



Optical microscope algorithm: A new metaheuristic inspired by microscope magnification for solving engineering optimization problems

Min-Yuan Cheng^{*}, Moh Nur Sholeh

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, No. 43, Sec. 4, Keelung Rd., Taipei, 106, Taiwan, ROC



ARTICLE INFO

Article history:

Received 18 May 2023

Received in revised form 31 July 2023

Accepted 25 August 2023

Available online 3 September 2023

Keywords:

Engineering problem

Magnification

Metaheuristic

Optical microscope algorithm

Optimization

ABSTRACT

A novel optimization algorithm called the optical microscope algorithm (OMA) is developed and applied in this study. Drawing inspiration from the magnification capabilities of an optical microscope on the target object, OMA uses the naked eye for initial observation and simulates the magnification process through an objective lens and an eyepiece. The performance of OMA, which is user friendly and does not require predefined operating parameters, is validated through two experiments: (1) OMA is compared to nine well-known metaheuristic algorithms using constraint handling with 50 benchmark functions involving multiple dimensions. The results indicate that OMA consistently outperforms all other algorithms and requires a short computational time. (2) OMA is applied to solve engineering problems, including structural optimization and multiple resources leveling in the multiple projects scheduling (MRLMP). In those cases, OMA not only demonstrates superiority but also requires the fewest evaluations of objective functions. The novel OMA, which is robust, easy to implement, and uses fewer control parameters, can be deployed to solve for various numerical optimization problems. The source code of OMA is publicly accessible at <https://www.mathworks.com/matlabcentral/fileexchange/134541-optical-microscope-algorithm-oma>.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

Engineering optimization is a challenging field that encourages researchers to refine further and optimize current engineering designs. Metaheuristics have become increasingly popular as strategies for solving complex problems in engineering, the social sciences, economics, and politics [1] due to their conceptual simplicity, ease of implementation, and robust results [2,3]. The application of metaheuristics is computer-dependent, advances in computer processing power have accelerated the development of metaheuristics [4].

Research in metaheuristics worldwide has produced optimization methods that have demonstrated their superiority through various characteristics. Agrawal et al. [5] used the characteristics and behavior of metaheuristic algorithms to distinguish them into four categories, including swarm-intelligence-based, evolution-based, physics-based, and human-based. Swarm-intelligence-based algorithms draw inspiration from the social behavior of

insects, animals, fish, and birds, such as Particle Swarm Optimization (PSO) [6], Ant Colony Optimization (ACO) [7], Artificial Bee Colony (ABC) [8], and Fish Swarm Algorithm (FSA) [9]. Evolution-based algorithms draw inspiration from the genetic evolution process, which operates based on the best criteria for survival. Examples of evolution-based algorithms include the Genetic Algorithm (GA) [10], Differential Evolution (DE) [11], Evolutionary Programming (EP) [12], and Evolution Strategy (ES) [13]. Physics-based algorithms are inspired by the rules of physics and include algorithms, such as Electromagnetic Field Optimization (EFO) [14], Gravitational Search Algorithm (GSA) [15], Simulated Annealing (SA) [16], and Black Hole Algorithm (BHA) [17]. Algorithms based on human behaviors include Tabu Search (TS) [18], Teaching-Learning-Based Optimization (TLBO) [19], League Championship Algorithm (LCA) [20], and Forensic-Based Investigation (FBI) [21]. Beyond these four, there is another category of algorithms called nature-inspired algorithms, which are inspired by the behavior of individual animals in the wild. Examples include the Firefly Algorithm (FA) [22], Symbiotic Organism Search (SOS) [23], Bad Eagle Search (BES) [24], and Jellyfish Search (JS) [25].

* Corresponding author.

E-mail addresses: myc@mail.ntust.edu.tw (M.-Y. Cheng), mns@live.undip.ac.id (M.N. Sholeh).

Diversification (exploration) and intensification (exploitation) are critical phases of metaheuristic algorithms. The main challenge for all metaheuristics is to balance these two phases to obtain near-optimal solutions efficiently [26,27]. Some researchers have attempted various methods to achieve this balance, such as using control parameters. Parameter tuning is one of the key factors that influence the efficiency of any metaheuristic algorithm [26]. An algorithm with correctly set parameters will converge to global positions more quickly, further increasing the efficiency of an algorithm [27].

Recent advancements in optimization have witnessed the introduction of innovative approaches to address specific challenges. These include whale optimization algorithm (WOA) applied to multi-objective scenarios, such as cost and emissions scheduling of thermal plants in energy hubs [28], the crow search optimization algorithm (CSOA) applied to multi-area economic emission dispatch in large-scale power plants considering grid tie-line limitations [29], and dynamic optimal scheduling strategies using efficient black widow optimization (EBWO) for multi-charging scenarios of plug-in-electric vehicles in a smart grid environment [30]. These studies highlight the ongoing progress and the growing need for optimization techniques tailored to specific constraints and objectives.

The other research field successfully combines metaheuristics and machine learning approaches and proved to be able to obtain outstanding results in different areas. The chaotic firefly algorithm with enhanced exploration (CFAEE) proves its effectiveness in tackling global optimization problems [31]. The GA-based hierarchical feature selection (GA-based HFS) approach showcases its superiority in handwritten word recognition systems, enhancing recognition accuracy [32]. The quasi-reflection-based learning artificial bee colony (ABCQRBEST) algorithm optimizes hidden units and weight connections in artificial neural networks, improving learning and prediction capabilities [33]. These results emphasize the immense potential of integrating metaheuristics and machine learning techniques in various domains.

Motivated by these developments and the demand for further advancements, this paper introduces a new simple and powerful metaheuristic algorithm called Optical Microscope Algorithm (OMA). Inspired by the iterative zoom-in process of an optical microscope, OMA is designed to obtain targets by utilizing two distinct phases: an objective lens for exploration and an eyepiece for exploitation.

The primary objective of this study is to develop an optimization algorithm that surpasses existing methods in terms of performance, ease of implementation, and control parameter utilization. OMA is introduced as a novel approach to pursue this goal, bringing forth several notable contributions to the field.

The remainder of the paper is organized as follows: Section 2 describes the magnification concept and presents the algorithm in detail; Section 3 compares the performance of OMA with other well-known algorithms on mathematical benchmark functions and engineering design-problem case studies; Sections 4 and 5 provide the discussion and conclusions, respectively.

2. Optical microscope algorithm

The proposed OMA was designed to simulate the repetitive zoom-in or magnification of objects, a process generally performed by a microscope. The authors were inspired by the transformation of vision that occurs first at the naked eye level and then through the lenses of a microscope to develop a method for finding target solutions in the magnification process. Both the microscope magnification process and the developed algorithm are described in detail in the following subsections.

2.1. Microscope magnification

Optical (aka light) microscopes are a type of microscope that uses visible light and a system of lenses to obtain magnified images of small objects [34,35]. The simplest compound microscope is constructed with two convex lenses and is observed using the observer's unaided (naked) eyes to locate a target object.

After identifying the objects with the naked eye, the magnification process is carried out using objective lenses and an eyepiece. Objective lenses are placed close to the target object and are usually arranged in sets of three, providing different magnification levels [36]. The eyepiece, also referred to as the ocular, consists of several lenses set inside a cylindrical barrel [37]. The best magnification and focus are achieved by adjusting both the objective lens and the eyepiece.

2.2. Optical microscope algorithm (OMA)

Various physics-based algorithms have been developed, including EFO, which simulates the behavior of electromagnets with different polarities and leverages a nature-inspired ratio [14]; GSA, which is based on the mathematical modeling of Newton's laws of gravity and motion [15]; SA, which relies on probability and statistical mechanics [16]; and Harmony Search (HS), which imitates the improvisation process used by musicians to find pleasing harmonies [38].

OMA is a physics-based algorithm that simulates the process an observer uses to zoom-in on objects, starting from the observer's eyes and then through microscope lenses. The four-step process that OMA uses to obtain the best target object is shown in Fig. 1. Steps (2) and (3) involve an iterative process.

- (1) **Naked eye:** Observation begins by properly mounting the object on the specimen stage. Next, the observer makes naked-eye observations of the object to obtain information regarding its size, shape, and characteristics.
- (2) **Objective lens:** Observation of the initial target objects continues using the objective lens. The magnifying power (MP) is the factor used to determine image magnification in this step and is a fixed variable, set by the specifications of the particular objective optical element used.
- (3) **Eyepiece:** The eyepiece further magnifies the objective lens and its magnifying power (MP). The total magnification of a particular microscope is calculated by multiplying the magnification values of the objective lens and the eyepiece.
- (4) **Best target object:** The results of each magnification cycle are compared to obtain the magnification value of target object.

To simulate this magnification concept, the process used to zoom-in on a leaf is presented in Fig. 2, which illustrates the staged magnification of the image to the cellular level. The cells become the target objects of the microscope in the iterative zoom-in process. Since the cells are microscopic, they are invisible to the naked eye [39].

The two phases of OMA are the objective lens and the eyepiece. This algorithm assumes the term "magnification" to be the target object to be magnified. In the magnification process, the i th potential target object to be magnified is denoted as M_i and referred to as the solution; $i = 1, 2, \dots, NP$, with NP equal to the total number of potential target objects (population size). Within the optimization process, the best outcome achieved by OMA is termed the best target object (M_{best}), which serves as the best solution. OMA is an iterative process that ends when the number of current iterations (num_iter) reaches the present maximum number of iterations (max_iter).

Optical Microscope Algorithm (OMA)

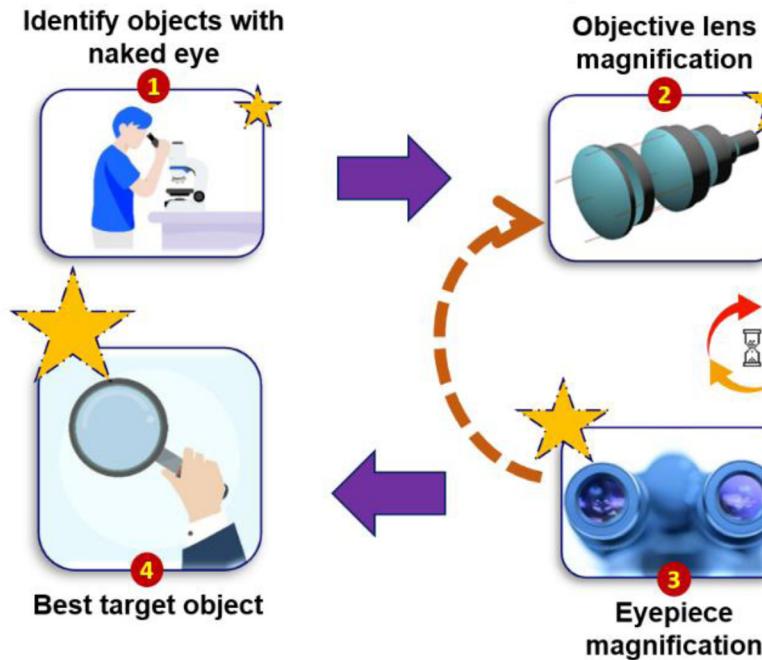
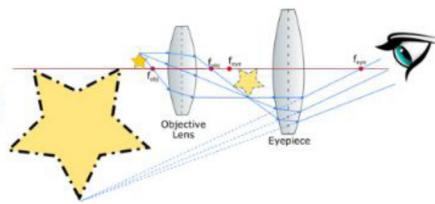


Fig. 1. Optical Microscope Algorithm (OMA) illustration.

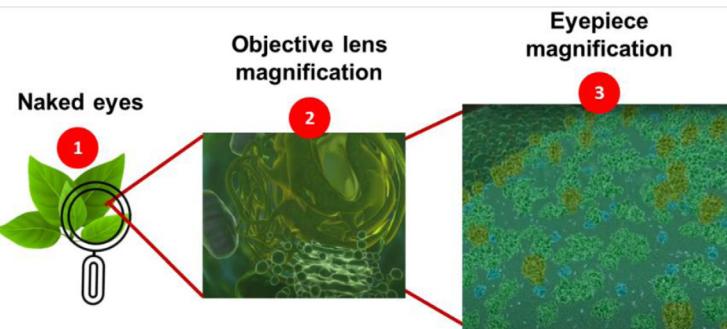


Fig. 2. Zoom-in process using a microscope.

2.2.1. Objective lens magnification

The magnification of the target object in this algorithm follows the magnification principle used by compound microscopes (i.e., total visual magnification) and is modeled using Eq. (1) [40].

$$M_{total} = M_O * M_E \quad (1)$$

where M_{total} represents the total visual magnification of the microscope, M_O is the magnification value of the objective lens, and M_E is the magnification value of the eyepiece. The magnification equation by the objective lens is generally expressed in Eq. (2) [40].

$$M_O = \frac{L}{f_o} \quad (2)$$

where L is the tube length of the microscope and f_o is the focal length of the objective lens. To calculate both values, a reference

is needed from the location of the best target object (M_{best}), which is magnified with an objective lens. The magnifying power of the objective lens is determined based on the range of useful magnification in Table 1, where the maximum magnification is 100x with a numerical aperture (NA) value of 1.40.

The modified target object ($M_{i new}$) for this phase is mathematically expressed in Eq. (3).

$$M_{i new} = M_i + M_{O new} * M_{best} \quad (3)$$

M_i denotes the initial target object, $M_{O new}$ represents the magnifying power of the objective lens, and M_{best} is the best target object. The value of $M_{O new}$ is determined by calculating the range of object magnification by the lens, ranging from the lowest scale with a magnification power of 2.5x (NA = 0.08) to the largest scale with a magnification power of 100x (NA = 1.40). The magnification scale (m') is used for the initial magnification representing random magnification values from 0 to 1.

Table 1
Range of useful magnification [41].

Objective lens (NA)	Eyepiece (NA)	10x (0.35)	12.5x (0.38)	15x (0.42)	20x (0.44)	25x (0.55)
2.5x (0.08)	–	–	–	✓	✓	–
4x (0.12)	–	–	✓	✓	✓	–
10x (0.35)	✓	✓	✓	✓	✓	–
25x (0.55)	✓	✓	✓	✓	–	–
40x (0.70)	✓	✓	✓	–	–	–
60x (0.95)	✓	✓	✓	–	–	–
100x (1.40)	✓	✓	–	–	–	–

Note: “✓” indicates a good combination.

