



An Advanced Swarm Intelligence-Based Algorithm for Engineering Problem Solving

Gholamreza Memarzadeh^{1*}, Farshid Keynia², Rasoul Memarzadeh³

^{1, 2}. Department of Energy Management and Optimization, Institute of Science and High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran.

³. Department of Civil Engineering, Faculty of Engineering, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran.

ARTICLE INFO

Article type:
Research Paper

Article History:
Received 4 January 2024
Accepted 7 September 2026

Keywords:
Optimization Algorithm,
Metaheuristic,
Coot Bird Algorithm,
Genetic Algorithm,
Economic Dispatch,
DG Placement.

ABSTRACT

Nowadays, global optimization algorithms are widely used for solving engineering problems. These algorithms are complex and characterized by various properties, for instance, they can be non-linear, non-separable, symmetric, asymmetrical, smooth with narrow ridges, unimodal, and multimodal, and can involve non-differentiable functions and high dimensionality. These properties create challenges for existing optimization algorithms, and the global optimum finding, is one of the long-standing goals in this area of study. In this paper, an efficient hybrid metaheuristic search algorithm based on swarm intelligent has been developed. In order to evaluate the performance of the presented algorithm, it has been implemented on 13 benchmark functions and two real engineering problems in the fields of power engineering. The proposed functions are among the constrained and unconstrained optimization problems. Nevertheless, the proposed method has optimized the desired problems well and has performed better than GA, PSO and CBSA methods, which are among the well-known optimization methods.

Cite this article: Memarzadeh, G., Keynia, f., Memarzadeh, R. (2026). An Advanced Swarm Intelligence-Based Algorithm for Engineering Problem Solving. *Energy Systems: Technology and Management*, 2 (1), 1-22. DOI: <http://doi.org/10.48306/estm.2026.434012.1000>



© Gholamreza Memarzadeh, Farshid Keynia, Rasoul Memarzadeh / **Publisher:** Graduate University of Advanced Technology.
DOI: <http://doi.org/10.48306/estm.2026.434012.1000>

* Corresponding author.
Email: reza.meamarzadeh@gmail.com

1. Introduction

Optimization, with its ability to adjust problem inputs to achieve optimal outputs or results, holds tremendous significance in the field of engineering. By utilizing various techniques and algorithms, the optimization process aims to identify the best possible solution to a given problem [1]. However, the task of selecting the optimum answer becomes exceptionally challenging when dealing with complex engineering problems, and the time required to solve these problems becomes a crucial factor in the optimization process. In essence, optimization seeks to identify the most favorable solution among the available options within an acceptable timeframe [2]. Optimization methods can be broadly classified into two main types: derivative-based optimization and optimization without the need for derivatives.

Derivative-based methods determine the search direction by utilizing information derived from the derivative of the objective function, ultimately leading to the attainment of an optimal solution. However, these methods often prove unsuitable for complex problems encountered in real-world engineering scenarios. The complexity arises from the involvement of a large number of decision variables, each exerting intricate effects on the objective function. In contrast, derivative-free methods do not rely on the derivative of the objective function for the search process. Instead, these methods evaluate the objective function at various points within the problem's search space. By utilizing this information along with heuristic and intuitive rules, they determine the appropriate search direction. Such methods are commonly referred to as metaheuristic algorithms, which have the capability to find near-optimal solutions within an acceptable timeframe, although they do not guarantee finding the global optimum. Nonetheless, simulation results for numerous optimization problems have consistently indicated that by appropriately configuring the algorithm's coefficients to address specific problem characteristics, it is often possible to achieve

the optimal solution within a reasonable time frame.

The field of metaheuristic optimization encompasses several well-known methods, each exhibiting unique characteristics and performance. These methods include the genetic algorithm (GA) [3], simulated annealing (SA) [4], particle swarm optimization (PSO) [5], harmony search (HS) [6], ant colony optimization (ACO) [7], tabu search algorithm (TSA) [8], bat algorithm (BA) [9], teaching-learning-based optimization (TLBO) [10], immune system algorithm (ISA) [11], imperialist competitive algorithm (ICA) [12], gravitational search algorithm (GSA) [13], coot bird search algorithm (CBSA) [14], crow search algorithm (CSA) [15], Lévy flight distribution (LFD) [16], biogeography-based optimizer (BBO) [17], Yin-Yang-Pair optimization (YYPO) [18], group search optimizer (GSO) [19], volleyball premier league algorithm based on sine cosine algorithm (VPLSCA) [20], The combination search algorithm (CoSA) is implemented with the help of two recently developed supply-demand-based optimization (SDO) and Harris hawks optimization (HHO) optimization algorithms [21], firefly algorithm (FA) [22], dandelion optimizer (DO) [23], barnacles mating optimizer (BMO) [24], Cuckoo catfish optimizer (CCO) [25], Gazelle optimization algorithm (GOA) [26], Dream Optimization Algorithm (DOA) [27], Spider wasp optimizer (SWO) [28], and Dhole optimization algorithm (DOA) [29]. These methods, being derivative-free, employ heuristic processes to explore and identify optimal solutions.

