.	0	6.73E-05	2.33E-09	0	2.20E-01	0
	t(s)	13.64	21.30	24.20	24.77	24.47	24.47
FBI	Mean	**0**	**0**	**0**	**0**	1.79E+01	6.67E-01
	Std.	0	0	0	0	1.82E-01	0
	t(s)	0.72	1.22	0.83	2.02	0.97	0.86

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

chosen for the purpose: FBI, TLBO, FPA, SOS, and ABC. Tables 3–6 indicate that although FBI, TLBO, FPA, SOS, and ABC were all excellent algorithms, FBI was the most efficient optimizer, performing optimally with all 12 benchmark functions. ABC, TLBO, SOS, and FPA found the global minima of 11, 10, 8, and 3 of the 12 functions, respectively.

To provide the fairest possible comparison, the population size of all models was set to 50 with the same initial population position; the maximum number of evaluations was 500,000 for all models for all functions. The convergence line charts are illustrated in Fig. 9. Each convergence line chart depicts a different and only small range of evaluation numbers to provide a clear visualization.

Fig. 9 reveals that FBI usually not only found the global minima much earlier than other algorithms but also converged much faster. This result was confirmed by the convergence line charts for ten functions: Step, Sphere, Sum Squares, Quartic, Rastrigin, Schaffer, 6 Hump Camel Back, Griewank, Ackley, and Penalized. In particular, for the Step, Sphere, Sum Squares, Rastrigin, Schaffer, Griewank, and Ackley functions, FBI converged much faster than the others did. The FBI had a slower convergence with only two functions, Schwefel and Michalewicz10.

Table 5a
Results of solving multimodal and separable problems (F18 to F22).

Algorithm	Index	F18 min = 0.998	F19 min = 0.398	F20 min = 0	F21 min = 0	F22 min = 0
GA	Mean	**0.9980**	**0.3979**	**0**	**0**	5.29E+01
	Std.	0	0	0	0	4.56E+00
	t(s)	4.26	2.58	2.60	2.61	3.35
PSO	Mean	**0.9980**	**0.3979**	**0**	**0**	4.40E+01
	Std.	0	0	0	0	1.17E+01
	t(s)	3.77	2.29	2.30	2.31	2.97
DE	Mean	**0.9980**	**0.3979**	**0**	**0**	1.17E+01
	Std.	0	0	0	0	2.54E+00
	t(s)	3.91	2.37	2.38	2.40	3.07
ABC	Mean	**0.9980**	**0.3979**	**0**	**0**	**0**
	Std.	0	0	0	0	0
	t(s)	5.05	2.98	3.08	2.98	3.28
FA	Mean	12.6705	**0.3979**	**0**	**0**	3.33E+01
	Std.	3.80E-10	0	0	0	2.98E+00
	t(s)	17.90	10.86	10.92	10.97	10.97
WOA	Mean	1.0641	**0.3979**	**0**	1.56E-07	**0**
	Std.	3.62E-01	0	0	1.37E-07	0
	t(s)	3.13	1.14	1.13	1.08	1.73
FPA	Mean	12.6705	**0.3979**	**0**	**0**	7.77E+01
	Std.	0	0	0	0	1.53E+01
	t(s)	5.05	3.06	3.08	3.09	3.97
TLBO	Mean	**0.9980**	**0.3979**	**0**	**0**	8.94E+00
	Std.	0	0	0	0	6.48E+00
	t(s)	5.41	6.30	4.98	5.28	6.16
WCA	Mean	**0.9980**	**0.3979**	**0**	**0**	9.66E+01
	Std.	0	0	0	0	3.70E+01
	t(s)	5.45	6.35	5.03	5.33	6.21
SOS	Mean	**0.9980**	**0.3979**	**0**	**0**	**0**
	Std.	0	0	0	0	0
	t(s)	10.69	6.83	0.16	0.25	0.39
EFO	Mean	1.0033	**0.3979**	2.92E-04	**0**	9.55E-03
	Std.	1.30E-01	0	1.12E-02	0	9.70E-02
	t(s)	0.14	0.03	0.06	0.03	0.08
GSA	Mean	1.6330	**0.3979**	**0**	**0**	1.29E+01
	Std.	7.03E-01	0	0	0	3.28E+00
	t(s)	11.31	9.34	9.36	9.30	24.31
FBI	Mean	**0.9980**	**0.3979**	**0**	**0**	**0**
	Std.	0	0	0	0	0
	t(s)	2.11	0.47	0.56	0.45	1.28

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

4.2. Project scheduling problem

To evaluate the stability and accuracy of each algorithm, optimization performance is expressed in terms of the best result found (best), the average result (mean), the standard deviation results (std.), and the worst result (worst) after 30 independent runs for the RCPSP problem (Table 8). The best and worst results indicate the ability of the algorithm to discover the optimal solution. The mean and the standard deviation describe solution quality. The success rate is the frequency with which each algorithm found the optimal solution.

Table 8 indicates that all algorithms found the optimal solution in every case, but only FBI and SOS had 100% success rates. On the basis of the mean values, the top five algorithms for application to the case study were selected, namely FBI, SOS, TLBO, FPA, and ABC. These were the same five that had yielded the best five for solving 50 benchmark functions. Fig. 10 depicts the best project duration corresponding to the number of evaluations for the top three algorithms: FBI, SOS, and TLBO. Fig. 10 demonstrates that FBI found the optimal solution in fewer evaluations than the other algorithms did. Therefore, FBI is the most efficient optimizer among the compared algorithms when applied to this case study.

Table 5b
Results of solving multimodal and separable problems (F23 to F26).

Algorithm	Index	F23 min = -12569.5	F24 min = -1.8013	F25 min = -4.6877	F26 min = -9.6602
GA	Mean	-11593.4	**-1.8013**	-4.6448	-9.4968
	Std.	9.33E+01	0	9.79E-02	1.41E-01
	t(s)	3.60	2.76	2.93	3.32
PSO	Mean	-6909.1359	-1.5729	-2.4909	-4.0072
	Std.	4.58E+02	1.20E-01	2.57E-01	5.03E-01
	t(s)	3.19	2.44	2.59	2.94
DE	Mean	-10266	**-1.8013**	-4.6835	-9.5912
	Std.	5.22E+02	0	1.25E-02	6.42E-02
	t(s)	3.30	2.53	2.69	3.05
ABC	Mean	**-12569.49**	**-1.8013**	**-4.6877**	**-9.6602**
	Std.	0	0	0	0
	t(s)	3.75	3.33	3.50	3.75
FA	Mean	-9172.218	**-1.8013**	-4.6877	-9.4633
	Std.	2.65E+02	0	6.28E-09	7.87E-02
	t(s)	15.13	11.58	12.30	13.96
WOA	Mean	-12460.300	**-1.8013**	-4.1474	-6.9109
	Std.	2.54E+02	1.57E-10	4.56E-01	7.93E-01
	t(s)	2.09	1.30	1.50	1.44
FPA	Mean	-9681.0753	**-1.8013**	-4.6876	-8.4033
	Std.	2.88E+02	0	1.65E-05	3.30E-01
	t(s)	4.27	3.27	3.47	3.94
TLBO	Mean	-8499.0341	**-1.8013**	-4.6396	-9.3335
	Std.	7.53E+02	0	6.91E-02	2.29E-01
	t(s)	4.78	4.92	5.28	6.16
WCA	Mean	-8351.68	**-1.8013**	-4.5663	-7.2813
	Std.	7.70E+02	0	1.13E-01	7.10E-01
	t(s)	4.82	4.97	5.33	6.21
SOS	Mean	**-12569.49**	**-1.8013**	**-4.6877**	-9.6598
	Std.	0	0	0	1.25E-03
	t(s)	0.13	0.11	0.11	0.13
EFO	Mean	-12099.76	**-1.8013**	-4.6590	-9.4276
	Std.	6.21E+02	0	5.05E-02	1.54E-01
	t(s)	0.14	0.05	0.08	0.05
GSA	Mean	-2795.589	**-1.8013**	-4.5581	-9.2534
	Std.	5.31E+02	0	8.21E-02	1.75E-01
	t(s)	24.50	9.31	11.16	13.95
FBI	Mean	**-12569.49**	**-1.8013**	**-4.6877**	**-9.6602**
	Std.	0	0	0	0
	t(s)	1.19	0.70	0.88	1.11

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

4.3. CEC-2017 hybrid metaheuristics and benchmark functions

Table 9 provides the statistical outcomes of the four algorithms (EBO with CMAR, jSO, LSHADE, and FBI) for 30 benchmark functions with $D = 10$.