Therefore, Eq. (3) can be given as follows:

$$M_{i \text{ new}} = M_i + m^r * 1.40 * M_{\text{best}} \quad (4)$$

The modified target object ($M_{i \text{ new}}$) is then compared with the current object, and the better of the two is selected as the best magnification (M_{best}).

2.2.2. Eyepiece magnification

The second lens of the microscope is the eyepiece, which is used to magnify the object after the objective lens. The magnification equation for the eyepiece is generally expressed in Eq. (5) [40].

$$M_0 = \frac{D}{f_e} \quad (5)$$

where D is the least distance of vision, and f_e is the focal length of the eyepiece. The eyepiece phase is a more specific phase of advanced magnification. Therefore, to determine the length of both, a reference is needed from the distance of the local search space, which is enlarged with an eyepiece. To simulate the magnification effect of the eyepiece, the magnification space is determined based on the distance between the selected target object (i) and the other target object from the population (j). The target object (j) is chosen randomly to calculate the local search space.

This modified magnification is considered an effective exploitation of the local search space. Eqs. (6) and (7) are used to simulate the magnification and modification patterns of the target object, respectively.

$$\text{space} = \begin{cases} M_j - M_i & \text{if } f(M_i) \geq f(M_j) \\ M_i - M_j & \text{if } f(M_i) < f(M_j) \end{cases} \quad (6)$$

$$M_{i \text{ new}} = M_i + MP_E * \text{space} \quad (7)$$

MP_E represents the magnifying power of the eyepiece, and space denotes the local search space for magnification. The value of MP_E is determined by calculating the range of object magnification by the lens, ranging from the lowest scale with a magnification power of 10x (NA = 0.35) to the highest scale with a magnification power of 25x (NA = 0.55).

Like the objective lens phase, the magnification scale (m^r) is used for the initial magnification, representing random magnification values from 0 to 1. The degree of reduction in the boundary search is determined by the magnification power, which is obtained from the eyepiece's numerical aperture (NA) value piece (0.55 in Table 1) multiplied by space . Therefore, Eq. (7) can be expressed as follows:

$$M_{i \text{ new}} = M_i + m^r * 0.55 * \text{space} \quad (8)$$

Afterward, the modified target object ($M_{i \text{ new}}$) is subsequently compared with the current object, and the one yielding a better result is selected as the best magnification (M_{best}).

2.2.3. Schematic representation of OMA

Exploration and exploitation are the two main phases of a metaheuristic algorithm. In the proposed OMA, the magnification by the objective lens represents exploration, while the magnification of eyepiece represents exploitation. To illustrate all the concepts, the schematic flowchart and pseudocode of OMA are described in Figs. 3 and Fig. 4, respectively.

2.2.4. Implementation of OMA for numerical optimization

The stepwise procedure used to implement OMA is described in this section. The sphere function was chosen as the mathematical function to illustrate the step-by-step procedure used by OMA to solve numerical problems.

The sphere function is suitable for single-objective optimization, meaning that it presents a single-objective function. Additionally, this function is unimodal, meaning that it has one mode and a single global optimum, depicted in the landscape of a sphere function with two variables: x_1 and x_2 [42]. The initialization range and the number of design variables (D) for this function were set in this study at $[-100, 100]$ and 2, respectively. Eq. (9) is used to describe the dimensional domain, while Fig. 5 depicts the sphere function's three-dimensional and contour plots.

$$f(x) = \sum_{i=1}^n x_i^2 \quad (9)$$

This equation defines the sphere function $f(x)$, where x is a vector with n elements. The symbol x_i represents the individual elements of the vector x , and the subscript i indicates the index of each element.

The steps used to implement OMA in numerical optimization in this study are as follows:

Step 1: OMA parameters

Optimization parameters considered for sphere functions include the number of target objects that should be magnified (NP) = 10 and the maximum number of iterations (max_iter) = 30. The stopping criteria include reaching the maximum number of iterations and finding a global optimum below 1E-06.

Step 2: Identify target objects

The initial target object is produced by generating a uniform random number from the benchmark function's range and dimension (D). Target objects are expressed as follows:

$$\text{Target objects} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9 \\ M_{10} \end{bmatrix} = \begin{bmatrix} 40.8932 & 89.1521 \\ 33.4127 & 64.5785 \\ 6.5864 & -32.5616 \\ -29.6397 & 33.8342 \\ 94.9544 & -79.6972 \\ -68.4413 & 76.3690 \\ -11.8065 & 97.6836 \\ -70.8200 & -7.9964 \\ 30.4242 & -41.3835 \\ 10.7947 & -99.8983 \end{bmatrix}$$

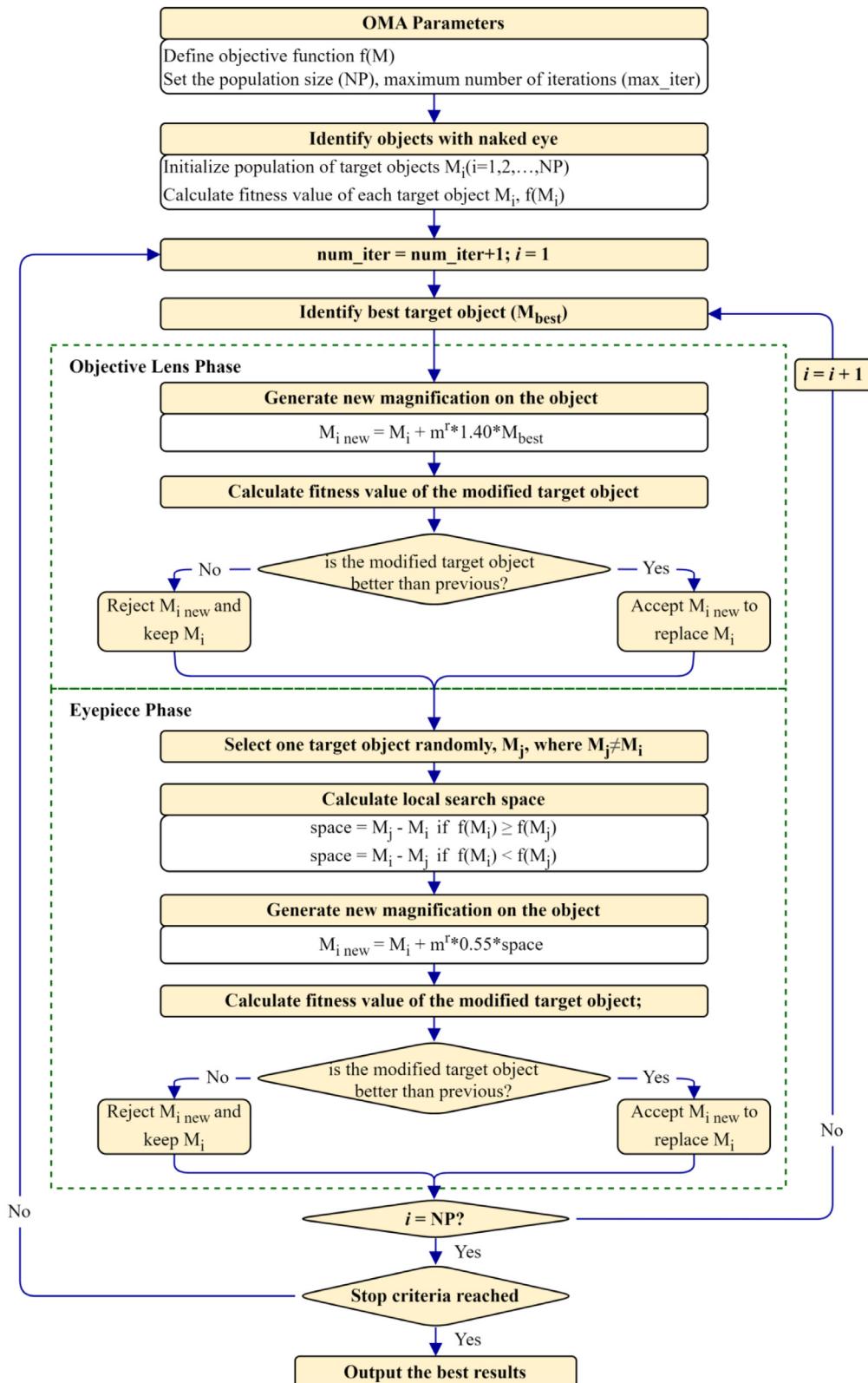


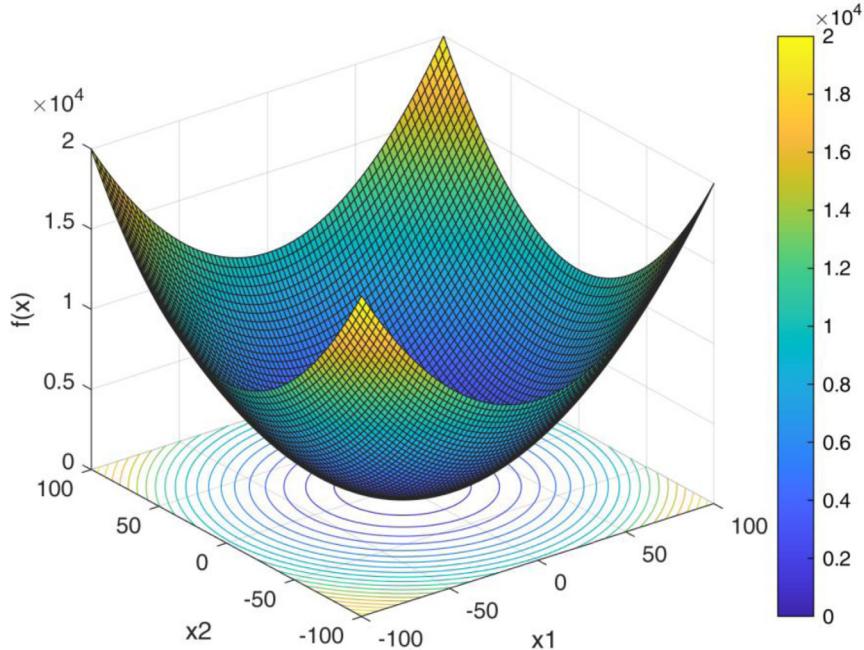
Fig. 3. Schematic flowchart for the Optical Microscope Algorithm (OMA).

Algorithm: Optical Microscope Algorithm (OMA)

```

1 Define the objective function  $f(M)$ 
2 Set the population size ( $NP$ ) and maximum iteration ( $max\_iter$ )
// Identify objects with the naked eye
3 Initialize the population of target objects  $M_i (i=1,2,...,NP)$ 
4 Calculate the fitness value of each target object  $M_i, f(M_i)$ 
5  $num\_iter = num\_iter + 1; i=1$ 
6 REPEAT
7   For  $i=1:NP$  do
8     Identify the best target object ( $M_{best}$ )
// PHASE 1: Objective lens magnification
9     Generate new magnification on the object  $M_{i\ new}$  using Eq. (4)
10    Calculate the fitness value and update target object
// PHASE 2: Eyepiece magnification
11    Select one target object randomly,  $M_j$ , where  $M_j \neq M_i$ 
12    Calculate the local search space using Eq. (6)
13    Generate new magnification on the object  $M_{i\ new}$  using Eq. (8)
14    Calculate the fitness value and update target object
15  End for i
16 UNTIL the stop criteria are met (e.g.,  $num\_iter > max\_iter$ )
17 Output the best results
18 END

```

Fig. 4. Pseudocode for OMA.**Fig. 5.** Three-dimensional and contour plots for the sphere function.

and the fitness value =

$$\begin{bmatrix} 0.9620 \\ 0.5287 \\ 0.1104 \\ 0.2023 \\ 1.5368 \\ 1.0516 \\ 0.9681 \\ 0.5079 \\ 0.2638 \\ 1.0096 \end{bmatrix} \times 10^4$$

M_3 earned the lowest minimum fitness value among all the target objects and was thus chosen as M_{best} .

Step 3: Objective lens magnification

- $M_i = [40.8932 89.1521]$, with i initially set to 1, and point M_1 matched to M_i .
- Modified target object M_1 is calculated using Eq. (4). $M_{1\ new} = M_1 + rand(0,1) * 1.40 * M_3 = [48.2700 80.0348]$, with the corresponding fitness value = $[0.8736] \times 10^4$.

- The modified target object M_1 is compared to the current M_1 , with the better fitness value selected as the solution for the next iteration. As the modified M_1 was better than the current one, M_1 was updated to [48.2700 80.0348].

Step 4: Eyepiece magnification

- $M_i = [40.8932 89.1521]$, with i initially set to 1, and point M_1 matched to M_i . M_7 is randomly selected from the population.
- $space$ is calculated using Eq. (6). $space = M_7 - M_1 = [-52, 6997 8.5315]$ because $f(M_1) \geq f(M_7)$.
- Modified target object M_1 is calculated using Eq. (8). $M_{1\ new} = M_1 + rand(0, 1) * (0.55 * space) = [20.6032 90.0906]$, with the corresponding fitness value $= [0.8541] \times 10^4$.
- The modified target object M_1 is compared to the current M_1 , with the better fitness value selected as the solution for the next iteration. As the modified M_1 was better than the current one, M_1 was updated to [20.6032 90.0906].

Step 5: Go to step 3 if the current M_i is not the last target object to be zoomed in; otherwise, proceed to the next step.

Step 6: Stop if one of the termination criteria is reached; otherwise, return to step 3 and continue with the next iteration.

3. OMA validation

Several numerical optimization problems from the literature are examined in this study to validate the performance of the proposed OMA. This section is divided into two subsections. Section 3.1 describes a set of complex mathematical benchmark problems that will be used to test the performance of OMA against other metaheuristic algorithms. It also compares the performances of OMA and other algorithms in solving benchmark functions. Section 3.2 compares OMA in completing case studies.