Metaheuristic algorithms are developed with three primary objectives in mind: achieving faster problem-solving capabilities, handling large-scale problems efficiently, and obtaining robust algorithms. Notable examples of successful metaheuristic algorithms in the realm of engineering optimization include the genetic algorithm (GA) and the coot bird search algorithm (CBSA). Each algorithm possesses its own strengths and weaknesses. Genetic algorithms excel at reaching global

regions but often struggle to find exact solutions. On the other hand, CBSA leverages group interactions to enhance the search for optimal solutions. Recognizing the advantages offered by both algorithms, the main objective of this paper is to develop an efficient hybrid technique called CBSA-GA, which aims to combine the strengths of CBSA and GA to find optimal solutions for engineering optimization problems. The proposed algorithm is implemented and evaluated on 13 benchmark functions and three real engineering problems in the power engineering domain, showcasing its effectiveness and applicability. The hybrid nature of CBSA-GA allows for a synergistic combination of the exploration capabilities of CBSA and the global search abilities of GA, thereby enhancing the overall optimization performance.

In Section 2 of this paper, we provide a detailed description of the proposed hybrid CBSA-GA algorithm. The algorithmic framework, including the integration of CBSA and GA components, is presented along with the step-by-step procedure for solving engineering optimization problems. We discuss how CBSA and GA complement each other in the hybrid algorithm, taking advantage of CBSA's group interactions and GA's global search strategy. The algorithm's parameters and settings are also elaborated upon, emphasizing their influence on the optimization process.

Section 3 focuses on the experimental evaluation of the CBSA-GA algorithm. We conduct extensive experiments on a set of 13 benchmark functions commonly used in optimization research. These functions encompass a range of complexities, allowing us to thoroughly assess the algorithm's performance. Additionally, we apply the CBSA-GA algorithm to three real engineering problems in the power engineering domain. These problems highlight the algorithm's applicability to practical scenarios and demonstrate its effectiveness in finding optimal solutions. The experimental results, including convergence analysis, solution quality evaluation, and computational

efficiency, are presented and discussed in detail.

In Section 4, we draw conclusions based on the findings of this study. We summarize the key contributions of the CBSA-GA algorithm in addressing engineering optimization problems. The strengths and limitations of the proposed approach are critically analyzed, providing insights into its performance and potential areas for further improvement. Additionally, we discuss the broader implications of this research and its relevance to the field of engineering optimization.

In conclusion, this paper introduces the concept of optimization in engineering and highlights the significance of selecting the best solution within a specified timeframe. We categorize optimization methods into derivative-based and derivative-free approaches, with a focus on metaheuristic algorithms as effective derivative-free methods. By leveraging the advantages of CBSA and GA, we propose the hybrid CBSA-GA algorithm as a powerful technique for engineering optimization problems. The algorithm's implementation and evaluation on benchmark functions and real engineering problems demonstrate its efficiency and effectiveness. The subsequent sections of the paper provide a detailed description of the algorithm, present experimental results, and draw conclusions. Overall, this study contributes to the advancement of optimization techniques in engineering and offers a valuable approach for finding optimal solutions in practical applications.

2. Problem description

In this section, we present the innovative hybrid CBSA-GA algorithm, which combines the cooperative search behavior of CBSA with the evolutionary principles of GA. The algorithm aims to address optimization problems by leveraging the strengths of both approaches, leading to enhanced performance and solution quality.

2.1. Genetic algorithm (GA)

The genetic algorithm, initially presented by John Holland in the 1960s and further developed by his colleagues at the University of Michigan, has since been extensively researched and applied in various fields. Its applications span a wide range, including optimization of numerical functions, image processing, machine learning, structural design and optimization, optimal placement of distributed generation resources, and many other areas.