A hypothesis test was conducted to compare FBI with three algorithms for solving the 30 CEC-2017 benchmark problems. The following hypotheses were tested using a one-tailed t test with equal sample sizes and with unequal and unknown variances:

H_0: FBI and one algorithm (among four) perform equally.

H_1: FBI significantly outperforms any other algorithm.

$$\text{Degree of freedom}: \upsilon = \frac{\left(s_1^2/n_1 + s_2^2/n_2\right)^2}{\frac{\left(s_1^2/n_1\right)^2}{n_1-1} + \frac{\left(s_2^2/n_2\right)^2}{n_2-1}}$$

$$\text{Statistical test}: t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$

where n_1 and n_2 are numbers of experimental runs; υ is the degrees of freedom used in the test; s_1^2 and s_2^2 are the unbiased

Table 6a
Results of solving multimodal and nonseparable problems (F27 to F32).

Algorithm	Index	F27 min = 0	F28 min = -1.03163	F29 min = 0	F30 min = 0	F31 min = -186.73	F32 min = 3
GA	Mean	4.24E-03	**-1.0316**	6.83E-02	**0**	**-186.731**	5.2506
	Std.	4.76E-03	0	7.82E-02	0	0	5.8701
	t(s)	2.60	2.69	2.57	2.58	2.77	2.54
PSO	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	0	0	0	0	0	0
	t(s)	2.30	2.38	2.28	2.29	2.45	2.25
DE	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	0	0	0	0	0	0
	t(s)	2.38	2.47	2.36	2.37	2.54	2.33
ABC	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	0	0	0	0	0	0
	t(s)	3.08	3.33	3.00	2.98	3.28	2.98
FA	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	84
	Std.	0	0	0	0	1.17E-09	4.27E-11
	t(s)	10.92	11.30	10.80	10.86	11.64	10.69
WOA	Mean	1.46E-03	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	7.97E-03	0	0	0	2.87E-08	7.75E-08
	t(s)	1.14	1.30	1.11	1.16	1.23	1.14
FPA	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	84
	Std.	0	0	0	0	0	0
	t(s)	3.08	3.19	3.05	3.06	3.28	3.02
TLBO	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	0	0	0	0	0	0
	t(s)	4.78	4.92	4.59	4.56	5.00	4.77
WCA	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	0	0	0	0	0	0
	t(s)	4.82	4.97	4.63	4.60	5.04	4.81
SOS	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	0	0	0	0	0	0
	t(s)	0.42	0.13	0.17	0.19	0.09	6.50
EFO	Mean	1.98E-02	**-1.0316**	**0**	4.53E-05	**-186.731**	3.1998
	Std.	2.17E-02	0	0	3.20E-03	9.01E-03	2.67E+00
	t(s)	0.06	0.06	0.06	0.05	0.05	0.05
GSA	Mean	1.03E-02	**-1.0316**	**0**	**0**	-186.3053	**3**
	Std.	1.07E-02	0	0	0	5.00E-01	0
	t(s)	9.38	9.42	9.34	9.34	9.38	9.28
FBI	Mean	**0**	**-1.0316**	**0**	**0**	**-186.731**	**3**
	Std.	0	0	0	0	0	0
	t(s)	0.69	0.56	0.55	0.58	0.59	0.52

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

estimators of the variances of the two samples; the denominator of t is the standard error of the difference between two means $\bar{x}_1$ and $\bar{x}_2$ (average). A confidence level of 95% was used for this test ($\alpha = 0.05$).

Table 10 presents the results of the hypothesis testing for FBI and the three winners of the CEC competition. Although FBI slightly underperformed both EBO with CMAR and jSO, it was highly competitive with LSHADE. Notably, all the winners of the CEC competition (EBO with CMAR, jSO, and LSHADE) are hybrid metaheuristic models, which improve upon the effectiveness of conventional models.

4.4. FBI component analysis for solving high-dimensional problems

The experiment discussed in Section 4.1 confirmed that FBI outperformed all other algorithms for solving 50 common benchmark problems. Among the tested 50 benchmark functions, 14 functions had $D = 30$ (D: number of dimensions), as presented in Table 1. This section addresses the ability of FBI to solve high-dimensional problems when the number of dimensions of those 14 functions are increased from $D = 30$ to $D = 1000$. Those functions are as follows: Step, Sphere, Sum Squares, Quartic, Schwefel 2.22, Schwefel 1.2, Rosenbrock, Dixon–Price, Rastrigin,

Table 6b
Results of solving multimodal and nonseparable problems (F33 to F38).

Algorithm	Index	F33 min = 0.00031	F34 min = -10.15	F35 min = -10.4	F36 min = -10.53	F37 min = 0	F38 min = 0
GA	Mean	0.00562	-5.6605	-5.3441	-3.8298	3.03E-01	1.04E-02
	Std.	8.17E-03	3.8667	3.5171	2.4520	1.93E-01	9.08E-03
	t(s)	2.72	4.60	4.81	4.64	3.35	3.02
PSO	Mean	0.00049	-2.0870	-1.9899	-1.8797	3.61E-02	1.14E+01
	Std.	3.66E-04	1.1785	1.4206	0.4325	4.89E-02	7.36E+00
	t(s)	2.41	4.08	4.26	4.11	2.97	2.68
DE	Mean	0.00043	**-10.1532**	**-10.4029**	**-10.5364**	2.40E-02	1.43E-04
	Std.	2.73E-04	0	0	0	4.60E-02	1.45E-04
	t(s)	2.49	4.22	4.42	4.26	3.07	2.77
ABC	Mean	0.00043	**-10.1532**	**-10.4029**	**-10.5364**	4.11E-02	2.95E-03
	Std.	6.04E-05	0	0	0	2.31E-02	2.29E-03
	t(s)	3.16	5.63	5.63	5.69	3.84	3.52
FA	Mean	**0.00031**	**-10.1532**	**-10.4029**	**-10.5364**	8.01E-04	1.29E-05
	Std.	1.28E-05	1.28E-08	1.35E-08	1.23E-08	5.88E-04	1.51E-05
	t(s)	11.41	19.34	20.22	19.50	14.07	12.69
WOA	Mean	0.00042	**-10.1532**	**-10.4029**	**-10.5364**	8.08E-01	3.44E-01
	Std.	2.39E-04	9.38E-07	6.61E-06	2.35E-06	8.33E-01	3.55E-01
	t(s)	1.31	3.61	3.83	3.41	0.91	1.63
FPA	Mean	**0.00031**	**-10.1532**	**-10.4029**	**-10.5364**	3.01E-03	3.06E-04
	Std.	0	0	0	0	3.06E-03	3.30E-04
	t(s)	3.22	5.45	5.70	5.50	3.97	3.58
TLBO	Mean	**0.00031**	**-10.1532**	-10.2258	-10.3561	2.14E-03	4.48E-05
	Std.	0	0	9.70E-01	9.87E-01	1.96E-03	7.50E-05
	t(s)	5.13	11.27	10.25	10.00	6.30	5.63
WCA	Mean	0.00037	-9.1397	-9.8755	-10.3561	3.43E-02	2.12E-04
	Std.	2.32E-04	2.06E+00	1.61E+00	9.87E-01	9.14E-02	2.02E-04
	t(s)	5.17	11.37	10.34	10.09	6.35	5.67
SOS	Mean	**0.00031**	-9.3035	**-10.4029**	**-10.5364**	5.64E-03	6.47E-05
	Std.	0	1.93E+00	0	0	2.33E-02	7.05E-05
	t(s)	7.31	0.13	0.16	0.11	8.36	8.14
EFO	Mean	0.0012	-5.633	-7.0513	-7.0622	2.73E-01	2.75E-02
	Std.	3.75E-03	3.47E+00	3.59E+00	3.75E+00	6.78E-01	4.93E-02
	t(s)	0.05	0.16	0.16	0.14	0.06	0.05
GSA	Mean	0.0013	-7.3515	**-10.4029**	**-10.5364**	3.33E+00	1.65E-02
	Std.	3.31E-04	2.85E+00	0	0	1.92E+00	1.68E-02
	t(s)	10.52	12.56	12.63	12.64	11.13	10.86
FBI	Mean	**0.00031**	**-10.1532**	**-10.4029**	**-10.5364**	**2.44E-04**	**2.25E-06**
	Std.	0	0	0	0	3.03E-04	3.12E-06
	t(s)	0.59	2.72	2.63	2.67	1.13	0.81

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

Schwefel, Griewank, Ackley, Penalized, and Penalized 2. Fig. 11 displays three-dimensional plots of those benchmark functions.