3.1. Mathematical benchmark problems

OMA should be validated against appropriate benchmark functions, which are widely used for calculating and validating the optimization performance of algorithms. These functions provide global optimal values that can be used to evaluate and compare algorithm performance.

3.1.1. Mathematical benchmark functions

In this experiment, 50 benchmark functions (labeled as F1 – F50) referenced from the literature [25,43] were used to evaluate OMA. The functions are presented in detail in Table 2, and the 3D view of selected benchmark functions is shown in Fig. 6.

All of the benchmark functions can be distinguished into unimodal/multimodal and separable/non-separable categories, ranging from 2 to 30 dimensions. Unimodal test functions, having a single optimum are used to compare the exploitation and convergence of an algorithm. On the other hand, multimodal functions are used to test the ability of algorithms to eliminate local minima [44].

OMA was tested on one of the benchmark functions. For example, the convergence curve of the Ackley is presented in Fig. 7. Ackley is a non-convex function characterized by a nearly flat outer region and a large hole in the middle [45]. The visualization of the convergence process in two dimensions, with a population size of 10 and 30 iterations is shown in Fig. 8.

The abovementioned benchmark functions were used in this study for validation and comparison against nine other algorithms taken from well-known metaheuristics, including GA and

Table 2
Benchmark functions for experimentation.

No.	Function	Range	D	Opt.	Type
F1	Stepint	[-5.12, 5.12]	5	0	US
F2	Step	[-100, 100]	30	0	US
F3	Sphere	[-100, 100]	30	0	US
F4	SumSquares	[-10, 10]	30	0	US
F5	Quartic	[-1.28, 1.28]	30	0	US
F6	Beale	[-4.5, 4.5]	2	0	UN
F7	Easom	[-100, 100]	2	-1	UN
F8	Matyas	[-10, 10]	2	0	UN
F9	Colville	[-10, 10]	4	0	UN
F10	Trid 6	[-D ² , D ²]	6	-50	UN
F11	Trid 10	[-D ² , D ²]	10	-210	UN
F12	Zakharov	[-5, 10]	10	0	UN
F13	Powell	[-4, 5]	24	0	UN
F14	Schwefel 2.22	[-10, 10]	30	0	UN
F15	Schwefel 1.2	[-100, 100]	30	0	UN
F16	Rosenbrock	[-30, 30]	30	0	UN
F17	Dixon–Price	[-10, 10]	30	0	UN
F18	Foxholes	[-65.536, 65.536]	2	0.998	MS
F19	Branin	[-5, 10] × [0, 15]	2	0.398	MS
F20	Bohachevsky 1	[-100, 100]	2	0	MS
F21	Booth	[-10, 10]	2	0	MS
F22	Rastrigin	[-5.12, 5.12]	30	0	MS
F23	Schwefel	[-500, 500]	30	-12,569.5	MS
F24	Michalewicz 2	[0, π]	2	-1.8013	MS
F25	Michalewicz 5	[0, π]	5	-4.6877	MS
F26	Michalewicz 10	[0, π]	10	-9.6602	MS
F27	Schaffer	[-100, 100]	2	0	MN
F28	Six-Hump Camel	[-5, 5]	2	-1.0316	MN
F29	Bohachevsky 2	[-100, 100]	2	0	MN
F30	Bohachevsky 3	[-100, 100]	2	0	MN
F31	Shubert	[-10, 10]	2	-186.7309	MN
F32	GoldStein–Price	[-2, 2]	2	3	MN
F33	Kowalik	[-5, 5]	4	0.000307	MN
F34	Shekel 5	[0, 10]	4	-10.1532	MN
F35	Shekel 7	[0, 10]	4	-10.4029	MN
F36	Shekel 10	[0, 10]	4	-10.5364	MN
F37	Perm	[-D, D]	4	0	MN
F38	PowerSum	[0, 1]	4	0	MN
F39	Hartman 3	[0, D]	3	-3.8628	MN
F40	Hartman 6	[0, 1]	6	-3.3224	MN
F41	Griewank	[-600, 600]	30	0	MN
F42	Ackley	[-32, 32]	30	0	MN
F43	Penalized	[-50, 50]	30	0	MN
F44	Penalized 2	[-50, 50]	30	0	MN
F45	Langermann 2	[0, 10]	2	-1.0809	MN
F46	Langermann 5	[0, 10]	5	-1.5	MN
F47	Langermann 10	[0, 10]	10	-1.34	MN
F48	Fletcher–Powell 2	[-π, π]	2	0	MN
F49	Fletcher–Powell 5	[-π, π]	5	0	MN
F50	Fletcher–Powell 10	[-π, π]	10	0	MN

D = dimension (D), Opt. = optimal value, U = unimodal, M = multimodal, S = separable, N = non-separable.

DE (evolutionary-based), PSO and ABC (swarm-based), SOS, BES and JS (nature-inspired), GSA (physics-based), and FBI (human-activity based). JS (2021), FBI (2020), and BES (2020) are the three most recently popularized algorithms.

The use of the same number of iterations is harmful to algorithm comparisons [46]. The maximum number of evaluations, commonly known as the maximum number of fitness function evaluations, is one of the most widely used stopping criteria. Three parameters—population size, the number of phases, and the number of iterations—are all multiplied to determine the total number of evaluations [47]. Following Karaboga and Bahriye [43], the benchmark functions were set to a population size of 50 and 500,000 for the maximum number of evaluations. In order to ensure comparison consistency, all metaheuristic algorithms were also evaluated under the same conditions.

Internal parameter settings for the compared metaheuristic algorithms are shown in Table 3. Parameters are key to algorithms, as they are learned from historical training data [48].

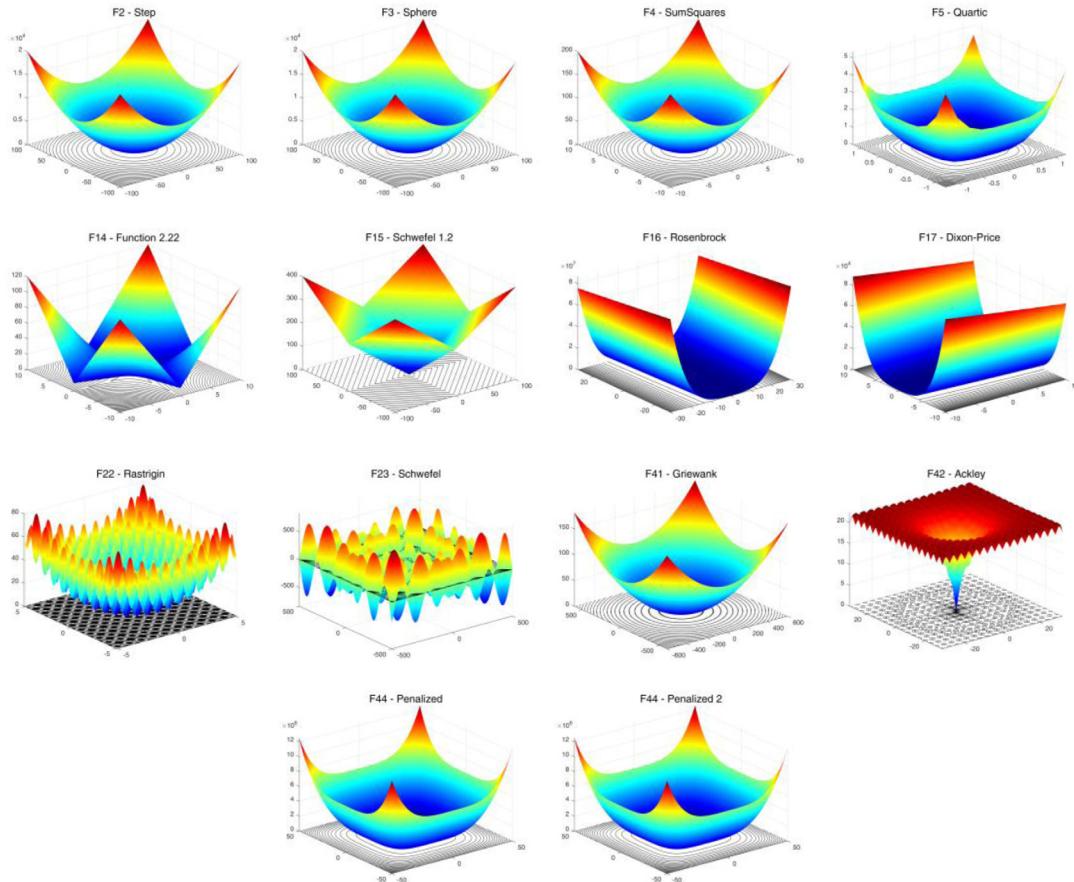


Fig. 6. 3D view of selected benchmark functions ($D = 30$).

Table 3
Parameter settings for compared metaheuristic algorithms.

Algorithm	Parameters
Optical Microscope Algorithm (OMA)	NP; NE
Jellyfish Search (JS)	NP; NE
Forensic-Based Investigation (FBI)	NP; NE
Bald Eagle Search (BES)	NP; NE
Symbiotic Organisms Search (SOS)	Position change controller (α) = 2.0; Point search (a) = 10.0; Number of search cycles (R) = 1.5
Gravitational Search Algorithm (GSA)	NP; NE
Artificial Bee Colony (ABC)	NP; NE
Differential Evolution (DE)	Power of R = 0.7; Rnorm = 0.2
Particle Swarm Optimization (PSO)	NP; NE
Genetic Algorithm (GA)	Trial limit = NP * dimensions Real constant F = 0.5; Crossover rate = 0.9 Inertia weight (ω) = 1.0; Inertia weight damping ratio (ω damp) = 0.99; Personal learning coefficient (c1) = 1.5; Global learning coefficient (c2) = 2.0 NP; NE Crossover rate = 0.8; Mutation rate = 0.01

NP = population size, NE = maximum number of evaluations.

To guarantee a fair comparison, all metaheuristic algorithms were executed in MATLAB R2022a on a PC with an Intel Core i5-8400 CPU, a clock speed of 2.80 GHz, 16.0 GB of RAM, and Windows 10.

The performance of each algorithm in terms of solving the benchmark functions is illustrated in Table 4. The bold and italic numbers represent the comparatively best values. The mean and standard deviation for OMA and other algorithms were obtained after 30 independent runs, using the standard established in a previous study [43].

The Wilcoxon signed-rank test statistic was used for statistical analysis. This test assesses the difference between two samples and offers an alternative location-based test that considers the sizes and directions of these disparities [49]. This test addresses the hypotheses using Eqs. (10) and (11).

$$H_0: \text{mean}(A) = \text{mean}(B) \quad (10)$$

$$H_1: \text{mean}(A) \neq \text{mean}(B) \quad (11)$$

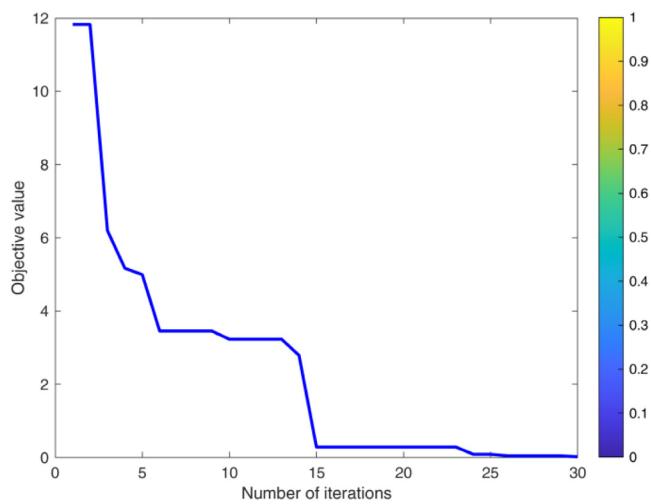


Fig. 7. Example of visualization and convergence (Ackley function).

where A and B represent the outcomes of the first and second algorithms, respectively. This test also examines whether one algorithm surpasses the other by Eqs. (12) and (13).

$$R^+ = \sum_{d_i > 0} rank(d_i) + \frac{1}{2} \sum_{d_i=0} rank(d_i) \quad (12)$$

$$R^- = \sum_{d_i < 0} rank(d_i) + \frac{1}{2} \sum_{d_i=0} rank(d_i) \quad (13)$$

d_i represents the difference between the performance values of the two algorithms in solving the i th out of n problems. R^+ represents the sum of ranks for the problems where the first algorithm outperforms the second. While R^- represents the sum of ranks for the problems where the second algorithm outperforms the first. The ranks of $d_i = 0$ are evenly distributed among these sums. In the case of an odd number of sums, one of them is disregarded.

MATLAB was used to determine the p value for comparing the algorithms at a significance level of $\alpha = 0.05$. The null hypothesis is rejected when the p value is below the specified significance level. R^+ represents an algorithm with a higher mean that exhibits superiority over the algorithm across different sets of experiments. If ($R^+ = \frac{n \times (n+1)}{2}$), this algorithm consistently outperforms all others across all experiments.

The term “winner” in Table 4 indicates the winner between the OMA and other algorithms using the Wilcoxon signed-rank test. The symbols “+” and “-” represent that the OMA result is significantly better and worse than the comparative algorithm. Meanwhile, the symbol “~” indicates an insignificant result.

F1 to F5 are unimodal because each has only one optimal value. OMA found the global optimum of four out of the five functions. This result is excellent because no algorithm was able to reach the global optimum of F5. OMA was the winner for F5 over other algorithms by finding the lowest global optimum.

Functions F6 to F17 are non-separable, unimodal functions. Finding the global optimum of a non-separable function is more difficult than finding the global optimum of a separable function [50]. OMA and JS performed the best overall by most efficiently identifying the global optimum of 11 of the 12 functions.

Functions F18 to F26 are multimodal and separable problems. OMA, JS, FBI, and SOS performed best overall by identifying 8 of the 9 functions. Although none of the algorithms found the global optimum function F23, OMA and FBI earned the closest values.

Functions F27 to F50 are multimodal and non-separable problems and pose the most-difficult tests for optimization algorithms [51]. OMA, JS, FBI, and SOS outperformed the other algorithms by identifying 21 of the 24 functions.