In the genetic algorithm, a chromosome represents a coded solution to an optimization problem. It comprises a fixed number of genes that encode the decision variables of the problem. These chromosomes collectively form a population, from which a new population is generated using genetic operators. To tackle an optimization problem, a suitable fitness function is designed. This function assigns a non-negative value to each chromosome, reflecting its efficiency or capability to solve the problem. Essentially, the fitness function acts as the objective function within the genetic algorithm. During the reproduction stage, genetic operators come into play to create the next generation. The selection operator plays a pivotal role in this process by choosing a subset of chromosomes from the current population for reproduction and transferring them to the mating pool. While the selection of chromosomes is random, it is biased towards favoring more competent individuals, granting them a higher chance of reproduction. The crossover operator, another vital genetic operator, performs the crossover operation on two or more parent chromosomes, resulting in the creation of two offspring for the new generation. This operator combines genetic information from different parents, fostering exploration and diversification within the search space. Following the crossover operator, the mutation operator may be applied, randomly selecting a gene from a chromosome and introducing slight alterations to its contents. The mutation operator serves as a mechanism to introduce occasional random

changes and prevent premature convergence of the algorithm.

Fig. 1 depicts the pseudo code of the genetic algorithm, providing an outline of the sequential steps involved in its execution. This representation offers a high-level overview of the algorithm's structure and serves as a helpful reference for its implementation. To summarize, the genetic algorithm is a potent optimization technique that emulates the principles of natural selection and evolution. It leverages genetic operators such as selection, crossover, and mutation to iteratively refine the population of candidate solutions. Through successive generations, the genetic algorithm explores the search space and gradually converges towards optimal or near-optimal solutions.

```

Begin
t=0;
initialize population P(t);
evaluate P(t);
while (not termination condition) do
    parent-selection P(t);
    recombine P(t) to yield C(t);
    mutate C(t);
    evaluate C(t);
    survive (P(t), C(t)) to yield P(t+1);
    t=t+1;
end
end

```

Fig. 1. Pseudo code of the GA

2.2. Coot bird search algorithm (CBSA)

Coots, belonging to the rail family, are small water birds known scientifically as Fulica. The behavior of these birds on the water surface serves as inspiration for a novel optimization method known as the Coot Bird Search Algorithm (CBSA). Coots exhibit various movements when navigating the water, including irregular movement, coordinated movement, and chain motion [30]. The CBSA algorithm aims to simulate these diverse behaviors observed in coots, particularly their regular and irregular movements on the water surface.

The CBSA algorithm incorporates four distinct movements to explore the water surface during the optimization process: random movement, chain movement, position

adjustment based on group leaders, and group guidance towards the optimal area. The algorithm follows a systematic process to find optimal solutions. Initially, a random population is generated, representing the positions of individual coots. The objective function of the optimization problem is then evaluated for each coot position. Additionally, the number of leaders and the number of coots are randomly determined. The best coot or leader is identified as the optimal solution.

During each iteration of the algorithm, the positions of the coots on the water surface are updated based on the different movement strategies. Random movement introduces exploration by allowing coots to move randomly within the search space. Chain movement mimics the sequential motion of coots on the water, promoting coordinated exploration. Position adjustment based on group leaders ensures that coots adjust their positions based on the guidance of influential leaders within their vicinity. Finally, leader

movement involves leaders guiding the entire group towards the optimal area, exploiting the collective intelligence of the swarm. Fig. 2 presents the pseudo code of the CBSA algorithm, outlining the sequential steps involved in its execution. This pseudo code serves as a reference and implementation guide for applying the algorithm to optimization problems.

In summary, the CBSA algorithm draws inspiration from the movements of coots on the water surface to develop an innovative optimization technique. By emulating the irregular, coordinated, and chain motions observed in coots, the algorithm explores the search space in a manner that balances exploration and exploitation. Through the integration of random movement, chain movement, position adjustment based on group leaders, and leader guidance, the CBSA algorithm iteratively refines the positions of coots on the water surface, ultimately converging towards optimal solutions.

```

Begin
t=0;
initialize population P(t);
random selection of leaders;
evaluate P(t);
find the global optimum;
while (not termination condition) do
    update the position of coots based on random movement, chain movement and position adjustment based
    on group leaders;
    calculate the fitness of coots;
    if the fitness of coots < the fitness of leaders
        replace the position of coots by leaders;
    end
    update the position of leaders;
    calculate the fitness of leaders;
    if the fitness of leaders < global optimum
        update global optimum;
    end
    t=t+1;
end
end

```

Fig. 2. Pseudo code of the CBSA

2.3. Hybrid CBSA-GA algorithm

This research paper introduces a novel hybrid optimization method known as the CBSA-GA algorithm, which combines the advantageous features of both the Genetic Algorithm (GA) and the Coot Bird Search Algorithm (CBSA). By integrating these two

powerful algorithms, we aim to overcome the limitations of individual approaches and provide an enhanced optimization solution that combines the benefits of both. The CBSA-GA algorithm offers several advantages in solving optimization problems. By incorporating elements from the GA and CBSA algorithms,

it promotes diversity within generations and maintains a balance between local exploitation and global exploration. This allows the algorithm to efficiently explore the search space, avoid premature convergence, and ultimately converge towards near-optimal solutions.