The structure of FBI algorithm includes two teams to balance exploration with exploitation. The investigation team (steps A1 and A2) performs the exploration role and finds the suspect in a search space, whereas the pursuit team (steps B1 and B2) performs an exploitation search around the suspected location with the highest probability. The second objective of this subsection is to analyze the effectiveness of the four components of the FBI (*i.e.*, A1, A2, B1, and B2) by evaluating the performance levels of FBI_1, FBI_2, FBI_3 and FBI_4 using the aforementioned 14 benchmark functions with $D = 1000$. FBI_1, FBI_2, FBI_3 and FBI_4 are FBI variants without A1, A2, B1, and B2, respectively. The results are presented in Table 11. The results of using FBI to solve the 14 benchmark functions with $D = 30$ are also presented for comparison.

As seen from Table 11, FBI is effective for solving high-dimensional problems. FBI is as effective at solving problems with 1000 dimensions as 11 of the 14 functions are for solving problems with 30 dimensions; those 11 functions are Step, Sphere, Sum Squares, Quartic, Schwefel 2.22, Schwefel 1.2, Dixon–Price, Rastrigin, Griewank, Penalized, and Penalized 2. FBI

Table 6c
Results of solving multimodal and nonseparable problems (F39 to F44).

Algorithm	Index	F39 min = -3.86	F40 min = -3.32	F41 min = 0	F42 min = 0	F43 min = 0	F44 min = 0
GA	Mean	**-3.863**	-3.2982	1.06E+01	1.47E+01	1.34E+01	1.25E+02
	Std.	0	5.01E-02	1.16E+00	1.78E-01	1.45E+00	1.20E+01
	t(s)	5.91	7.05	3.40	3.68	7.21	5.67
PSO	Mean	-3.6334	-1.8591	1.74E-02	1.65E-01	2.07E-02	7.68E-03
	Std.	1.17E-01	4.40E-01	2.08E-02	4.94E-01	4.15E-02	1.63E-02
	t(s)	5.23	6.25	3.01	3.26	6.39	5.02
DE	Mean	**-3.863**	-3.2169	1.48E-03	**0**	**0**	2.20E-03
	Std.	0	4.76E-02	2.96E-03	0	0	4.40E-03
	t(s)	5.42	6.47	3.12	3.38	6.62	5.20
ABC	Mean	**-3.863**	-3.3220	**0**	**0**	**0**	**0**
	Std.	0	0	0	0	0	0
	t(s)	7.11	8.52	3.67	3.94	7.94	6.16
FA	Mean	**-3.863**	**-3.3224**	1.93E-05	6.38E-04	1.73E+00	6.70E-07
	Std.	0	2.02E-10	3.19E-06	8.11E-05	1.80E-02	9.31E-08
	t(s)	24.82	29.64	14.30	15.46	30.31	23.83
WOA	Mean	**-3.863**	-3.2440	9.14E-04	**0**	2.22E-04	3.02E-03
	Std.	4.87E-05	1.07E-01	3.68E-03	0	1.21E-03	1.65E-02
	t(s)	4.73	6.02	2.14	2.31	6.11	4.39
FPA	Mean	**-3.863**	**-3.3224**	1.78E-02	6.34E-01	1.67E+00	1.59E-05
	Std.	0	0	2.76E-02	3.47E+00	5.38E-07	1.05E-05
	t(s)	7.00	8.36	4.03	4.36	8.55	6.72
TLBO	Mean	**-3.863**	-3.3184	**0**	**0**	3.46E-03	1.63E-02
	Std.	0	2.18E-02	0	0	1.89E-02	3.71E-02
	t(s)	13.05	15.23	6.11	6.64	14.66	11.22
WCA	Mean	**-3.863**	-3.2588	3.55E-02	3.60E+00	3.09E-01	6.04E-03
	Std.	0	6.05E-02	3.96E-02	1.36E+00	8.67E-01	2.30E-02
	t(s)	13.16	15.37	6.16	6.70	14.79	11.32
SOS	Mean	**-3.863**	-3.2389	**0**	**0**	**0**	**0**
	Std.	0	5.76E-02	0	0	0	0
	t(s)	0.14	0.13	0.38	0.48	1.30	1.16
EFO	Mean	**-3.863**	-3.2915	1.96E-03	8.59E-01	3.98E-01	3.57E-01
	Std.	1.89E-02	5.22E-02	5.30E-03	7.52E-01	6.24E-01	1.47E+00
	t(s)	0.20	0.25	0.06	0.17	0.36	0.28
GSA	Mean	**-3.863**	**-3.3224**	2.29E-03	2.42E-09	3.46E-03	**0**
	Std.	0	0	9.15E-03	3.36E-10	1.89E-02	0
	t(s)	13.42	16.16	24.75	24.75	28.36	26.55
FBI	Mean	**-3.863**	**-3.3224**	**0**	**0**	**0**	**0**
	Std.	0	0	0	0	0	0
	t(s)	3.56	4.72	1.44	1.78	5.20	3.45

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

is only less effective for solving $D = 1000$ problems than it is for solving $D = 30$ problems for three functions: Rosenbrock, Schwefel, and Ackley.

Table 11 supports the following conclusions concerning the effectiveness of the four components of FBI for solving high-dimensional problems:

(1) Although the number of dimensions was increased from 30 to 1000 and in each variant, one of the four components of FBI was inactive, FBI remained effective at finding the optimal values for high-dimensional problems. Half of the given functions were solved successfully (Step, Sphere, Sum Squares, Schwefel 2.22, Schwefel 1.2, Rastrigin, and Griewank). Therefore, all components in FBI are efficient; thus, when one is inactive, the others support effective investigation.

(2) FBI_1 was a more efficient optimizer than are FBI_2, FBI_3, and FBI_4. Therefore, A1 is the least effective among the four components of FBI for solving high-dimensional problems.

(3) Both FBI_1 and FBI_2 performed deficiently for some UN and MN functions: They were worse than FBI_3 and FBI_4 for solving a Rosenbrock function; FBI_1 was trapped in a local optimum in an Ackley function, and FBI_2 performs badly in Ackley, Penalized, and Penalized2 functions. Thus,

Table 6d
Results of solving multimodal and nonseparable problems (F45 to F50).