3.1.2. OMA vs. other algorithms in solving benchmark functions

The results obtained from the mathematical tests are summarized in Table 5. OMA, JS, and FBI outperformed all of the other algorithms by achieving the global optimum 44 out of 50 times (88%) for the benchmark functions. The six functions for which neither was able to achieve the global optimum were F5-Quartic, F16-Rosenbrock, F23-Schwefel, F37-Perm, F38-PowerSum, and F47-Langermann10. The other closest competitor is SOS, having achieved a global optimum score of 43 out of 50 (86%).

In terms of the lowest values achieved when the global optimum was not achieved, OMA was significantly more powerful than the other metaheuristic algorithms. OMA won over all of the other algorithms with the best performance of 94%. JS and FBI came in second place with a performance of 90%, while SOS was third with 88%. The results of the statistical analysis conducted with the Wilcoxon signed-rank test indicate that OMA performs better than JS (4 functions), FBI (4 functions), and SOS (6 functions). However, OMA performs slightly worse than JS, FBI, and SOS, with 1, 2, and 1 function differences, respectively.

OMA exhibited consistent superiority over other algorithms throughout the extensive 1500 trials, with 30 trials conducted for each of the 50 functions. Remarkably, OMA achieved an impressive success rate of 1236 out of the total 1500 trials, surpassing the notable performance of JS and SOS, which obtained 1204 and 1135 successful solutions, respectively.

In addition to benchmark functions, OMA performed exceptionally well in terms of computational time. The total time required by OMA on the 50 benchmark functions was 65.93 s, making it the second-fastest algorithm after JS.

The convergence capability of the algorithms was also compared in this research. Because an algorithm must be able to escape the local optima by balancing between exploration and exploitation to obtain the global optimum [52], the optimization process should not converge prematurely on the local optimum [53]. The OMA convergence curves on unimodal and multimodal benchmark functions are compared with those of JS, FBI, BES, SOS, GSA, ABC, DE, PSO, and GA in Fig. 9. The convergence curve illustrates the value of the objective function versus the computational time during optimization [54].

These figures show that OMA outperformed the other algorithms in the illustrated cases. The unimodal function curves (F3, F6, and F17) show that OMA has good characteristics that allow the quick and undisturbed exploitation of promising areas. Similar experimental results were obtained by OMA for the single multimodal function (F23, F26, and F46), demonstrating the ability of OMA to switch from local optima very quickly.

The box and whisker plot is chosen to visually represent the distributions of the objective value across multiple runs, as it offers a wealth of information including the minimum, median, second or third quartile, and maximum values of the testing samples. Fig. 10 represents the box plot depicting the performance of various algorithms. Notably, the objective distributions of the OMA method are significantly smaller than those of the other algorithms, thus showcasing its robust performance in the test problems.

Fig. 11 displays the search histories of the OMA across multiple benchmark functions. The observations indicate that in the initial stages of iteration, the solutions are evenly distributed throughout the problem space. However, as the iterations progress, the

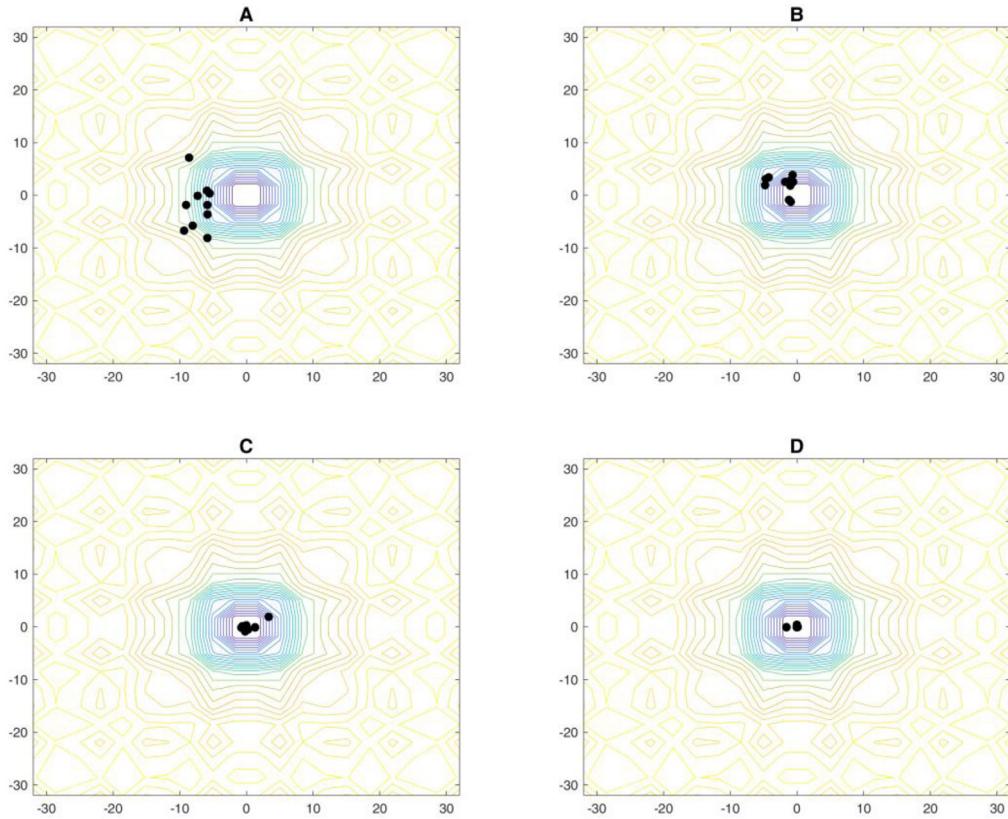


Fig. 8. Visualization of the convergence process for the Ackley function, (A) Iteration 1, (B) Iteration 5, (C) Iteration 15, (D) Iteration 30.

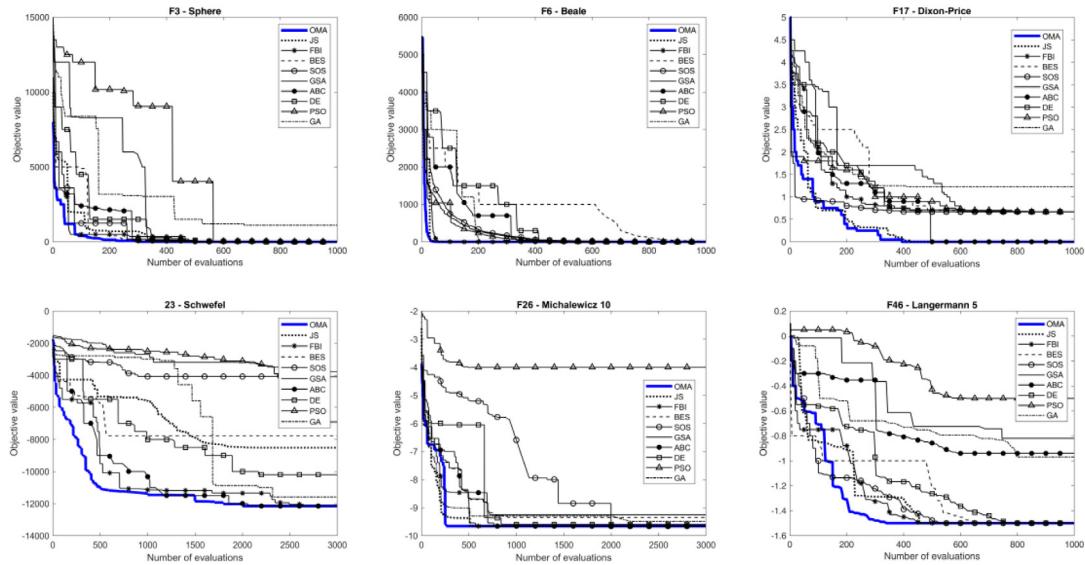


Fig. 9. Convergence curves for selected benchmark functions.

solutions concentrate their search efforts in the most promising area. Eventually, a majority of the target objects or population converge towards the global optimal solutions. This demonstrates that the OMA approach effectively enhances the quality of randomly-placed solutions by striking a balance between exploration and exploitation.

The advantage of OMA optimization process reflects the zoom-in process, which magnifies and focuses the object until the most optimal target object is obtained. Based on the results of the analysis and visualization, the OMA is well able to balance the

needs of exploration and exploitation. The convergence graphs show clearly that OMA improves the quality of the solution and moves from the local optima using the objective lens in the initial phase, and then quickly moves towards convergence with the global optimum using the eyepiece.

3.2. Case studies: Engineering design problems

This section examines the performance of OMA using two engineering design optimization problems, including structural

Table 4

Comparative results of OMA and other algorithms for 50 functions.

Fun.	Index	OMA	JS	FBI	BES	SOS	GSA	ABC	DE	PSO	GA
F1	Mean	0	0	0	0	0	0	0	0	0	0
	Std.	0	0	0	0	0	0	0	0	0	0
	t (s)	0.42	0.56	0.51	1.93	0.61	10.8	3.01	2.71	2.75	2.99
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F2	Mean	0	0	0	0	0	0	0	0	0	11.7
	Std.	0	0	0	0	0	0	0	0	0	76.7
	t (s)	0.67	0.54	0.91	2.11	0.44	14.97	3.06	2.82	2.96	3.11
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	+
F3	Mean	0	0	0	0	0	0	0	0	0	1110
	Std.	0	0	0	0	0	0	0	0	0	74.30
	t (s)	0.56	0.64	1.22	2.17	0.69	14.08	3.26	3.05	2.95	3.38
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	+
F4	Mean	0	0	0	0	0	0	0	0	0	148
	Std.	0	0	0	0	0	0	0	0	0	12.40
	t (s)	0.82	0.52	1.19	2.08	1.07	15.15	2.93	2.73	2.72	2.95
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	+
F5	Mean	1.23E-06	8.37E-05	1.68E-06	1.52E-06	4.03E-05	0.0115	3.00E-02	1.35E-03	1.16E-03	0.181
	Std.	1.02E-06	1.04E-04	2.00E-06	2.60E-04	2.09E-04	2.60E-03	4.74E-03	4.12E-03	2.76E-04	0.0271
	t (s)	2.64	2.09	2.98	2.69	5.01	15.38	4.30	3.85	3.91	4.45
	Winner	+	+	+	+	+	+	+	+	+	+
F6	Mean	0	0	0	0	0	0	0	0	0	0
	Std.	0	0	0	0	0	0	0	0	0	0
	t (s)	0.40	0.56	0.43	1.86	0.77	7.34	2.90	2.72	2.72	2.95
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F7	Mean	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	Std.	0	0	0	0	0	0	0	0	0	0
	t (s)	0.63	0.44	0.43	1.85	0.67	7.31	3.01	2.52	2.69	2.85
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F8	Mean	0	0	0	0	0	0	0	0	0	0
	Std.	0	0	0	0	0	0	0	0	0	0
	t (s)	0.45	0.44	0.51	1.81	0.63	7.31	2.90	2.16	2.18	2.47
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F9	Mean	0	0	0	0	0	0	9.31E-02	4.08E-02	0	0.0149
	Std.	0	0	0	0	0	0	6.65E-02	8.16E-02	0	7.36E-03
	t (s)	0.39	0.44	0.43	1.83	1.40	7.6	3.01	2.25	2.24	2.53
	Winner	≈	≈	≈	≈	≈	≈	+	+	≈	+
F10	Mean	-50	-50	-50	-50	-50	-50	-50	-50	-50	-50
	Std.	0	0	0	0	0	0	0	0	0	0
	t (s)	0.38	0.38	0.44	0.44	1.87	7.10	3.15	2.48	2.86	2.31
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F11	Mean	-210	-210	-210	-210	-210	-210	-210	-210	-210	-210
	Std.	0	0	0	0	0	0	0	0	0	0
	t (s)	0.43	0.48	0.53	1.90	0.89	8.39	3.12	2.49	2.39	2.70
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F12	Mean	0	0	0	0	0	0	2.45E-04	0	0	0.0134
	Std.	0	0	0	0	0	0	1.87E-04	0	0	4.53E-03
	t (s)	0.71	0.50	0.82	1.89	0.61	7.41	3.10	2.51	2.52	2.85
	Winner	≈	≈	≈	≈	≈	≈	+	≈	≈	+
F13	Mean	0	0	0	0	0	0	1.94E-04	3.16E-03	2.14E-07	1.10E-04
	Std.	0	0	0	0	0	0	6.73E-05	5.11E-04	1.28E-07	1.60E-04
	t (s)	1.29	1.13	1.63	2.26	3.82	11.33	3.56	3.10	3.15	3.44
	Winner	≈	≈	≈	≈	≈	≈	+	+	+	+
F14	Mean	0	0	0	0	0	0	0	0	0	11.00
	Std.	0	0	0	0	0	0	0	0	0	1.39
	t (s)	0.71	0.53	1.07	2.07	0.63	12.05	3.11	2.75	2.70	3.05
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	+
F15	Mean	0	0	0	0	0	0	0	0	0	7.40E+03
	Std.	0	0	0	0	0	0	0	0	0	1.14E+03
	t (s)	3.65	1.52	1.60	3.64	2.10	13.19	3.48	3.27	3.22	3.64
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	+
F16	Mean	1.56E-02	8.12E-01	2.75E-08	4.56E-05	0.19	20.41	8.83E-02	18.20	15.12	1.96E+05
	Std.	6.76E-01	1.55E-01	1.00E-06	9.09E-03	0.15	0.22	7.75E-02	5.04	24.23	3.85E+04
	t (s)	0.61	0.52	0.9	2.19	3.09	15.12	3.08	2.67	2.69	2.93
	Winner	+	-	-	+	+	+	+	+	+	+
F17	Mean	0	0.6667	0.6667	0.6667	0.6667	0	0.6667	0.6667	0.6667	1.222
	Std.	1.94E-11	0	0.0277	0	0	0	1.00E-09	1.00E-07	0.6667	0.6667
	t (s)	0.60	0.62	0.82	2.08	3.09	14.12	3.18	2.47	2.57	2.90
	Winner	≈	+	+	+	+	+	≈	+	+	+
F18	Mean	0.998	0.998	0.998	0.998	0.998	0.998	1.633	0.998	0.998	0.998
	Std.	0	0	0	0	0	0	0.703	0	0	0
	t (s)	3.93	3.34	3.79	2.89	5.87	9.83	4.28	3.64	3.97	4.26
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈

(continued on next page)

Table 4 (continued).