Fig. 3 depicts the flowchart of the hybrid CBSA-GA algorithm, illustrating its sequential steps. Initially, the CBSA algorithm is employed to generate the initial population. The CBSA algorithm, inspired by the movement behavior of coots on water surfaces, provides a unique perspective for exploring optimization landscapes. It generates a diverse set of individuals representing potential solutions to the problem at hand.

Once the initial population is generated, the algorithm proceeds to determine the COOT's followers and leaders. Each individual's position is evaluated by calculating the objective function specific to the optimization problem being solved. This objective function serves as a measure of the individual's performance and guides the search towards better solutions. Next, the algorithm updates the positions of the COOT's followers and leaders. This update process is based on the algorithm's structure, as explained in [14]. It incorporates various movements on the water surface, such as random movement, chain movement, position adjustment based on group leaders, and guidance towards the optimal area. These movements emulate the behaviors observed in coots and facilitate an

exploration-exploitation balance during the optimization process. After a predetermined number of algorithm iterations, the updated positions are handed over to the GA algorithm. The GA algorithm, renowned for its ability to efficiently search for optimal solutions, further refines the positions of the COOTs by applying selection, crossover, and mutation operators. The selection operator favors individuals with higher fitness, ensuring the preservation of good solutions. The crossover operator combines genetic information from different COOTs, promoting exploration of the search space. Lastly, the mutation operator introduces occasional random changes to maintain diversity and prevent stagnation. The hybrid CBSA-GA algorithm repeats this iterative process, alternating between the CBSA and GA phases, until a termination condition is met, typically a maximum number of iterations. At the end of the algorithm, the final position obtained represents a near-optimal solution for the given optimization problem.

The CBSA-GA algorithm offers numerous advantages in solving optimization problems across various domains. Its ability to combine the exploration capabilities of CBSA with the refinement capabilities of GA allows for efficient and effective optimization. The algorithm has been successfully applied in diverse fields, including engineering, finance, and image processing, demonstrating its versatility and robustness. Fig. 4 presents the pseudo code of the CBSA-GA algorithm, outlining the sequential steps involved in its execution.