Algorithm	Index	F45 min = -1.08	F46 min = -1.5	F47 min = NA	F48 min = 0	F49 min = 0	F50 min = 0
GA	Mean	**-1.0809**	-0.9684	0.636	**0**	4.30E-03	2.96E+01
	Std.	0	2.88E-01	3.75E-01	0	9.47E-03	1.60E+01
	t(s)	2.78	2.95	3.34	2.83	3.39	3.59
PSO	Mean	-0.6793	-0.5049	-0.0026	**0**	1.46E+03	1.36E+03
	Std.	2.75E-01	2.14E-01	3.52E-03	0	1.27E+03	1.33E+03
	t(s)	2.46	2.62	2.96	2.51	3.00	3.18
DE	Mean	**-1.0809**	**-1.5000**	-1.0528	**0**	5.99E+00	7.82E+02
	Std.	0	0	3.02E-01	0	7.33E+00	1.05E+03
	t(s)	2.55	2.71	3.06	2.60	3.11	3.29
ABC	Mean	**-1.0809**	-0.9382	-0.446	**0**	1.74E-01	8.23E+00
	Std.	0	2.08E-04	1.34E-01	0	6.82E-02	8.09E+00
	t(s)	3.39	3.45	3.80	3.25	3.91	3.92
FA	Mean	**-1.0809**	**-1.5000**	-9.85E-01	5.77E-11	6.92E+02	6.92E+02
	Std.	0	2.78E-10	3.11E-01	3.557E-11	5.17E-07	6.37E-07
	t(s)	11.69	12.41	14.02	11.91	14.24	15.07
WOA	Mean	**-1.0809**	-0.9491	-3.82E-01	4.56E-09	8.18E+02	5.36E+02
	Std.	0	3.70E-01	2.00E-01	1.36E-08	1.32E+03	9.11E+02
	t(s)	1.31	1.53	1.88	1.36	2.05	2.13
FPA	Mean	**-1.0809**	**-1.5000**	-7.37E-01	**0**	6.92E+02	6.92E+02
	Std.	0	1.90E-08	1.34E-01	0	0	0
	t(s)	3.30	3.50	3.95	3.36	4.02	4.25
TLBO	Mean	**-1.0809**	-1.407	-5.37E-01	**0**	1.03E+01	2.71E+01
	Std.	0	2.12E-01	2.22E-01	0	1.75E+01	7.68E+01
	t(s)	5.16	5.36	6.17	5.22	6.53	6.59
WCA	Mean	**-1.0809**	-1.66E+00	-1.36E-01	7.00E-11	2.99E+01	2.20E+01
	Std.	0	5.04E-01	4.25E-01	1.24E-10	2.56E+01	2.45E+01
	t(s)	5.20	5.41	6.23	5.27	6.59	6.65
SOS	Mean	**-1.0809**	-1.3313	-6.40E-01	**0**	**0**	**0**
	Std.	0	1.95E-01	3.50E-01	0	0	0
	t(s)	0.11	0.11	0.09	0.59	1.53	1.16
EFO	Mean	**-1.0809**	-1.3695	-5.93E-01	3.40E-10	2.12E+02	2.04E+02
	Std.	6.33E-04	2.51E-01	2.84E-01	9.32E-09	3.84E+02	3.51E+02
	t(s)	0.05	0.08	0.06	0.06	0.13	0.14
GSA	Mean	-1.0780	-0.8163	-2.14E-01	2.44E-01	2.91E+02	2.91E+02
	Std.	2.43E-03	9.55E-02	8.33E-02	2.57E-01	2.84E+02	2.86E+02
	t(s)	9.59	11.17	14.06	9.56	11.70	14.39
FBI	Mean	**-1.0809**	**-1.5000**	**-1.1309**	**0**	**0**	**0**
	Std.	0	0	3.27E-01	0	0	0
	t(s)	0.81	0.92	1.23	0.63	1.22	1.20

Std. = Standard deviation; t(s) = computational time (unit: second); bold numbers represent the best values.

Table 7
Results of solving 50 benchmark problems.

	GA	PSO	DE	ABC	FA	WOA	FPA	TLBO	WCA	SOS	EFO	GSA	FBI
(1)	15 30%	24 48%	32 64%	37 74%	23 46%	27 54%	28 56%	32 64%	20 40%	39 78%	19 38%	23 46%	**48** **96%**
(2)	15 30%	24 48%	32 64%	37 74%	23 46%	27 54%	28 56%	31 62%	20 40%	39 78%	19 38%	23 46%	**45** **90%**
(3)	170	156	151	192	715	102	202	326	329	97	58	729	**65**

(1): Count of best performance of the algorithm and the value in percentage.
(2): Count of the algorithm finding the global minimum and the value in percentage.
(3): Total computational time for running 50 benchmark functions (unit: second).

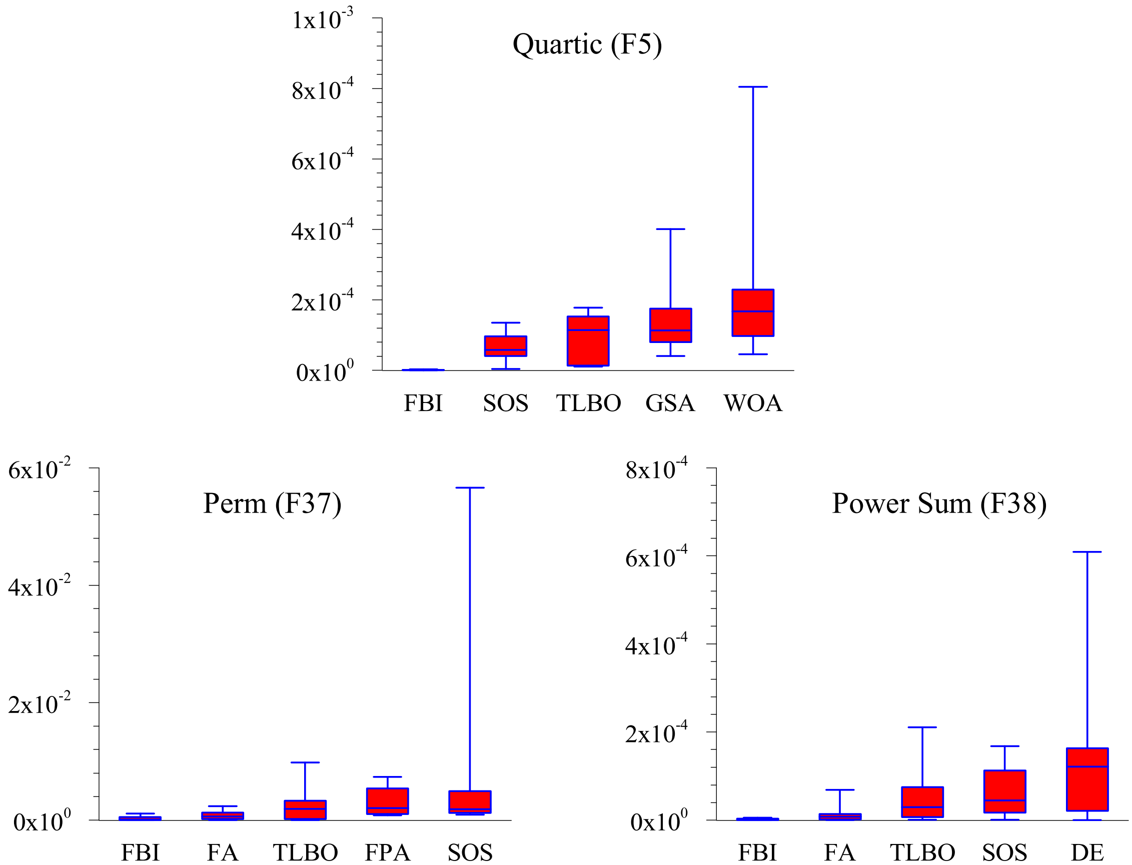

Fig. 8. Box plot analysis of FBI and other top-performing algorithms for relatively difficult-to-optimize benchmark functions.

Table 8
Result comparison for construction case study.

Algorithm	Success rate (%)	Optimal value				Ranking
		Best	Worst	Mean	Std.	
GA	36.67	117	120	118.35	1.18	–
DE	46.67	117	120	117.75	0.91	–
PSO	26.67	117	120	118.25	0.97	–
ABC	56.67	117	119	117.70	0.86	4
FA	10.00	117	120	119.04	1.10	-
WOA	16.67	117	122	118.87	1.28	–
FPA	36.67	117	118	117.63	0.49	3
TLBO	50.00	117	118	117.50	0.50	2
WCA	13.33	117	120	118.57	1.10	–
SOS	100	117	117	117.00	0.00	1
EFO	6.67	117	122	119.43	1.22	–
GSA	6.67	117	123	119.57	1.30	–
FBI	100	117	117	117.00	0.00	1

–: Not ranked in top five models.