Fun.	Index	OMA	JS	FBI	BES	SOS	GSA	ABC	DE	PSO	GA
F19	Mean	0.398									
	Std.	0									
	t (s)	0.38	0.50	0.41	1.92	3.42	9.34	2.98	2.37	2.39	2.58
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F20	Mean	0									
	Std.	0									
	t (s)	0.46	0.43	0.48	1.83	0.54	9.34	3.06	2.33	2.30	2.60
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F21	Mean	0									
	Std.	0									
	t (s)	0.47	0.46	0.40	1.83	0.67	9.33	2.98	2.20	2.31	2.61
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F22	Mean	0	0	0	0	0	12.9	0	11.70	44.00	52.90
	Std.	0	0	0	0	0	3.28	0	2.54	11.70	4.56
	t (s)	0.92	0.51	1.24	2.09	0.71	15.08	3.26	3.04	2.97	3.35
	Winner	≈	≈	≈	≈	≈	+	≈	+	+	+
F23	Mean	-12151	-9026.9	-12151	-8431.74	-12150.5	-2795.59	-12151	-10266	-6909.14	-11593.4
	Std.	0	-7510.3	0	1443.9	4.28E-04	531	0	5.22E+02	458	93.3
	t (s)	1.12	1.17	1.43	2.29	3.77	15.05	3.75	3.30	3.39	3.60
	Winner	+	≈	+	+	+	≈	+	+	+	+
F24	Mean	-1.8013	-1.5729	-1.8013							
	Std.	0	0.12	0							
	t (s)	0.48	0.5	0.55	1.89	0.82	8.33	3.33	2.42	2.54	2.76
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	+	≈
F25	Mean	-4.6877	-4.6877	-4.6877	-4.4959	-4.6877	-4.5581	-4.6877	-4.6835	-2.4909	-4.6448
	Std.	0	0	0	0.1929	0	0.0821	0	1.25E-02	0.257	0.0979
	t (s)	0.74	0.59	0.75	1.96	3.00	8.79	3.28	2.71	2.69	2.93
	Winner	≈	≈	+	≈	+	≈	+	+	+	+
F26	Mean	-9.6602	-9.6602	-9.6602	-9.2194	-9.6602	-9.2534	-9.6602	-9.5912	-4.0072	-9.4968
	Std.	0	0	0	0.6611	0	0.175	0	6.42E-02	0.053	0.141
	t (s)	1.04	1.12	1.24	2.15	3.49	8.49	3.63	3.04	2.94	3.32
	Winner	≈	≈	+	≈	+	≈	+	+	+	+
F27	Mean	0	0	0	0	0	0.0103	0	0	0	4.24E-03
	Std.	0	0	0	0	0	0.0107	0	0	0	4.76E-03
	t (s)	0.61	0.44	0.5	1.86	0.62	8.35	3.03	2.28	2.40	2.61
	Winner	≈	≈	≈	≈	≈	+	≈	≈	≈	+
F28	Mean	-1.0316									
	Std.	0									
	t (s)	0.68	0.50	0.57	1.94	0.72	7.36	3.21	2.48	2.58	2.69
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F29	Mean	0	0.0683								
	Std.	0	0.0782								
	t (s)	0.46	0.43	0.49	1.86	1.03	7.34	2.99	2.28	2.48	2.67
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	+
F30	Mean	0									
	Std.	0									
	t (s)	0.48	0.43	0.48	1.83	1.04	7.34	2.94	2.29	2.49	2.68
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F31	Mean	-186.7309									
	Std.	0									
	t (s)	0.48	0.54	0.51	1.87	0.36	7.35	3.22	2.40	2.65	2.87
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F32	Mean	3	5.2506								
	Std.	0	5.8701								
	t (s)	0.45	0.44	0.45	1.84	1.05	7.32	2.95	2.27	2.45	2.54
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	+
F33	Mean	0.000307	0.000307	0.000307	0.000307	0.000307	0.0013	0.00043	0.00043	4.90E-04	5.62E-03
	Std.	0	0	0	0	0	3.31E-04	6.04E-05	2.73E-04	3.66E-04	8.17E-03
	t (s)	0.43	0.47	0.5	1.89	2.66	7.63	3.10	2.56	2.61	2.82
	Winner	≈	≈	≈	≈	+	+	+	+	+	+
F34	Mean	-10.1532	-10.1532	-10.1532	-10.1532	-10.1532	-7.3515	-10.1532	-10.1532	-2.087	-5.6605
	Std.	0	0	0	0	0	2.85	0	0	1.1785	3.8667
	t (s)	2.54	1.75	2.8	2.75	4.97	10.14	5.53	4.15	4.28	4.60
	Winner	≈	≈	≈	≈	+	≈	≈	+	+	+
F35	Mean	-10.4029	-1.9899	-5.3441							
	Std.	0	1.4206	3.5171							
	t (s)	2.50	2.18	2.81	2.71	1.34	10.16	5.60	4.42	4.46	4.81
	Winner	≈	≈	≈	≈	≈	≈	≈	+	+	+
F36	Mean	-10.5364	-10.5364	-10.5364	-5.1285	-10.5364	-10.5364	-10.5364	-10.5364	-1.8797	-3.8298
	Std.	0	0	0	0.7967	0	0	0	0	0.4325	2.452
	t (s)	2.46	2.16	2.75	2.72	2.20	9.16	5.62	4.36	4.51	4.74
	Winner	≈	≈	+	≈	≈	≈	≈	+	+	+

(continued on next page)

Table 4 (continued).

Fun.	Index	OMA	JS	FBI	BES	SOS	GSA	ABC	DE	PSO	GA
F37	Mean	1.80E-05	3.43E-05	2.57E-04	1.56E-03	1.31E-01	3.333	4.11E-02	2.41E-02	3.61E-02	1.93E-01
	Std.	6.40E-04	4.97E-03	2.63E-04	0.1205	2.33E-01	1.92	2.30E-02	4.59E-02	4.69E-02	3.35E-01
	t (s)	1.32	1.38	1.40	2.27	3.72	7.78	3.74	3.03	2.97	3.45
	Winner	+	+	+	+	+	+	+	+	+	+
F38	Mean	1.92E-07	2.87E-10	2.51E-07	1.28E-03	8.01E-07	1.65E-02	2.96E-03	1.44E-04	11.4	1.04E-02
	Std.	6.51E-05	1.07E-06	3.12E-06	7.92E-01	7.05E-05	1.68E-02	2.28E-03	1.43E-04	7.36	9.08E-03
	t (s)	0.94	0.67	0.99	2.02	3.29	8.72	3.42	2.87	2.88	3.12
	Winner	-	+	-	+	+	+	+	+	+	+
F39	Mean	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.6334	-3.8628
	Std.	0	0	0	0	0	0	0	0	0.117	0
	t (s)	3.94	3.68	3.69	3.38	6.98	9.36	6.01	5.32	5.43	5.91
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈
F40	Mean	-3.3224	-3.3224	-3.3224	-3.3224	-3.2032	-3.3224	-3.3220	-3.2169	-1.8591	-3.2982
	Std.	0	0	0	0	0	0	0	4.76E-02	0.44	0.0501
	t (s)	5.13	3.66	5.42	3.84	8.46	10.04	8.42	6.48	6.45	7.15
	Winner	≈	≈	≈	≈	+	≈	+	+	+	+
F41	Mean	0	0	0	0	0	2.29E-03	0	1.48E-03	1.74E-02	10.6
	Std.	0	0	0	0	0	9.15E-03	0	2.92E-03	2.08E-02	1.16
	t (s)	0.99	0.59	1.15	2.11	0.71	15.44	3.57	3.22	3.21	3.40
	Winner	≈	≈	≈	≈	≈	+	≈	+	+	+
F42	Mean	0	0	0	0	0	0	0	0	1.65E-01	14.7
	Std.	0	0	0	0	0	0	0	0	4.94E-01	0.178
	t (s)	0.94	0.59	1.15	2.11	0.84	14.19	3.64	3.28	3.46	3.78
	Winner	≈	≈	≈	≈	≈	≈	≈	≈	+	+
F43	Mean	0	0	0	0	0	3.46E-03	0	0	2.07E-02	13.4
	Std.	0	0	0	0	0	1.89E-02	0	0	4.15E-02	1.45
	t (s)	7.26	5.84	7.87	5.56	2.75	14.09	7.84	6.42	6.49	7.31
	Winner	≈	≈	≈	≈	≈	+	≈	+	+	+
F44	Mean	0	0	0	2.9575	0	0	0	2.19E-03	7.68E-03	125
	Std.	0	0	0	3.30E-03	0	0	0	3.39E-03	1.63E-02	12
	t (s)	4.32	3.69	4.68	3.63	1.61	15.64	6.06	5.20	5.12	5.77
	Winner	≈	≈	+	≈	≈	≈	≈	+	+	+
F45	Mean	-1.0809	-1.0809	-1.0809	-1.0809	-1.0809	-1.0809	-1.0809	-1.0809	-0.6793	-1.0809
	Std.	0	0	0	0	0	0	0	0	0.275	0
	t (s)	0.50	0.53	0.46	1.89	0.20	8.40	3.29	2.55	2.66	2.88
	Winner	≈	≈	≈	≈	≈	≈	≈	+	≈	≈
F46	Mean	-1.5	-1.5	-1.5	-1.5	-1.5	-0.8163	-0.9382	-1.5	-0.5049	-0.9684
	Std.	0	0	0	0	0	0.0955	2.08E-04	0	0.214	0.288
	t (s)	0.73	0.58	0.77	2.01	4.03	8.79	3.44	2.70	2.72	2.95
	Winner	≈	≈	≈	≈	≈	+	≈	+	+	+
F47	Mean	-0.7977	-0.7977	-1.34	-0.669	-1.34	-0.214	-0.446	-1.0528	-2.60E-03	0.636
	Std.	0	0	0	0.0261	0	0.0833	0.1345	0.3022	3.52E-03	0.375
	t (s)	0.94	1.00	0.97	2.11	0.26	8.52	3.75	3.07	2.96	3.34
	Winner	≈	≈	≈	≈	≈	+	≈	+	+	+
F48	Mean	0	0	0	0	0	0.244	0	0	0	0
	Std.	0	0	0	0	0	0.257	0	0	0	0
	t (s)	0.56	0.49	0.55	1.87	0.26	8.39	3.25	2.61	2.61	2.93
	Winner	≈	≈	≈	≈	≈	+	≈	≈	≈	≈
F49	Mean	0	0	0	0	0	291	0.1740	5.990	1.46E+03	4.30E-03
	Std.	0	0	0	0	0	284	0.0682	7.330	1.27E+03	9.47E-03
	t (s)	1.25	0.86	1.15	2.10	0.52	8.93	3.90	3.12	3.10	3.49
	Winner	≈	≈	≈	≈	≈	+	+	+	+	+
F50	Mean	0	0	0	0	0	291	8.231	7.82E+02	1.36E+03	29.6
	Std.	0	0	0	0	0	286	8.093	1.04E+02	1.33E+03	16
	t (s)	1.12	0.88	1.26	2.22	0.58	8.60	3.86	3.26	3.48	3.59
	Winner	≈	≈	≈	≈	≈	+	+	+	+	+

Std. = Standard deviation, t (s) = computational time (unit: s).

Table 5
Results for solving the benchmark functions.

Algorithm	Achieve optimum		Best performance		Wilcoxon test (OMA vs.)			Trial number of converge	t (s)
	Total	Percentage	Total	Percentage	+	-	≈		
OMA	44	88%	47	94%				1236	65.93
JS	44	88%	45	90%	4	1	45	1204	54.31
FBI	44	88%	45	90%	4	2	44	1130	70.08
BES	39	78%	39	78%	10	1	39	1085	112.04
SOS	43	86%	44	88%	6	1	43	1135	99.88
GSA	29	58%	29	58%	21	0	29	810	511.27
ABC	37	74%	37	74%	13	0	37	1036	187.09
DE	32	36%	32	36%	17	1	42	918	154.22
PSO	24	48%	24	48%	26	0	24	675	157.17
GA	17	34%	17	34%	33	0	17	502	171.61

“+” denotes that OMA is better, “-” denotes that OMA is worse, “≈” denotes an insignificant result, t (s) = computational time (unit: s).

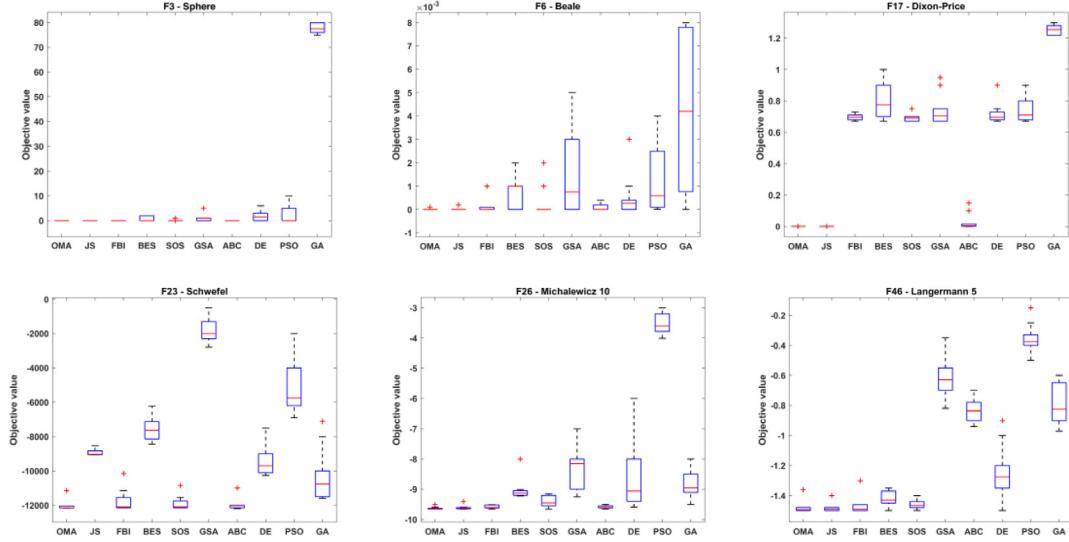


Fig. 10. Box and whiskers diagrams for selected benchmark functions.