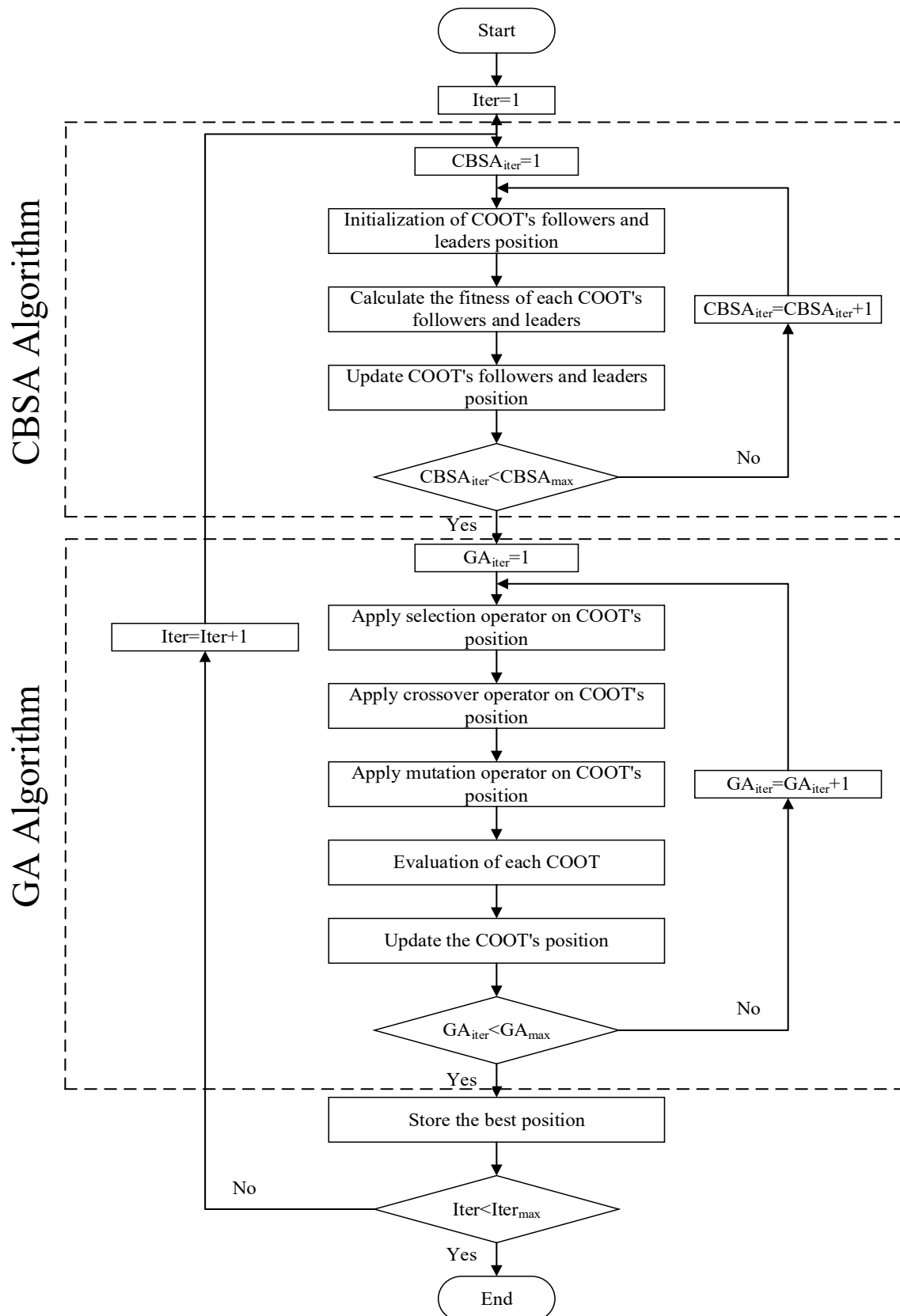


Fig. 3. The flowchart of the hybrid CBSA-GA algorithm

```

Begin
  Iter = 1;
  while (Iter < Iter_max) do

    % CBSA Algorithm Section
    CBSA_iter = 1;
    while (CBSA_iter < CBSA_max) do
      Initialization of COOT's followers and leaders position;
      Calculate the fitness of each COOT's followers and leaders;
      Update COOT's followers and leaders position;
      CBSA_iter = CBSA_iter + 1;
    end

    % GA Algorithm Section
    GA_iter = 1;
    while (GA_iter < GA_max) do
      Apply selection operator on COOT's position;
      Apply crossover operator on COOT's position;
      Apply mutation operator on COOT's position;
      Evaluation of each COOT;
      Update the COOT's position;
      GA_iter = GA_iter + 1;
    end

    Store the best position;
    Iter = Iter + 1;
  end
end

```

Fig. 4. Pseudo code of the CBSA-GA

3. Simulation and discussion of results

In this section, we present the simulation results of the CBSA-GA algorithm, which is applied to solve various optimization problems. The effectiveness of the algorithm is evaluated using 13 benchmark functions with dimensions of both 30 and 100, allowing for a comprehensive analysis of its performance. Additionally, we investigate the algorithm's performance on two real engineering optimization problems to assess its applicability in practical scenarios.

To evaluate the efficiency of the CBSA-GA algorithm, we compare its results with those obtained from other well-known optimization algorithms such as GA, PSO, and CBSA. This comparative analysis provides insights into the algorithm's efficiency, convergence speed, and solution quality when compared to state-of-the-art approaches. By assessing the CBSA-GA algorithm against multiple algorithms, we gain a comprehensive understanding of its strengths and weaknesses, facilitating

informed decision-making for different problem types.

Table 1 presents detailed information about the parameters and specifications of the CBSA-GA algorithm. These parameters are carefully selected to ensure the algorithm's effectiveness in exploring and exploiting the search space. The population size, maximum number of iterations, selection criteria, crossover and mutation rates, and other relevant settings are optimized to enhance the algorithm's performance across a wide range of optimization problem domains.

The benchmark functions utilized in this study encompass various optimization landscapes, including unimodal, multimodal, convex, and non-convex functions. Evaluating the CBSA-GA algorithm on these benchmark functions allows us to assess its capability to handle different types of optimization problems. Furthermore, by considering both 30 and 100 dimensions, we conduct a comprehensive analysis of the algorithm's

scalability and performance in higher-dimensional spaces.

In addition to the benchmark functions, we incorporate two real engineering optimization problems in our evaluation. These problems are carefully chosen to reflect practical scenarios and challenges encountered in engineering applications. By applying the CBSA-GA algorithm to these real-world problems, we aim to demonstrate its applicability and effectiveness in solving complex optimization challenges commonly encountered in engineering domains.

To quantify the performance of the CBSA-GA algorithm, we analyze its convergence behavior, solution quality, and computational efficiency. Statistical analyses, such as t-tests or ANOVA, may be conducted to determine the significance of observed differences between the CBSA-GA algorithm and the compared algorithms. These analyses serve to validate the algorithm's performance and provide a solid foundation for drawing meaningful conclusions based on the simulation results.