A1 and A2 are less effective for solving nonseparable problems. This finding is reasonable because A1 and A2 perform exploration, finding the suspect by searching all of the search space.

(4) Both FBI_3 and FBI_4 performed deficiently in some MS and MN functions: They are worse than FBI_1 and FBI_2 for solving a Schwefel function; they perform ineffectively in Ackley, Penalized, and Penalized2. Therefore, B1 and B2 are less effective for solving multimodal problems, because they only execute an exploitation search around the most probable suspected location (X_{best}).

5. Discussion

5.1. Novelty of FBI algorithm

(1) The structure of FBI algorithm includes two teams that balance exploration and exploitation. The investigation team (steps A1 and A2) performs the exploration role (it seeks the suspect by searching the whole search space), whereas the pursuit team (steps B1 and B2) executes exploitation searches around the suspected locations. This effective balance helps FBI remarkably outperform other metaheuristic algorithms for solving numerical optimization problems.

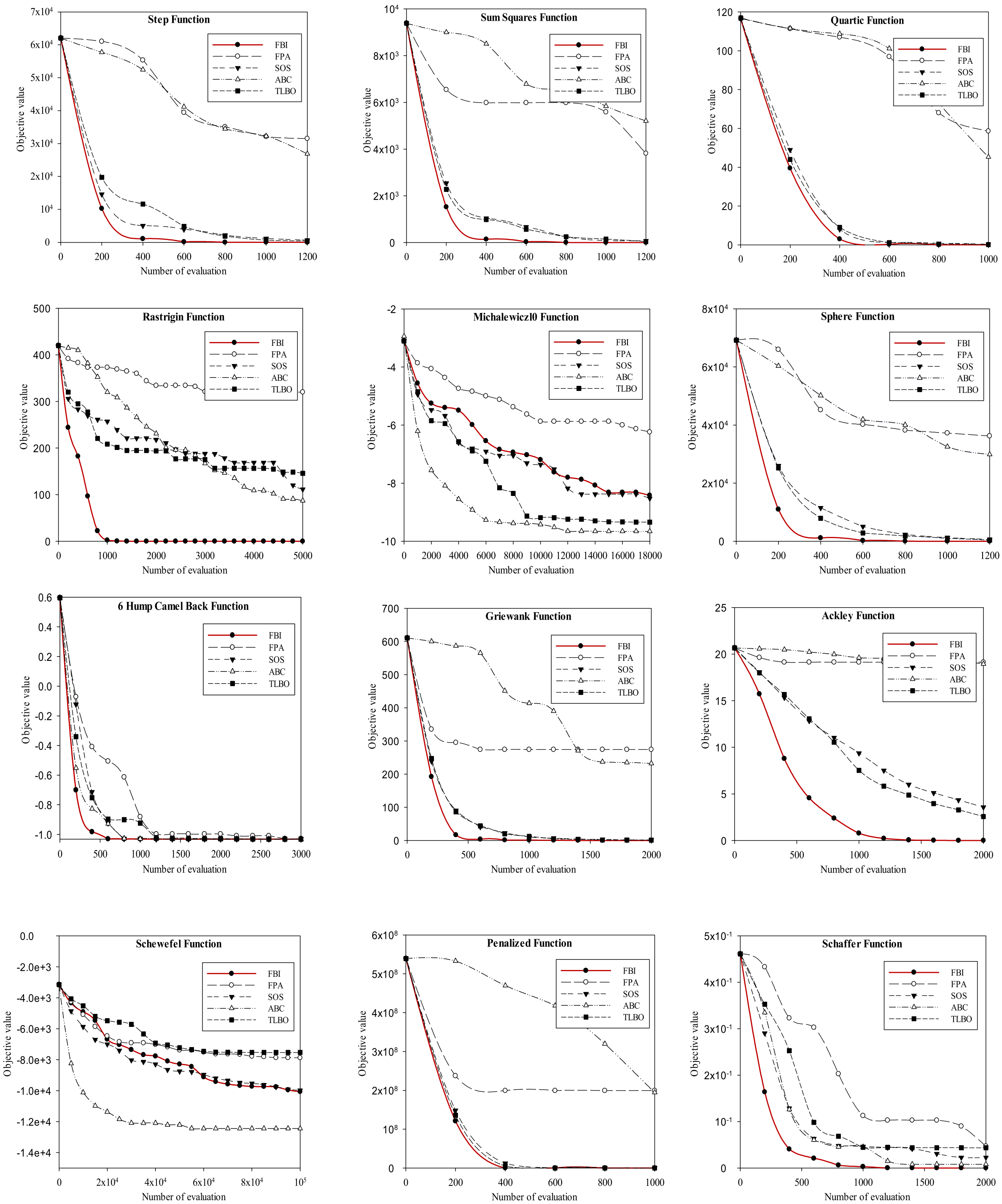

Fig. 9. Convergence comparisons.

(2) All steps in FBI are closely analogous to those of a large-scale forensic investigation by the police, as presented by Salet (2017) [37]. Step A1 is the "interpretation of findings": The team assesses the information and initially identifies possible suspect locations. Each possible location of the suspect is investigated. Step A2 is the "direction of inquiry": any location with a low probability is likely to be excluded. Step B1 is the "actions" process, in which all police agents in the pursuit team go to the designated location. Step B2 extends the "actions" process, as all police agents closely follow commands from headquarters, and in coordination with other team members, arrest the suspect.

(3) All components in FBI (A1, A2, B1, and B2) are effective optimizers, even for solving high-dimensional problems ($D = 1000$). However, A1 and A2 are less effective for solving nonseparable problems, whereas B1 and B2 are less effective for solving multimodal problems.

(4) The FBI algorithm does not require specific parameters (such as crossover rate and mutation rate in GA; or cognitive component, social component, and initial weight in PSO). It requires only a population size and a stopping criterion, which are found in all metaheuristic algorithms, to achieve the greatest performance. This feature favors the

Table 9
Results of solving CEC2017 benchmark functions ($D = 10$).