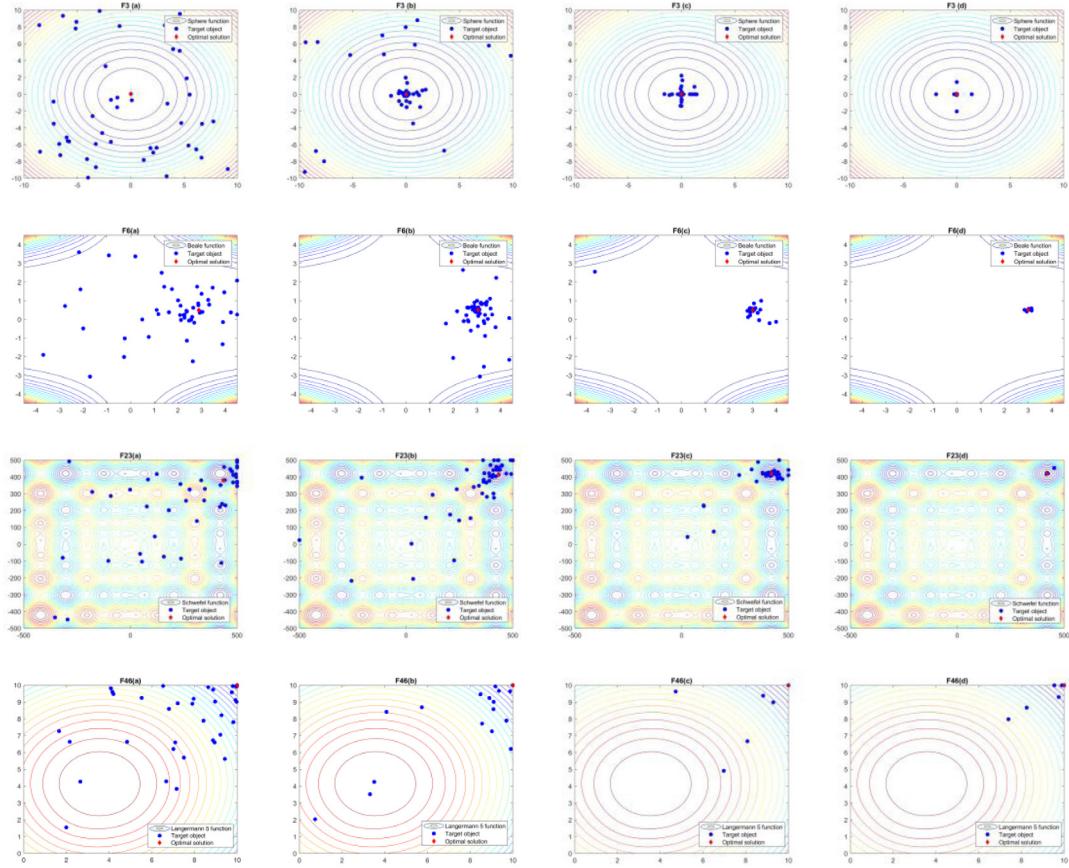


Fig. 11. Swarm plot diagrams for selected benchmark functions.

engineering design problem and multiple resources leveling in multiple projects (MRLMP). The optimization results of OMA and other algorithms are compared with data published in the literature.

3.2.1. 15-Bar truss

The ground structure of the 15-bar truss, illustrated in Fig. 12, consists of a combination of 15 members connected by 8 nodes. This model was previously optimized by [55–58]. The objective

of this case was to minimize the structure's weight, requiring the objective function $f(X)$ to be determined, as modeled in Eq. (14).

$$f(X) = \sum_{i=1}^m B_i A_i \rho_i L_i \quad (14)$$

where X is the decision variables $\{A_1; A_2; \dots; A_m; \xi_1; \xi_2; \dots; \xi_m\}$. B_i is a topological bit with a condition $B_i =$

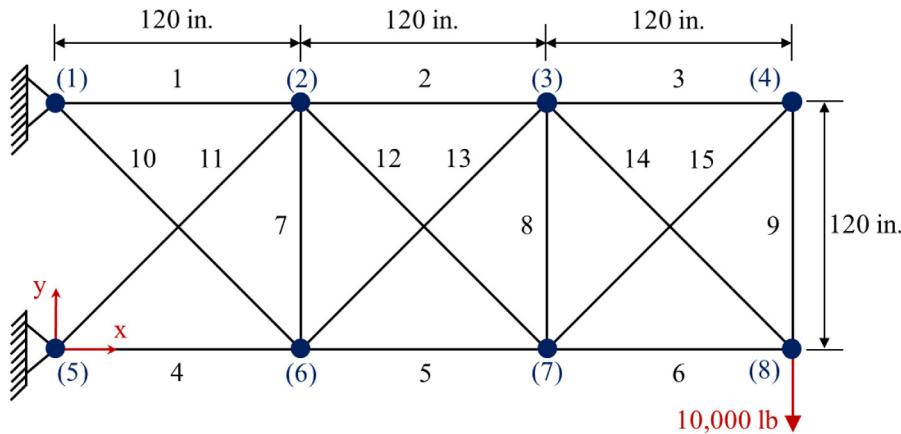


Fig. 12. Ground structure of the 15-bar truss.

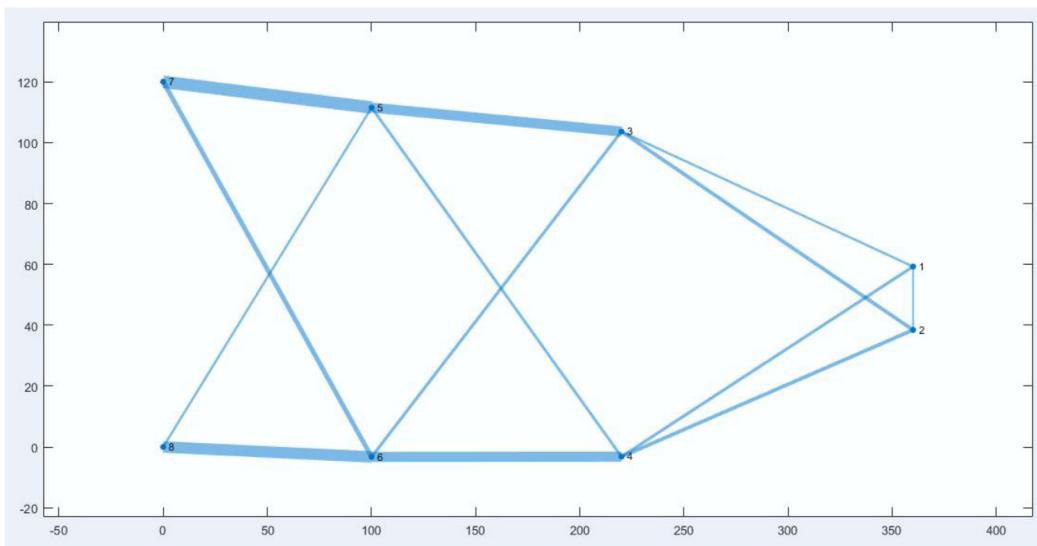


Fig. 13. Truss structure visualization with OMA.

$\begin{cases} 0, & \text{if } A_i < \text{Critical area} \\ 1, & \text{if } A_i \geq \text{Critical area} \end{cases}$, A_i is the cross-sectional area, ρ_i is density (steel density = 0.284 lb./in.³), L_i is length, E_i is modulus elasticity (steel modulus of elasticity = 29,000 ksi), and σ_i is stress.

The constraints of this case are: $g_1(X)$, used to check the validity of the structure; $g_2(X)$ used to check kinematic stability; $g_3(X)$, which was originally the member load capacity constraints, $\Phi n_i \geq P_{li}$; $g_4(X)$, which was originally displacement constraints, $|\delta_j| - |\delta_j^{max}| \leq 0$; $g_5(X)$, which is size constraints, $A_i^{critical} \leq A_i \leq A_i^{upper}$; $g_6(X)$, which is shape constraints, $\xi_j^{lower} \leq \xi_j \leq \xi_j^{upper}$; $g_7(X)$, which is tensile member load capacity constraints, $\Phi n_i \geq P_{li}$; and $g_8(X)$, which is compression member load capacity constraints, $\Phi n_i \geq P_{li}$.

The shape constraints were limited to certain moveable nodes only and explained as $100 \leq x_2 \leq 140$, $220 \leq x_2 \leq 260$, $100 \leq y_2 \leq 140$, $100 \leq y_3 \leq 140$, $50 \leq y_4 \leq 90$, $-20 \leq y_6 \leq 20$, $-20 \leq y_7 \leq 20$, and $20 \leq y_8 \leq 60$.

The truss topology and size optimization were investigated with a population size (NP) of 30. The iteration process of the algorithm under test is stopped when the value of δ is less than or equal to 10^{-6} or the maximum number of iterations reaches 1000. Otherwise, it will continue until it meets the termination criteria. Thirty independent runs are implemented for each problem due to the stochastic nature of the metaheuristic

algorithm. Furthermore, the test results are presented in the best weight, worst weight, average weight, standard deviation, and the corresponding number of structural analyses. Fig. 13 shows the best shape of the 15-bar planar truss obtained by OMA.

The comparison of the optimal designs with other references is presented in Table 6. OMA can outperform Improved-GA, FA, round-off improved ($\mu + \lambda$) constrained differential evolution (R-ICDE), and discrete improved ($\mu + \lambda$) constrained differential evolution (D-ICDE) while being able to equal the performance of SOS. Meanwhile, SOS obtained a minimum weight of 73.596 lb., more than 12900 structural analyses [58], whereas OMA obtained the same results only in 8500 structural analyses with the same parameters. The convergence history of the best topology is depicted in Fig. 14.

3.2.2. Multiple resources leveling in the multiple projects (MRLMP)

Resource leveling is used in project scheduling to minimize variability in resource usage in construction projects [59]. This variability requires temporary staff to be hired and fired to meet multiple short-term project needs. Construction project decision-makers currently rely on experience-based methods to manage fluctuations.

Two projects (project 1 and project 2) of different durations were executed to establish an enterprise, with the initial networks for both projects shown in Fig. 15.

Table 6
Optimal design comparison for the 15-bar truss structure.

Design variables	Improved-GA [55]	FA [56]	R-ICDE [57]	D-ICDE [57]	SOS [58]	OMA
Sizing variables (in. ²)						
A_1	1.081	0.954	1.081	1.081	0.954	0.954
A_2	0.539	0.539	0.539	0.539	0.539	0.539
A_3	0.287	0.220	0.270	0.141	0.141	0.141
A_4	0.954	0.954	0.954	0.954	0.954	0.954
A_5	0.954	0.539	0.954	0.539	0.539	0.539
A_6	0.220	0.220	0.220	0.287	0.270	0.347
A_7	0.111	0.111	0.111	0.111	0.111	0.111
A_8	0.111	0.111	0.111	0.111	0.111	0.111
A_9	0.287	0.287	0.287	0.141	0.141	0.174
A_{10}	0.220	0.440	0.222	0.347	0.440	0.347
A_{11}	0.440	0.440	0.444	0.440	0.440	0.347
A_{12}	0.440	0.220	0.444	0.270	0.220	0.220
A_{13}	0.111	0.220	0.174	0.270	0.270	0.220
A_{14}	0.220	0.270	0.174	0.287	0.270	0.270
A_{15}	0.347	0.220	0.347	0.174	0.141	0.270
Layout variables (in.)						
x_2	133.612	114.967	117.498	100.031	100.018	117.222
x_3	234.752	247.040	242.973	238.701	241.510	259.909
y_2	100.449	125.919	112.373	132.847	135.727	123.501
y_3	104.738	111.067	101.268	125.370	123.187	110.002
y_4	73.762	58.298	54.640	60.307	57.189	59.936
y_6	-10.067	-17.564	-12.395	-10.665	-16.331	-15.180
y_7	-1.339	-5.821	-14.391	-12.246	-8.822	-4.219
y_8	50.402	31.465	54.640	59.993	57.184	57.883
Best weight (lb.)	79.820	75.550	80.569	74.682	73.596	73.596
Worst weight (lb.)	N/A	N/A	N/A	N/A	N/A	80.156
Average weight (lb.)	N/A	N/A	N/A	N/A	79.900	76.411
Standard deviation	N/A	N/A	N/A	N/A	2.881	1.922
No. of analyses	8000	8000	7980	7980	12900	8250
Ranking	4	3	5	2	1	1

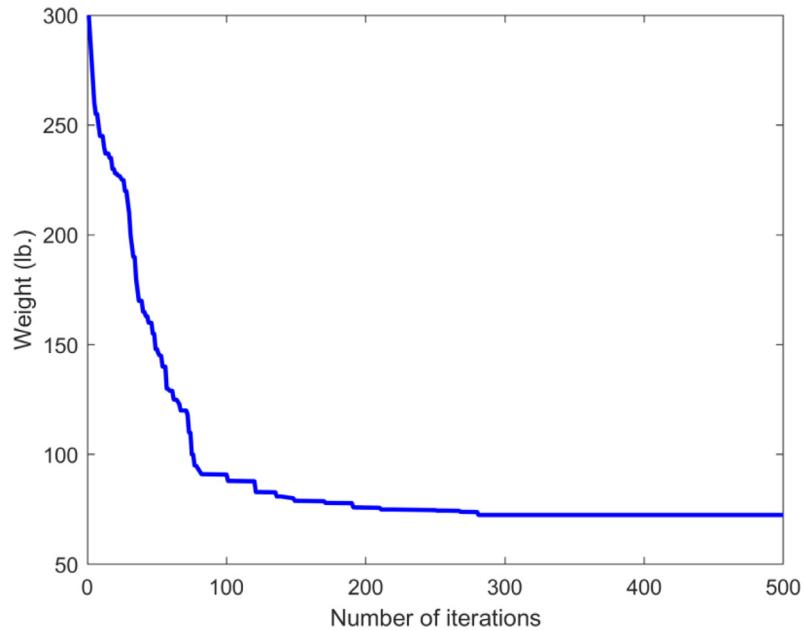


Fig. 14. Convergence history of OMA for the 15-bar truss.