Overall, the simulation results presented in this section offer a comprehensive evaluation of the CBSA-GA algorithm's performance in solving optimization problems. By considering benchmark functions, real engineering problems, and comparing the algorithm with

other approaches, we provide a thorough analysis of its capabilities. These findings contribute to a deeper understanding of the algorithm's strengths and limitations, assisting researchers and practitioners in selecting appropriate optimization techniques for their specific problem domains.

Table 1. The parameters and specifications of the proposed algorithms

Algorithm	Parameter	Value
GA	Crossover	0.8
	Mutation	0.3
PSO	Cognitive and social constants	$C_1=2, C_2=2$
	Inertial weight	Linearly decreases from 0.9 to 0.4
CBSA	Number of leaders	10% of the initial population
	Crossover	0.9
	Mutation	0.005
CBSA-GA	Maximum iteration of CBSA-GA	50
	Maximum iteration of GA	500
	Maximum iteration of PSO	100

3.1. Benchmark functions

This section focuses on the evaluation of the CBSA-GA algorithm on 13 benchmark functions, considering both 30 and 100 dimensions. The detailed characteristics of these functions can be found in

Table 2.

Table 2. Test functions, their search range and the optimal value.

Function	Dimensions	Range	Optimal value
$f_1(x) = \sum_{i=1}^n x_i^2$	[30,100,500]	[-100,100]	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[30,100,500]	[10,10]	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	[30,100,500]	[-100,100]	0
$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	[30,100,500]	[-100,100]	0
$f_5(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[30,100,500]	[-30,30]	0

$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	[30,100,500]	[-100,100]	0
$f_7(x) = \max\{ x_i , 1 \leq i \leq n\}$	[30,100,500]	$[-1.28, 1.28]$	0
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	[30,100,500]	[-500,500]	0
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[30,100,500]	$[-5.12, 5.12]$	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$	[30,100,500]	[-32,32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[30,100,500]	[-600,600]	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[30,100,500]	[-50,50]	0
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	[30,100,500]	[-50,50]	0

To assess the performance of the CBSA-GA algorithm, simulations were conducted on

Table 3. The outcomes unequivocally demonstrate that, across all 13 functions, the CBSA-GA algorithm consistently outperformed other methods in terms of

the 13 objective functions with 30 dimensions, and the corresponding results are presented in finding the best solution. Moreover, it exhibited the remarkable ability to converge to the global optimal value in functions f1, f9, and f11. These notable achievements illustrate the algorithm's exceptional exploration capability throughout the entire search space.

Table 3. Result of benchmark functions with 30 dimensions (50 runs).

Function	Index	GA [14]	PSO [14]	CBSA [14]	CBSA-GA
f_i	Best	1.5245	9.3905×10^{-8}	2.8240×10^{-47}	0
	Worst	7.4966	5.0237×10^{-5}	4.3670×10^{-23}	1.3731×10^{-23}