Func. No.	EBO with CMAR		jSO		LSHADE		FBI	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.64E+01	1.48E+01
2	This function is excluded in [60] because it shows unstable behavior and significant performance variations for the same algorithm implemented in MATLAB.							
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.10E−04	3.04E−04
5	0.00E+00	0.00E+00	1.76E+00	7.53E−01	2.58E+00	9.48E−01	2.17E+00	7.34E−01
6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	1.06E+01	1.75E−01	1.18E+01	6.01E−01	1.21E+01	6.79E−01	1.20E+01	4.82E−01
8	0.00E+00	0.00E+00	1.95E+00	7.36E−01	2.38E+00	8.60E−01	2.23E+00	7.57E−01
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10	3.72E+01	5.39E+01	3.59E+01	5.49E+01	3.86E+01	5.00E+01	1.05E+01	4.21E+00
11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.04E−02	1.02E−01	0.00E+00	0.00E+00
12	9.02E+01	7.44E+01	2.66E+00	1.66E+01	3.33E+01	5.37E+01	3.02E+02	6.99E+02
13	2.17E+00	2.53E+00	2.96E+00	2.33E+00	3.88E+00	2.33E+00	5.70E+00	4.49E+00
14	6.05E−02	2.36E−01	5.85E−02	2.34E−01	7.73E−01	9.19E−01	1.70E+00	2.14E+00
15	1.09E−01	1.74E−01	2.21E−01	1.98E−01	1.50E−01	1.88E−01	1.01E+00	1.12E+00
16	4.17E−01	1.98E−01	5.69E−01	2.62E−01	3.82E−01	1.77E−01	1.12E−01	3.36E−02
17	1.47E−01	2.03E−01	5.02E−01	3.45E−01	9.75E−02	1.28E−01	3.29E−02	2.01E−02
18	7.00E−01	2.77E+00	3.08E−01	1.93E−01	2.15E−01	1.97E−01	5.50E+00	2.75E+00
19	1.50E−02	1.88E−02	1.07E−02	1.24E−02	7.26E−03	9.69E−03	4.11E−01	4.03E−01
20	1.47E−01	1.57E−01	3.43E−01	1.28E−01	1.22E−02	6.06E−02	1.08E−01	1.99E−01
21	1.14E+02	3.52E+01	1.32E+02	4.79E+01	1.49E+02	5.15E+01	9.83E+01	1.33E+01
22	9.85E+01	1.10E+01	1.00E+02	0.00E+00	9.81E+01	1.39E+01	1.00E+02	2.76E−01
23	3.00E+02	7.07E−01	3.01E+02	1.57E+00	3.03E+02	1.42E+00	3.01E+02	1.26E+00
24	1.66E+02	9.97E+01	2.97E+02	7.85E+01	2.99E+02	7.95E+01	1.03E+02	3.97E+01
25	4.12E+02	2.12E+01	4.06E+02	1.73E+01	4.10E+02	1.98E+01	4.03E+02	1.30E+01
26	2.65E+02	4.74E+01	3.00E+02	0.00E+00	3.00E+02	0.00E+00	2.92E+02	4.28E+01
27	3.92E+02	2.40E+00	3.89E+02	2.23E−01	3.89E+02	1.76E−01	3.90E+02	0.00E+00
28	3.07E+02	7.18E+01	3.39E+02	9.56E+01	3.23E+02	8.02E+01	3.00E+02	0.00E+00
29	2.31E+02	3.77E+00	2.34E+02	2.93E+00	2.35E+02	2.42E+00	2.36E+02	2.29E+00
30	4.07E+02	1.78E+01	3.95E+02	4.45E−02	4.06E+02	2.08E+01	4.31E+02	1.04E+01

Table 10
Hypothesis test results.

No.	Function	**FBI**		EBO with CMAR	**FBI**		jSO	**FBI**		LSHADE
1	S&R Bent Cigar Function			B			B			B
2	S&R Sum of Different Power Function				This function has been excluded by [60]					
3	S&R Zakharov Function		ND			ND			ND	
4	S&R Rosenbrock's Function			B			B			B
5	S&R Rastrigin's Function			B			B	**B**		
6	S&R Expanded Scaffer's F6 Function		ND			ND			ND	
7	S&R Lunacek Bi_Rastrigin Function			B			B		ND	
8	S&R NonContinuous Rastrigin's Function			B			B	**B**		
9	S&R Levy Function		ND			ND			ND	
10	S&R Schwefel's Function	**B**			**B**			**B**		
11	Hybrid Function 1		ND			ND		**B**		
12	Hybrid Function 2			B			B			B
13	Hybrid Function 3			B			B			B
14	Hybrid Function 4			B			B			B
15	Hybrid Function 5			B			B			B
16	Hybrid Function 6	**B**			**B**			**B**		
17	Hybrid Function 6			B	**B**					B
18	Hybrid Function 6			B			B			B
19	Hybrid Function 6			B			B			B
20	Hybrid Function 6	**B**			**B**					B
21	Composition Function 1	**B**			**B**			**B**		
22	Composition Function 2			B		ND				B
23	Composition Function 3		ND			ND		**B**		
24	Composition Function 4	**B**			**B**			**B**		
25	Composition Function 5	**B**			**B**			**B**		
26	Composition Function 6			B	**B**			**B**		
27	Composition Function 7		ND			ND			ND	
28	Composition Function 8		ND				B	**B**		
29	Composition Function 9		ND			ND			ND	
30	Composition Function 10			B			B			B
Count of better performance		**6**		**15**	**8**		**13**	**11**		**12**

S&R: Shifted and rotated; B: Better; ND: No difference.

Table 11
FBI and component analysis for solving high-dimensional problems ($D = 1000$).

Function	C	Range	Min (D = 30)	Index	FBI D = 30	FBI D = 1000	FBI_1 D = 1000	FBI_2 D = 1000	FBI_3 D = 1000	FBI_4 D = 1000
Step	US	[−100, 100]	0	Mean	0	0	0	0	0	0
				Std.	0	0	0	0	0	0
Sphere	US	[−100, 100]	0	Mean	0	0	0	0	0	0
				Std.	0	0	0	0	0	0
Sum Squares	US	[−10, 10]	0	Mean	0	0	0	0	0	0
				Std.	0	0	0	0	0	0
Quartic	US	[−1.28, 1.28]	0	Mean	9.29E−07	1.03E−06	2.81E−06	2.67E−06	4.88E−06	1.20E−06
				Std.	4.88E−07	5.92E−07	1.73E−06	2.26E−06	4.18E−06	5.07E−07
Schwefel 2.22	UN	[−10, 10]	0	Mean	0	0	0	0	0	0
				Std.	0	0	0	0	0	0
Schwefel 1.2	UN	[−100, 100]	0	Mean	0	0	0	0	0	0
				Std.	0	0	0	0	0	0
Rosenbrock	UN	[−30, 30]	0	Mean	1.79E+01	9.83E+02	9.99E+02	9.97E+02	9.90E+02	9.90E+02
				Std.	1.82E−01	5.06E+00	2.70E−01	3.74E−01	1.81E−01	3.34E−02
Dixon–Price	UN	[−10, 10]	0	Mean	6.67E−01	6.67E−01	6.67E−01	9.97E−01	6.67E−01	6.67E−01
				Std.	0	0	0	3.97E−03	0	0
Rastrigin	MS	[−5.12, 5.12]	0	Mean	0	0	0	0	0	0
				Std.	0	0	0	0	0	0
Schwefel	MS	[−500, 500]	−12 569.49	Mean	−12 569.49	−10 291.36	−6988.52	−7000.30	−6600.68	−6364.79
				Std.	3.12E−07	1.66E+02	2.81E+02	5.02E+02	2.98E+02	3.05E+02
Griewank	MN	[−600, 600]	0	Mean	0	0	0	0	0	0
				Std.	0	0	0	0	0	0
Ackley	MN	[−32, 32]	0	Mean	0	−3E+14	−7.5E+13	−7.5E+13	−3E+14	−1.3E+05
				Std.	0	0	1.29E+14	1.3E+14	1.34E+10	1.72E+04
Penalized	MN	[−50, 50]	0	Mean	0	0	0	4.09E−11	3.10E−11	4.40E−01
				Std.	0	0	0	1.13E−11	0	4.84E−02
Penalized2	MN	[−50, 50]	0	Mean	0	0	0	4.77E−10	2.38E−07	2.36E+00
				Std.	0	0	0	2.09E−10	2.28E−07	6.67E−02

Note: C: Characteristic, D: Dimension, U: Unimodal, M: Multimodal, S: Separable, N: Nonseparable, FBI_1: FBI without A1 step, FBI_2: FBI without A2 step, FBI_3: FBI without A3 step, FBI_4: FBI without A4 step.

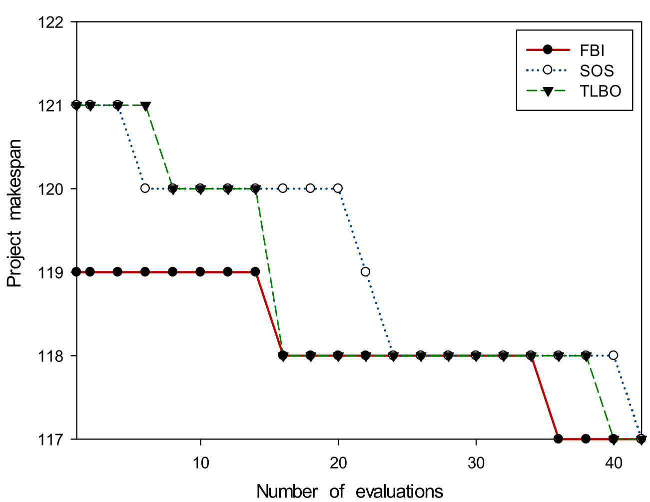

Fig. 10. Best duration curves for project.

stability of the performance of FBI and enables it to be easily applied to new optimization problems.