The case study was adapted from Guo et al. [60]. Every activity uses two resources, including humans (R_1) and funds (R_2). Each recourse has the same importance weights $w_1 = w_2 = 0.5$. The objective of MRLMP is to minimize the daily variance in resource utilization without changing the total project duration.

The mathematic models of resource intensity (RI) and time condition (T) for this case are presented in Eqs. (15) and (16), respectively. Resource intensity is a measure of the resources required to complete a process or activity and is used as a measure of resource use efficiency [61].

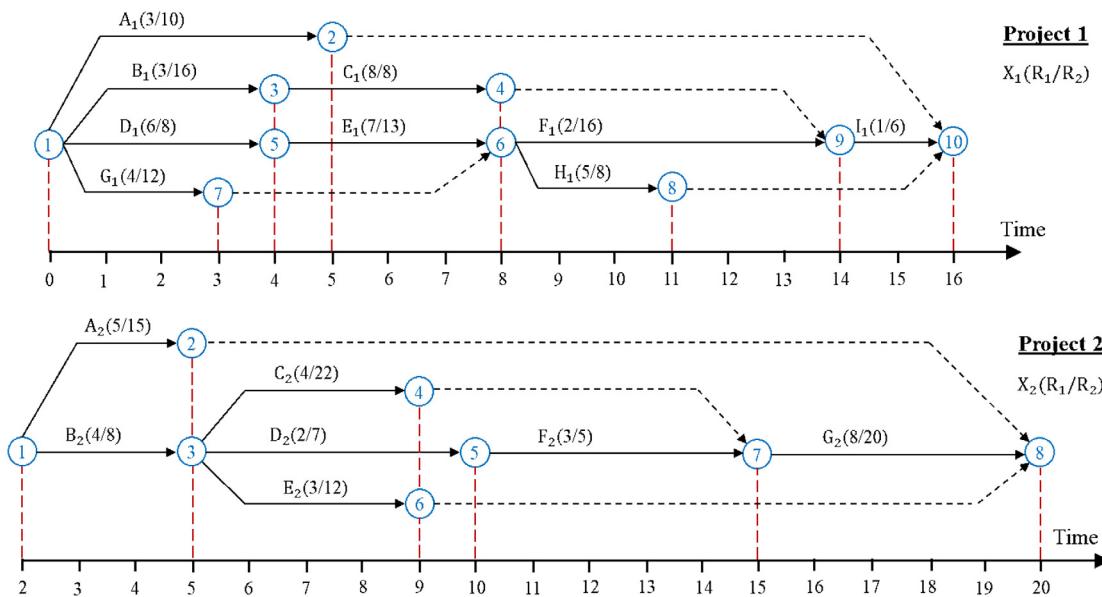


Fig. 15. Networks of two projects.

Table 7
Settings for MRLMP parameters.

Input parameters	Value
Number of decision variables (D)	12
Population size (NP)	100
Mutant factor (F)	0.5
Crossover probability (C_r)	0.8
Number of trials (limit)	NP x D
Amplification coefficient (λ)	30
Maximum iteration	200

$$\min RI = \frac{1}{20} \sum_{i=1}^{20} [0.5(SR_1(t) - \overline{SR}_1)^2 + 0.5(SR_2(t) - \overline{SR}_2)^2] \quad (15)$$

subject to (S.T) $\left\{ \begin{array}{ll} 0 \leq T_s(A_1) \leq 11 & 8 \leq T_s(H_1) \leq 13 \\ 0 \leq T_s(B_1) \leq 6 & 2 \leq T_s(A_2) \leq 17 \\ T_s(B_1) + 4 \leq T_s(C_1) \leq 10 & 5 \leq T_s(C_2) \leq 11 \\ 0 \leq T_s(G_1) \leq 5 & 5 \leq T_s(E_2) \leq 16 \end{array} \right.$ (16)

The variables $SR_1(t)$ and $SR_2(t)$ respectively represent the relative demand of humans and funds in two projects on the day (t). The symbols \overline{SR}_1 and \overline{SR}_2 indicate the average values of $SR_1(t)$ and $SR_2(t)$. T_s is defined as the difference between the maximum of late finish time and the minimum of early start time for two projects. The activities A to H correspond to the work activities within the project.

This case study used parameters based on proposed values from the literature and several experiments, as shown in Table 7. Five different algorithms were used to verify the comparative performance of OMA. GA and PSO have been used to solve this problem [60], while DE, ABC, SOS, and OMA were executed in this study.

The optimal results, optimal non-critical activity start times obtained from the proposed new model, and other benchmark algorithms for the case are listed in Table 8. The optimal resource intensities (RIs) obtained by OMA were respectively 84.36%, 30.84%, 0.7%, 0.29%, 0.21% and 0.01% less than the initial schedule, GA, PSO, DE and SOS.

The resource profile of projects by different algorithms is illustrated in Fig. 16. The resources required for the initial work,

GA, and PSO are based on Guo et al. [60]. The resources required for DE, ABC, SOS, and OMA reflect the results of this study. It can be seen in Fig. 16 that the proposed algorithm significantly reduces undesirable resource fluctuations.

Fig. 17 illustrates the best fitness value of different approaches by the number of function evaluations. OMA outperformed the other approaches in terms of convergence since the proposed model found the best solution in fewer function evaluations than other benchmark algorithms.

4. Discussion

This section consists of two sub-sections, with the advantages of OMA described in Section 4.1 and the performance of OMA as a robust new metaheuristic algorithm described in Section 4.2.

4.1. Advantages of OMA

Based on the findings of this study, the advantages of OMA include:

- The characteristics of OMA are similar to most of the best-known population-based algorithms (e.g., GA, PSO, DE, and ABC). Moreover, OMA performs specific operations on a group of candidate solutions iteratively to achieve a global solution.
- Unlike GA and other evolutionary algorithms, OMA does not reproduce new generations. Thus, global solutions are derived from the initial population.
- OMA, similar to FBI and SOS, uses only two parameters: population size and the maximum number of iterations. Algorithms such as DE, ABC, GSA, and BES require tuning at least one additional algorithm-specific parameter.
- The two magnification phases in the OMA use three equations to balance exploration and exploitation, which simplifies the understanding and use of this new algorithm. The objective lens phase performs the role of exploration, and the eyepiece phase performs the role of exploitation (local search). This effective balance helps OMA outperform other metaheuristic algorithms in solving numerical optimization problems.

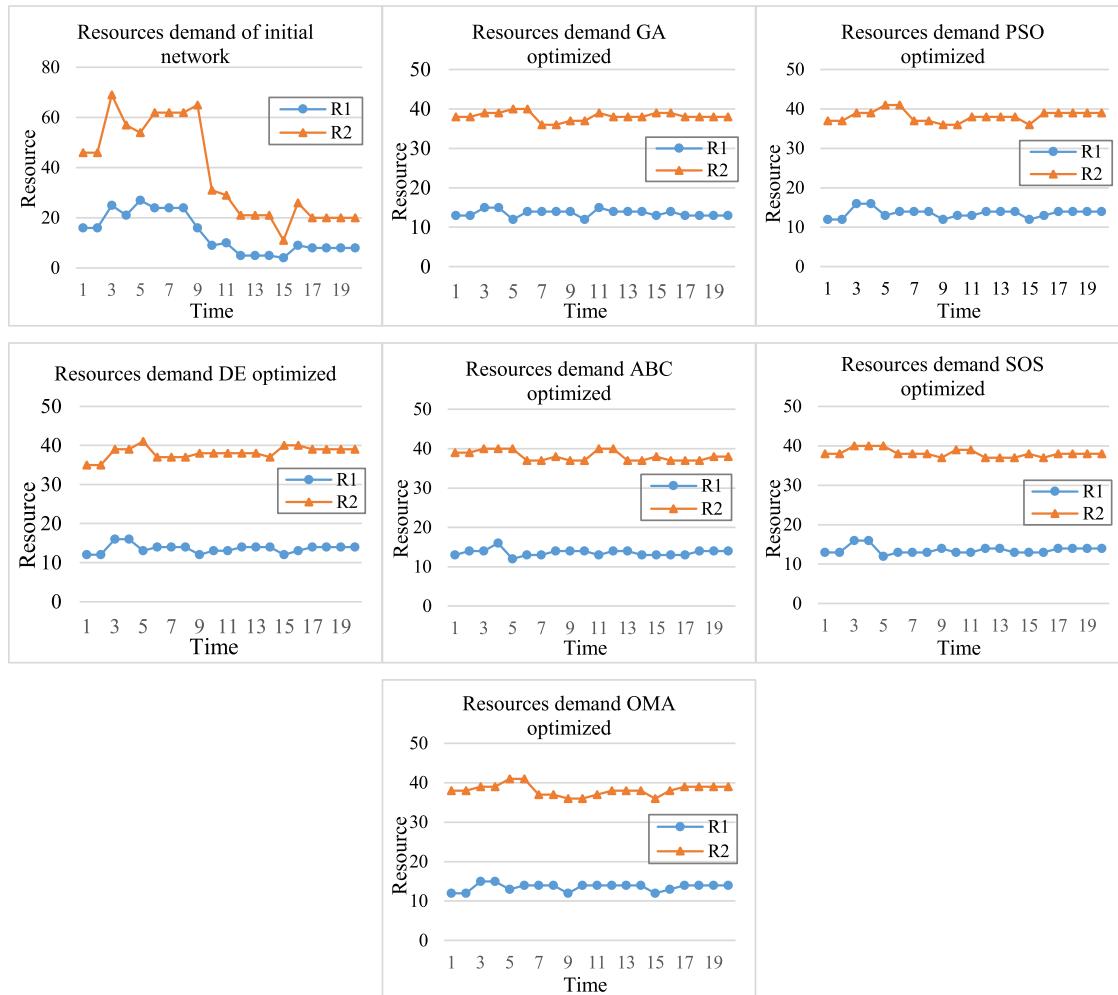


Fig. 16. Resource profile of projects by different algorithms.

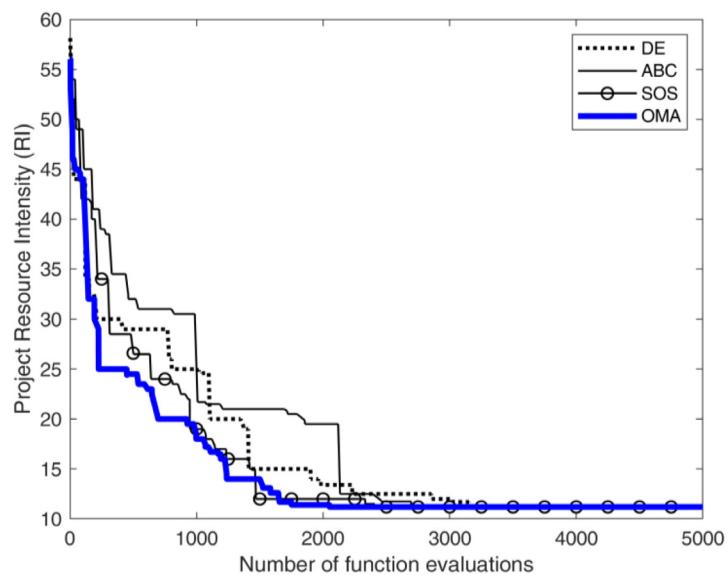


Fig. 17. Best project resource intensity curves for the compared algorithms.

Table 8
Comparison of algorithm optimal resource intensity.

Algorithm	RI	RI ₁	RI ₂	Actual start time of non-key activities: T _s								Rank
				A ₁	B ₁	C ₁	G ₁	H ₁	A ₂	C ₂	E ₂	
Initial [60]	71.700	60.040	366.430	0	0	4	0	8	2	5	5	7
GA [60]	16.210	10.040	40.130	1	0	7	5	8	17	11	13	6
PSO [60]	11.290	2.940	41.930	9	0	8	0	12	17	5	14	5
DE	11.244	4.912	40.255	1	0	8	6	8	17	11	13	4
ABC	11.235	4.005	40.165	1	0	7	5	8	17	11	13	3
SOS	11.212	4.023	40.021	1	0	6	6	7	17	11	13	2
OMA	11.211	4.022	40.022	1	0	6	6	8	16	11	13	1

- (e) OMA adapts from global to local search. While similar to FBI and BES in many respects, OMA uses a significantly different strategy that focuses on two search-detailing strategies, namely objective lens and eyepiece, to incrementally improve candidate solutions.
- (f) The objective lens is a strategy OMA uses that is similar to those used in PSO, FBI, BES, and SOS. The objective lens changes the solution by calculating the differences between other solutions. The OMA strategy differs slightly from the other algorithms as it uses the best solution as a reference point to refine the search for a promising target object near the best solution.
- (g) As a local search, the eyepiece focuses on the areas captured during the objective lens phases to search for more specific target objects. Although BES also utilizes this method in the “swoop” phase [24], OMA is more effective in finding global solutions because its boundaries are more specific in the local search due to magnification power.
- (h) OMA uses greedy selection at the end of each phase to determine whether to keep the old solution or modify it in the population. FBI and SOS also use this selection mechanism.
- (i) OMA uses a numerical aperture (NA) value of microscope magnification. Therefore, no further analysis is needed to determine a specific parameter by a trial and error. This method helps OMA reduce the convergence speed of the search process.

4.2. Performance evaluation

This study validated OMA on 50 (unimodal, multimodal, separable, and non-separable) mathematical benchmark functions. OMA's performance was compared against other metaheuristic algorithms, namely GA, PSO, GSA, SOS, BES, FBI, and JS. The last three algorithms are the latest metaheuristics, with the former two introduced in 2020 and the latter introduced in 2021. Therefore, besides being significantly more robust and easy to implement, OMA's performance is comparable to the latest algorithms, making OMA the best algorithm currently available. In addition, OMA was applied to construction engineering optimization problems.