f_2	Std	3.6872	2.8706×10^{-6}	1.4570×10^{-24}	1.9419×10^{-24}
	Avg	1.3063	9.0693×10^{-6}	7.9728×10^{-24}	2.7463×10^{-25}
	Best	2.2110×10^{-1}	1.1259×10^{-4}	4.0984×10^{-24}	2.3018×10^{-284}
	Worst	6.5960×10^{-1}	1.0800×10^{-2}	9.0834×10^{-13}	5.1375×10^{-9}
	Std	4.7510×10^{-1}	1.8000×10^{-3}	5.6782×10^{-14}	9.7651×10^{-10}
f_3	Avg	1.0590×10^{-1}	2.4000×10^{-3}	1.8801×10^{-13}	2.3886×10^{-10}
	Best	2.8924×10^3	3.1124×10^1	9.1935×10^{-48}	8.1531×10^{-280}
	Worst	1.1739×10^4	4.5637×10^2	8.5873×10^{-22}	9.2284×10^{-21}
	Std	5.1669×10^3	1.5501×10^2	2.8626×10^{-23}	1.3147×10^{-21}
	Avg	2.1462×10^3	1.0194×10^2	1.5678×10^{-22}	2.1112×10^{-22}
f_4	Best	4.9927	1.0955	9.1191×10^{-25}	3.7150×10^{-230}
	Worst	2.1162×10^1	6.1639	3.0541×10^{-12}	4.6303×10^{-10}
	Std	9.1426	2.5264	1.4861×10^{-13}	6.5482×10^{-11}
	Avg	3.0510	1.0886	6.0000×10^{-13}	9.2608×10^{-12}
	Best	1.7212×10^2	1.6161×10^1	2.8344×10^1	22.3856
f_5	Worst	9.0158×10^2	1.0518×10^2	6.6751×10^1	26.3867
	Std	4.1400×10^2	3.7766×10^1	3.1804×10^1	0.8812
	Avg	1.7821×10^2	2.4842×10^1	7.8738	25.6459
	Best	1.5385	3.8856×10^{-8}	4.7900×10^{-2}	3.0786×10^{-9}
	Worst	7.5916	9.8496×10^{-6}	5.4550×10^{-1}	1.1747×10^{-7}
f_6	Std	3.6049	1.5514×10^{-6}	1.9360×10^{-1}	2.1312×10^{-8}
	Avg	1.5701	2.4394×10^{-6}	1.2700×10^{-1}	3.0607×10^{-8}
	Best	5.5600×10^{-2}	1.0200×10^{-2}	3.0816×10^{-4}	2.2934×10^4
	Worst	2.0180×10^{-1}	5.6100×10^{-2}	2.2100×10^{-2}	0.0087
	Std	1.2240×10^{-1}	2.5300×10^{-2}	4.8000×10^{-3}	0.0017
f_7	Avg	3.9000×10^{-2}	1.0300×10^{-2}	4.4000×10^{-3}	0.0018
	Best	-1.1491×10^4	-7.8895×10^3	-9.2190×10^3	-1.2214×10^4
	Worst	-1.0316×10^4	-4.8503×10^3	-6.3040×10^3	-8.8240×10^3
	Std	-1.0872×10^4	-6.5312×10^3	-7.5838×10^3	592.4578
	Avg	3.0684×10^2	8.0228×10^2	7.3974×10^2	-1.1154×10^4
f_8	Best	2.8791	9.5516×10^1	0.0000	0
	Worst	1.4370×10^1	9.5516×10^1	5.0591×10^{-12}	4.5475×10^{-13}
	Std	7.7912	4.8255×10^1	1.8948×10^{-13}	9.4114×10^{-14}
	Avg	2.8269	1.6229×10^1	9.2219×10^{-13}	3.1832×10^{-14}
	Best	3.2410×10^{-1}	4.6059×10^{-5}	8.8818×10^{-16}	4.4409×10^{-16}
f_{10}	Worst	1.6306	2.8136	1.9263×10^{-8}	8.5754×10^{-10}
	Std	7.3590×10^{-1}	1.2718	6.4475×10^{-10}	1.2429×10^{-10}
	Avg	3.1770×10^{-1}	8.2150×10^{-1}	3.5164×10^{-9}	2.4023×10^{-11}
	Best	9.1080×10^{-1}	3.8588×10^{-8}	0.0000	0
	Worst	1.0681	7.0900×10^{-2}	6.1950×10^{-14}	2.2871×10^{-14}
f_{11}	Std	1.0221	1.4800×10^{-2}	2.1057×10^{-15}	3.6179×10^{-15}
	Avg	3.0900×10^{-2}	1.6700×10^{-2}	1.1305×10^{-14}	7.5051×10^{-16}
	Best	2.9000×10^{-3}	3.0857×10^{-9}	1.2000×10^{-3}	1.8602×10^{-11}
	Worst	2.4180×10^{-1}	9.3380×10^{-1}	1.3627	5.8585×10^{-10}
	Std	3.1500×10^{-2}	1.7650×10^{-1}	9.7700×10^{-2}	1.1733×10^{-10}
f_{12}	Avg	4.5400×10^{-2}	2.4850×10^{-1}	2.8540×10^{-1}	1.8237×10^{-10}
	Best	7.6300×10^{-2}	3.4979×10^{-7}	7.0200×10^{-2}	3.4171×10^{-10}
	Worst	6.1460×10^{-1}	6.2250×10^{-1}	2.9777	1.5146×10^{-08}
	Std	3.0580×10^{-1}	3.4600×10^{-2}	5.4750×10^{-1}	3.4787×10^{-09}

Avg	1.3670×10^{-1}	1.1420×10^{-1}	5.7420×10^{-1}	3.7298×10^{-09}
-----	-------------------------	-------------------------	-------------------------	--------------------------

Subsequently, the proposed algorithm was further tested on the same set of 13 benchmark functions, but with an increased

dimensionality of 100. The results of this implementation are showcased in

Table 4. Notably, the findings indicate that the algorithm's performance remains unaffected by the higher dimensions of the problem, demonstrating its stability across all test functions. Encouragingly, similar to the 30-dimensional case, the algorithm successfully converged to the global optimal value in functions f_9 and f_{11} . These results reaffirm the algorithm's remarkable exploration ability, highlighting its effectiveness across diverse search spaces.