5.2. Performance evaluation

As presented in the previous sections, the experiments of this study posed many challenges to validate the performance of FBI through a variety of problems. The first experiment supported the claim that FBI dominated 12 other algorithms for solving 50 well-known benchmark problems (which featured unimodal, multimodal, separable, and nonseparable cases). FBI can optimize 48 out of 50 functions (96%), whereas SOS, the second-best model, can optimize 39 functions (68%); this fact indicates the remarkable supremacy of FBI over the other algorithms. FBI's fast convergence enables it to determine the optimal solution sooner than other methods; thus, FBI avoids the danger of fixating on some premature optima prior to reaching stopping criteria. In addition, FBI is an effective algorithm that completed the optimization process for 50 benchmark functions in 65 seconds. Unquestionably, FBI dominated the competition for solving unimodal, multimodal, separable, and nonseparable problems in terms of efficiency and computational time.

The second experiment verifies the performance of FBI in practical applications. FBI was employed to discover the best resource scheduling for a construction project. The result proved that FBI produced the optimal resource scheduling with a success rate of 100% within a notably small number of evaluations. The facts affirm that FBI can avoid the problem of stopping its optimization process before approaching the optimal solution; thus, FBI has a high chance of minimizing the schedule time of a project. According to the analyzed results, FBI is the most efficient and robust optimizer.

In the third experiment, FBI obtained a competitive performance with CEC winners in solving 30 composition benchmark functions of CEC-2017, including jSO, EBO with CMAR, and LSHADE. The statistical results and hypothesis testing support the claim that FBI is a significant contributor for solving extremely sophisticated optimization problems and can enable users to exercise more options when addressing complex engineering-related challenges. Notably, FBI is a newly developed optimization algorithm and not similar to jSO, EBO, and LSHADE, which are integrated combinations of multiple techniques.

The fourth experiment confirmed the performance of FBI on extremely high-dimensional problems by increasing the number of dimensions of 14 well-known benchmark functions, described in Section 3.1, from $D = 30$ to $D = 1000$. Interestingly, FBI exhibited excellent performance in solving those problems. The effectiveness of the four components of the FBI for solving high-dimensional problems was analyzed; the results revealed that although each component has advantages and disadvantages for solving various types of problems, all components in FBI are

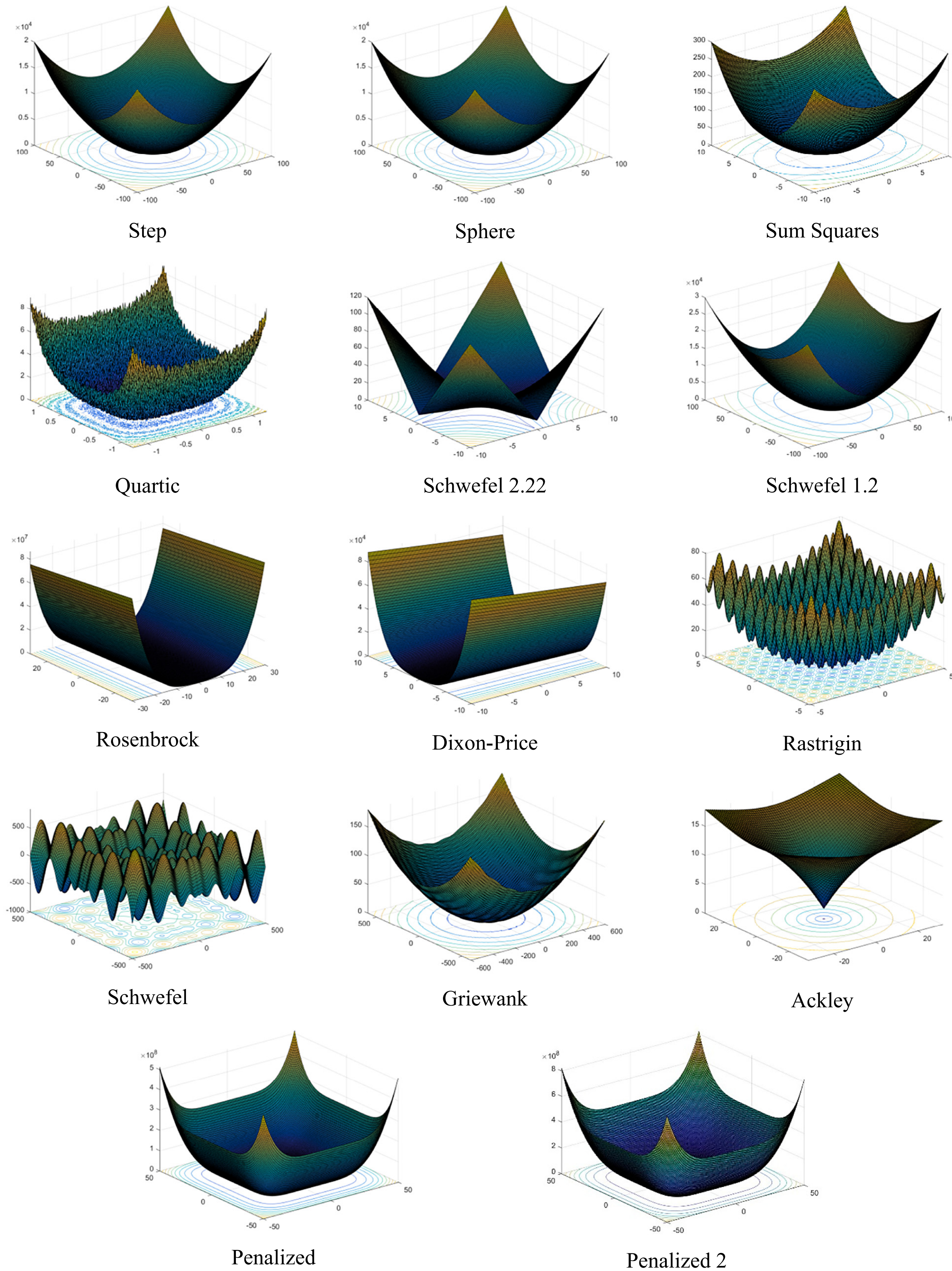

Fig. 11. Perspective views ($D = 2$) of 14 benchmark functions.

efficient; thus, when one is inactive, the others support effective discovery.

6. Conclusion

Inspired by the forensic investigation process of police officers, this work developed a novel metaheuristic algorithm, FBI. Aside from population size and stopping criterion, (which are common to all metaheuristic algorithms), FBI does not depend on preset parameters to achieve its greatest performance. This feature enhances the performance stability of FBI and enables it to be easily applied to new optimization problems.

All steps in FBI are closely analogous to those of a large-scale forensic investigation by police. The structure of FBI algorithm is composed of two teams that aim to properly balance exploration and exploitation in search processes. The investigation team performs the exploration role and seeks the suspect by searching the whole search space. The pursuit team executes exploitation searches around the most probable locations. This mutual balance helps FBI remarkably outperform other metaheuristic algorithms for solving numerical optimization problems.

The performance of FBI was validated in four experiments (all of which demonstrated its robustness, efficiency, stability, and user-friendliness). The analytical results indicated that FBI was superior to 12 well-known and contemporary metaheuristic algorithms in all experiments and highly competitive with IEEE CEC winners. The efficiency of FBI is revealed by its high-performance levels for solving a variety of problems (even high-dimensional problems), its relatively short computational time for solving problems and its power of reaching optimal solutions more quickly than other algorithms. We thus conclude that the novel FBI algorithm is a powerful optimization tool.

A metaheuristic optimization platform was developed to make the comparison much easier and faster and to display performance indicators clearly, logically, and graphically. Some future work could investigate the efficient combination of FBI and machine learning techniques. FBI can be upgraded to become an entirely parameter-free algorithm, without even population size or termination criteria, which would be able to solve multiobjective optimization problems. The metaheuristic platform can continue to be developed and updated with additional optimization algorithms.