Its good performance on the problems in this study supports the capability of OMA to generate solutions at a level of quality that is significantly better than that of other metaheuristic algorithms. OMA performed consistently better on all benchmark function and optimization problems than the other algorithms. In the previous section, the experiments in this study validated the performance of OMA on various problems, further confirming OMA as the best algorithm currently available.

The fast convergence of OMA allows it to determine optimal solutions faster than other algorithms, thus avoiding the danger of converging prematurely on an optimum before the stopping criteria. In this study, OMA clearly dominated the competition

in terms of efficiency and computational time for all benchmark functions and optimization problems.

The performance of OMA compared to the other considered algorithms is discussed in three aspects, as follows:

(a) OMA vs. JS and FBI

JS and FBI were the most recently introduced algorithm used for comparison in this study. The OMA, JS, and FBI performance results differed only slightly on the benchmark functions test, with all of them achieving the same minimum global count on benchmark functions. However, OMA outperformed JS and FBI in terms of the best performance on those functions. JS slightly outperformed OMA in computational time because of its use of trial and error to determine the constants of β and γ as specific algorithm parameters [25].

(b) OMA vs. BES and SOS

BES and SOS were the second-most recent algorithms compared in this study. The characteristics of OMA are the same as those of SOS, requiring only two parameters (i.e., NP and max_iter) and not requiring algorithm-specific parameters. BES is similar to OMA in identifying the best space to find the target object. OMA outperformed the two algorithms in terms of solving the 50 global optimum benchmark functions, solving one function more than SOS and five functions more than BES in the set of benchmark functions. Moreover, OMA's computation time was superior to all two of the algorithms at all function scales. In the case study, while OMA and SOS had the same performance on truss optimization, OMA was slightly superior to the SOS on project scheduling problem.

(c) OMA vs. GSA, ABC, DE, PSO, and GA

PSO, DE, and GA are the earliest algorithms introduced, making them the progenitors of nearly all algorithms developed and used today. GSA is the most popular physics-inspired algorithm (a category that includes OMA), and ABC is an efficient and popular algorithm. In terms of achieving the global optimum with benchmark functions, OMA was superior to GSA, ABC, DE, PSO, and GA, solving 29, 37, 32, 24, and 17 functions, respectively. Likewise, OMA outperformed ABC and DE in solving the project scheduling problem.

5. Conclusion

Inspired by the zoom-in (magnification) process, a new metaheuristic algorithm called the Optical Microscope Algorithm (OMA) was developed and tested in this study. With the exception of population size (NP) and the maximum number of iterations (max_iter), which are common to all metaheuristic algorithms, OMA does not require predefined parameters to achieve its best performance. This advantage enhances OMA's performance stability and facilitates its ready application to other optimization problems.

All phases in OMA zoom-in gradually in a phased search for the best target object. This algorithm treats the objective lens phase as exploration and the eyepiece phase as the local search (exploitation for magnification using microscope lenses). This exploration-exploitation balance improves consistency and helps OMA outperform other metaheuristic algorithms in solving optimization problems.

The performance of OMA was further validated in this study using several experiments, the results of which demonstrated the efficiency, durability, stability, and ease of operation of OMA. The results support that OMA is superior to the best-known and most recently introduced metaheuristic algorithms. The following reasons support OMA as a powerful and useful optimization tool: (1) Efficiency: OMA is highly efficient at solving various problems; (b) Speed: OMA solves problems using less computational time than most competitors; and (c) Power: The power of OMA to achieve an optimal solution is faster than most metaheuristic algorithms.

Although the proposed Optical Microscope Algorithm (OMA) exhibits several advantages, it is important to acknowledge its limitations. Two notable limitations of OMA include parameter sensitivity and real-world applicability. Firstly, despite not requiring predefined parameters, OMA still relies on a few common parameters, such as population size (NP) and the maximum number of iterations (*max_iter*). It is crucial to recognize that the performance of OMA can be sensitive to selecting these parameters. Future research should investigate effective strategies for parameter tuning in different problem domains. By addressing this parameter sensitivity, OMA's performance can be optimized, and its robustness across various problem types can be enhanced.

Secondly, while OMA has demonstrated promising results in solving optimization problems, its applicability to real-world scenarios and complex, dynamic systems requires further investigation. Evaluating OMA's behavior in environments that involve uncertainties, noisy data, or changing conditions is imperative. This evaluation will provide insights into the algorithm's practical feasibility and effectiveness in addressing real-world optimization challenges. Assessing OMA's performance under these conditions will help determine its suitability for real-world applications.

CRediT authorship contribution statement

Min-Yuan Cheng: Writing – review & editing, Validation, Supervision, Resources, Methodology, Investigation, Conceptualization. **Moh Nur Sholeh:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.knosys.2023.110939>.

References

- [1] K. Hussain, M.N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, *Artif. Intell. Rev.* 52 (2019) 2191–2233.
- [2] M. Braik, A. Hammouri, J. Atwan, M.A. Al-Betar, M.A. Awadallah, White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems, *Knowl.-Based Syst.* 243 (2022) 108457.
- [3] S. Gao, C.W. de Silva, Estimation distribution algorithms on constrained optimization problems, *Appl. Math. Comput.* 339 (2018) 323–345.
- [4] B.A. Hassan, T.A. Rashid, Operational framework for recent advances in backtracking search optimisation algorithm: A systematic review and performance evaluation, *Appl. Math. Comput.* 370 (2020) 124919.
- [5] P. Agrawal, H.F. Abutarboush, T. Ganesh, A.W. Mohamed, Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019), *IEEE Access* 9 (2021) 26766–26791.
- [6] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.
- [7] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Comput. Intell. Mag.* 1 (4) (2006) 28–39.
- [8] 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.
- [9] X.-L. Li, An optimizing method based on autonomous animats: Fish-swarm algorithm, *Syst. Eng.-Theory Pract.* 22 (11) (2002) 32–38.
- [10] J.H. Holland, Genetic algorithms, *Sci. Am.* 267 (1) (1992) 66–73.
- [11] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (4) (1997) 341.
- [12] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, 1996.
- [13] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, *IEEE Trans. Evol. Comput.* 3 (2) (1999) 82–102.
- [14] H. Abedinpourshotbaran, S.M. Shamsuddin, Z. Beheshti, D.N. Jawawi, Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm, *Swarm Evol. Comput.* 26 (2016) 8–22.
- [15] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: A gravitational search algorithm, *Inform. Sci.* 179 (13) (2009) 2232–2248.
- [16] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (4598) (1983) 671–680.
- [17] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Inform. Sci.* 222 (2013) 175–184.
- [18] F. Glover, Tabu search—part I, *ORSA J. Comput.* 1 (3) (1989) 190–206.
- [19] 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 (3) (2011) 303–315.
- [20] A.H. Kashan, League championship algorithm (LCA): An algorithm for global optimization inspired by sport championships, *Appl. Soft Comput.* 16 (2014) 171–200.
- [21] J.-S. Chou, N.-M. Nguyen, FBI inspired meta-optimization, *Appl. Soft Comput.* 93 (2020) 106339.
- [22] X.-S. Yang, Firefly algorithms for multimodal optimization, in: *Stochastic Algorithms: Foundations and Applications: 5th International Symposium SAGA*, Springer, 2009, pp. 169–178.
- [23] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: A new metaheuristic optimization algorithm, *Comput. Struct.* 139 (2014) 98–112.
- [24] H.A. Alsaltar, A. Zaidan, B. Zaidan, Novel meta-heuristic bald eagle search optimisation algorithm, *Artif. Intell. Rev.* 53 (2020) 2237–2264.
- [25] J.-S. Chou, D.-N. Truong, A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, *Appl. Math. Comput.* 389 (2021) 125535.
- [26] A.E. Eiben, S.K. Smit, Evolutionary algorithm parameters and methods to tune them, *Auton. Search* (2012) 15–36.
- [27] V.A. Tatsis, K.E. Parsopoulos, Dynamic parameter adaptation in metaheuristics using gradient approximation and line search, *Appl. Soft Comput.* 74 (2019) 368–384.
- [28] I. Ahmed, A. Basit, M. Rehan, K.-S. Hong, Multi-objective whale optimization approach for cost and emissions scheduling of thermal plants in energy hubs, *Energy Rep.* 8 (2022) 9158–9174.
- [29] I. Ahmed, M. Rehan, A. Basit, S.H. Malik, K.-S. Hong, Multi-area economic emission dispatch for large-scale multi-fueled power plants contemplating inter-connected grid tie-lines power flow limitations, *Energy* 261 (2022) 125178.
- [30] I. Ahmed, M. Rehan, A. Basit, M. Tufail, K.-S. Hong, A dynamic optimal scheduling strategy for multi-charging scenarios of plug-in-electric vehicles over a smart grid, *IEEE Access* 11 (2023) 28992–29008.
- [31] N. Bacanin, R. Stoean, M. Zivkovic, A. Petrovic, T.A. Rashid, T. Bezdan, Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization, *Mathematics* 9 (21) (2021) 2705.
- [32] S. Malakar, M. Ghosh, S. Bhownik, R. Sarkar, M. Nasipuri, A GA based hierarchical feature selection approach for handwritten word recognition, *Neural Comput. Appl.* 32 (2020) 2533–2552.

- [33] N. Bacanin, et al., Artificial neural networks hidden unit and weight connection optimization by quasi-refection-based learning artificial bee colony algorithm, *IEEE Access* 9 (2021) 169135–169155.
- [34] A. Trukhova, M. Pavlova, O. Sinityna, I. Yaminsky, Microlens-assisted microscopy for biology and medicine, *J. Biophotonics* 15 (9) (2022) e202200078.
- [35] A. Khodavardipour, M. Mehregan, A. Rajabi, Y. Shiri, Microscopy and its application in microbiology and medicine from light to quantum microscopy: A mini review, *Avicenna J. Clin. Microbiol. Infect.* 6 (4) (2019) 133–137.
- [36] B. Yan, et al., Superlensing microscope objective lens, *Appl. Opt.* 56 (11) (2017) 3142–3147.
- [37] S. Bradbury, *The Evolution of the Microscope*, Elsevier, 2014.
- [38] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: Harmony search, *Simulation* 76 (2) (2001) 60–68.
- [39] M. Mishra, P. Chauhan, Applications of microscopy in bacteriology, *Microscopy Res.* 4 (1) (2016) 1–9.
- [40] R.F. Egerton, *Physical Principles of Electron Microscopy*, Springer, 2005.
- [41] J.A. DeRose, M. Doppler, Guidelines for understanding magnification in the modern digital microscope era, *Microscopy Today* 26 (4) (2018) 20–33.
- [42] K.M. Sallam, S.M. Elsayed, R.A. Sarker, D.L. Essam, Landscape-assisted multi-operator differential evolution for solving constrained optimization problems, *Expert Syst. Appl.* 162 (2020) 113033.
- [43] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, *Appl. Math. Comput.* 214 (1) (2009) 108–132.
- [44] Q. Shang, et al., Multi-space evolutionary search with dynamic resource allocation strategy for large-scale optimization, *Neural Comput. Appl.* 34 (10) (2022) 7673–7689.
- [45] W. Cai, L. Yang, Y. Yu, Solution of ackley function based on particle swarm optimization algorithm, in: 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA, IEEE, 2020, pp. 563–566.
- [46] M. Ravber, S.-H. Liu, M. Mernik, M. Črepišek, Maximum number of generations as a stopping criterion considered harmful, *Appl. Soft Comput.* 128 (2022) 109478.
- [47] M. Mernik, S.-H. Liu, D. Karaboga, M. Črepišek, On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation, *Inform. Sci.* 291 (2015) 115–127.
- [48] W.-Y. Lee, S.-M. Park, K.-B. Sim, Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm, *Optik* 172 (2018) 359–367.
- [49] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* 1 (1) (2011) 3–18.
- [50] M.S. Kiran, O. Findik, A directed artificial bee colony algorithm, *Appl. Soft Comput.* 26 (2015) 454–462.
- [51] G. Zhang, Y. Li, A memetic algorithm for global optimization of multimodal nonseparable problems, *IEEE Trans. Cybern.* 46 (6) (2015) 1375–1387.
- [52] J. Luo, B. Shi, A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems, *Appl. Intell.* 49 (2019) 1982–2000.
- [53] Q. Askari, I. Younas, M. Saeed, Political optimizer: A novel socio-inspired meta-heuristic for global optimization, *Knowl.-Based Syst.* 195 (2020) 105709.
- [54] F. Yin, et al., Multifidelity genetic transfer: An efficient framework for production optimization, *SPE J.* 26 (04) (2021) 1614–1635.
- [55] W. Tang, L. Tong, Y. Gu, Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables, *Int. J. Numer. Methods Eng.* 62 (13) (2005) 1737–1762.
- [56] L.F.F. Miguel, R.H. Lopez, L.F.F. Miguel, Multimodal size, shape, and topology optimisation of truss structures using the firefly algorithm, *Adv. Eng. Softw.* 56 (2013) 23–37.
- [57] V. Ho-Huu, T. Nguyen-Thoi, M. Nguyen-Thoi, L. Le-Anh, An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures, *Expert Syst. Appl.* 42 (20) (2015) 7057–7069.
- [58] D. Prayogo, G. Gaby, B. Wijaya, F. Wong, Reliability-based design with size and shape optimization of truss structure using symbiotic organisms search, in: IOP Conference Series: Earth and Environmental Science, vol. 506, (no. 1) IOP Publishing, 2020, 012047.
- [59] M.-Y. Cheng, D. Prayogo, D.-H. Tran, Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search, *J. Comput. Civ. Eng.* 30 (3) (2016) 04015036.
- [60] Y. Guo, N. Li, T. Ye, Multiple resources leveling in multiple projects scheduling problem using particle swarm optimization, in: 2009 Fifth International Conference on Natural Computation, vol. 3, IEEE, 2009, pp. 260–264.
- [61] B. Gajdzik, W. Sroka, ResourceIntensity. vs, Investment in production installations—The case of the steel industry in Poland, *Energies* 14 (2) (2021) 443.