CRediT authorship contribution statement

Jui-Sheng Chou: Conceptualization, Methodology, Validation, Investigation, Resources, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Ngoc-Mai Nguyen:** Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization.

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.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology, Taiwan, for financially supporting this research under contract MOST 107-2221-E-011-035-MY3.

Source code and supplementary data

Source code and supplementary data are available at https://www.researchgate.net/profile/Jui-Sheng_Chou.

References

[1] A.H. Gandomi, A.H. Alavi, Krill herd: A new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (2012) 4831–4845.
[2] G.-G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Memet. Comput. 10 (2018) 151–164.
[3] K. Hussain, M.N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. (2018) 1–43.
[4] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A.H. Gandomi, A novel hybridization strategy for krill herd algorithm applied to clustering techniques, Appl. Soft Comput. 60 (2017) 423–435.
[5] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[6] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.
[7] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (2006) 28–39.
[8] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[9] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspir. Comput. 2 (2010) 78–84.
[10] J.J.Q. Yu, V.O.K. Li, A social spider algorithm for global optimization, Appl. Soft Comput. 30 (2015) 614–627.
[11] E. Ali, S. Abd-Elazim, Bacteria foraging optimization algorithm based load frequency controller for interconnected power system, Int. J. Electr. Power Energy Syst. 33 (2011) 633–638.
[12] M. Gen, R. Cheng, Genetic Algorithms and Engineering Optimization, John Wiley & Sons, 2000.
[13] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[14] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: J.R. González, D.A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 65–74.
[15] W. Shen, X. Guo, C. Wu, D. Wu, Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm, Knowl.-Based Syst. 24 (2011) 378–385.
[16] R.-q. Zhao, W.-s. Tang, Monkey algorithm for global numerical optimization, J. Uncertain Syst. 2 (2008) 165–176.
[17] J. Kennedy, Particle swarm optimization, in: Encyclopedia of Machine Learning, Springer, 2011, pp. 760–766.
[18] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceeings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009, India), IEEE, 2009, pp. 210–214.
[19] B.K. Sahu, S. Pati, P.K. Mohanty, S. Panda, Teaching–learning based optimization algorithm based fuzzy-PID controller for automatic generation control of multi-area power system, Appl. Soft Comput. 27 (2015) 240–249.
[20] X.-S. Yang, Flower pollination algorithm for global optimization, in: International Conference on Unconventional Computing and Natural Computation, Springer, 2012, pp. 240–249.
[21] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166.
[22] S.R. Rayapudi, An intelligent water drop algorithm for solving economic load dispatch problem, Int. J. Electr. Electron. Eng. 5 (2011) 43–49.
[23] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[24] H. Abedinpourshotorban, S. Mariyam Shamsuddin, Z. Beheshti, D.N.A. Jawawi, Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm, Swarm Evol. Comput. 26 (2016) 8–22.
[25] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Inform. Sci. 179 (2009) 2232–2248.
[26] D.-H. Tran, M.-Y. Cheng, M.-T. Cao, Solving resource-constrained project scheduling problems using hybrid artificial bee colony with differential evolution, J. Comput. Civ. Eng. 30 (2016) 04015065.
[27] K. Sörensen, Metaheuristics—the metaphor exposed, Int. Trans. Oper. Res. 22 (2015) 3–18.
[28] A.F. Nematollahi, A. Rahiminejad, B. Vahidi, A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization, Appl. Soft Comput. 59 (2017) 596–621.
[29] M.D. Li, H. Zhao, X.W. Weng, T. Han, A novel nature-inspired algorithm for optimization: Virus colony search, Adv. Eng. Softw. 92 (2016) 65–88.
[30] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.
[31] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.

[32] H.M. Zawbaa, E. Emary, C. Grosan, V. Snasel, Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach, Swarm Evol. Comput. 42 (2018) 29–42.
[33] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[34] S. Christodoulou, Scheduling resource-constrained projects with ant colony optimization artificial agents, J. Comput. Civ. Eng. 24 (2009) 45–55.
[35] P. Lamas, E. Demeulemeester, A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations, J. Sched. 19 (2016) 409–428.
[36] College of Policing, Investigation process, 2013, https://www.app.college.police.uk/app-content/investigations/investigation-process/.
[37] R. Salet, Framing in criminal investigation: How police officers (re) construct a crime, Police J. 90 (2017) 128–142.
[38] Horse Sanctuary UK Limited, S.W.A.T., 2018, http://www.bushywood.com/SWAT.htm.
[39] T. Droog, International Campaign to Free Jose Maria Sison Launched, 2007, http://www.statewatch.org/news/2007/sep/sison-campaign.pdf.
[40] VnExpress, Drug lords killed in massive raid in northern Vietnam, 2018, https://e.vnexpress.net/news/news/drug-lords-killed-in-massive-raid-in-northern-vietnam-3770389.html.
[41] R. Gehl, D. Plecas, Introduction to Criminal Investigation: Processes, Practices and Thinking, Justice Institute of British Columbia, New Westminster, BC, 2016.
[42] L. Huang, Y. Ding, M. Zhou, Y. Jin, K. Hao, Multiple-solution optimization strategy for multirobot task allocation, IEEE Trans. Syst. Man Cybern.: Syst. (2018) 1–12.
[43] A.Y. Lam, V.O. Li, Chemical-reaction-inspired metaheuristic for optimization, IEEE Trans. Evol. Comput. 14 (2010) 381–399.
[44] M. Cheng, D. Tran, Two-phase differential evolution for the multiobjective optimization of time–cost tradeoffs in resource-constrained construction projects, IEEE Trans. Eng. Manage. 61 (2014) 450–461.
[45] R.H. Möhring, A.S. Schulz, F. Stork, M. Uetz, Solving project scheduling problems by minimum cut computations, Manage. Sci. 49 (2003) 330–350.
[46] K. Sears, G. Sears, R. Clough, Construction Project Management: A Practical Guide to Field Construction Management, fifth ed., John Wiley & Sons, 2008.
[47] N.H. Awad, M.Z. Ali, J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017.
[48] J. Brest, M.S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jSO, in: 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 1311–1318.
[49] A. Kumar, R.K. Misra, D. Singh, Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase, in: 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 1835–1842.
[50] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1658–1665.
[51] A.P. Engelbrecht, Fitness function evaluations: A fair stopping condition? in: 2014 IEEE Symposium on Swarm Intelligence, IEEE, 2014, pp. 1–8.
[52] J.-S. Chou, N.-T. Ngo, Modified firefly algorithm for multidimensional optimization in structural design problems, Struct. Multidiscip. Optim. 55 (2017) 2013–2028.
[53] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (2011) 303–315.
[54] M.S. Kıran, O. Fındık, A directed artificial bee colony algorithm, Appl. Soft Comput. 26 (2015) 454–462.
[55] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[56] J. Yao, N. Kharma, P. Grogono, Bi-objective multipopulation genetic algorithm for multimodal function optimization, IEEE Trans. Evol. Comput. 14 (2010) 80–102.
[57] A. Saxena, R. Kumar, S. Das, β-Chaotic map enabled grey wolf optimizer, Appl. Soft Comput. 75 (2019) 84–105.
[58] S.K. Dinkar, K. Deep, Opposition based Laplacian ant lion optimizer, J. Comput. Sci. 23 (2017) 71–90.
[59] K. Hussain, M.N.M. Salleh, S. Cheng, R. Naseem, Common benchmark functions for metaheuristic evaluation: A review, JOIV: Int. J. Inform. Vis. 1 (2017) 218–223.
[60] N.H. Awad, M.Z. Ali, P.N. Suganthan, J.J. Liang, B.Y. Qu, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Technical Report, Nanyang Technological University, Jordan University of Science and Technology, Jordan, and Zhengzhou University, Singapore, Jordan, Zhengzhou China, 2016.