The consistent performance of the CBSA-GA algorithm across both 30 and 100 dimensions indicates its robustness and reliability. Its ability to adapt and maintain high-quality solutions despite an increase in problem complexity is a significant advantage. This finding suggests that the algorithm can be applied to real-world problems with higher dimensions, making it a valuable tool in various domains where high-dimensional optimization is required.

Table 4. Result of benchmark functions with 100 dimensions (50 runs).

Function	Index	GA [14]	PSO [14]	CBSA [14]	CBSA-GA
f_1	Best	5.3853×10^2	6.9653×10^1	3.0297×10^{-53}	3.3955×10^{-193}
	Worst	1.1458×10^3	4.1826×10^2	6.8693×10^{-16}	1.0562×10^{-23}
	Avg	8.1352×10^2	1.7389×10^2	2.2900×10^{-17}	2.9921×10^{-25}
	Std	1.5153×10^2	7.7260×10^1	1.2542×10^{-16}	1.5897×10^{-24}
f_2	Best	1.3601×10^1	6.4414×10^0	2.8292×10^{-24}	9.7189×10^{-191}
	Worst	1.9811×10^1	2.1111×10^1	1.8238×10^{-9}	4.5936×10^{-8}
	Avg	1.6708×10^1	1.0261×10^1	6.1960×10^{-11}	9.1880×10^{-10}
	Std	1.6276×10^0	3.2575×10^0	3.3281×10^{-10}	6.4963×10^{-9}
f_3	Best	2.3725×10^4	1.5460×10^4	5.4708×10^{-50}	1.7239×10^{-178}
	Worst	6.6340×10^4	4.5354×10^4	1.4417×10^{-17}	9.4260×10^{-21}
	Avg	4.4276×10^4	2.9170×10^4	4.8057×10^{-19}	2.1678×10^{-22}
	Std	1.1133×10^4	7.8458×10^3	2.6321×10^{-18}	1.3438×10^{-21}
f_4	Best	2.2930×10^1	1.9029×10^1	6.0397×10^{-25}	1.4039×10^{-107}
	Worst	4.4164×10^1	3.3467×10^1	3.9719×10^{-8}	2.4806×10^{-11}
	Avg	3.1569×10^1	2.3898×10^1	1.3252×10^{-9}	4.9675×10^{-13}
	Std	4.9567×10^0	3.0609×10^0	7.2514×10^{-9}	3.5079×10^{-12}
f_5	Best	2.4757×10^4	4.2886×10^3	9.8321×10^1	94.9172
	Worst	9.2137×10^4	4.5207×10^4	1.6813×10^3	202.5690
	Avg	5.1870×10^4	1.2503×10^4	1.6269×10^2	101.7183
	Std	1.6860×10^4	1.0208×10^4	2.9090×10^2	18.2356
f_6	Best	5.5967×10^2	5.4796×10^1	6.9538×10^0	3.6327×10^{-5}
	Worst	1.0809×10^3	6.7712×10^2	1.0185×10^2	1.9342×10^{-4}
	Avg	7.8926×10^2	2.0617×10^2	1.6013×10^1	8.6798×10^{-5}
	Std	1.3942×10^2	1.3756×10^2	2.0123×10^1	3.6100×10^{-5}
f_7	Best	1.0732×10^0	3.1330×10^{-1}	4.4694×10^{-4}	8.8372×10^{-5}
	Worst	2.0674×10^0	1.2000×10^0	3.9800×10^{-2}	0.0173

f_8	Avg	1.5439×10^0	5.0590×10^{-1}	7.6000×10^{-3}	0.0040
	Std	1.5439×10^0	1.9450×10^{-1}	7.4000×10^{-3}	0.0038
	Best	-2.9413×10^4	-2.2982×10^4	-2.5166×10^4	-3.3367×10^4
	Worst	-2.3490×10^4	-1.5181×10^4	-1.4408×10^4	-2.5619×10^4
	Avg	-2.6667×10^4	-1.9454×10^4	-1.8551×10^4	-3.0228×10^4
f_9	Std	1.4553×10^3	1.9668×10^3	3.1119×10^3	2.0010×10^3
	Best	1.1965×10^2	1.4270×10^2	0.0000×10^0	0
	Worst	2.0523×10^2	2.7619×10^2	1.0800×10^{-11}	2.6148×10^{-12}
	Avg	1.6581×10^2	1.8986×10^2	7.6928×10^{-13}	1.1141×10^{-13}
	Std	2.0264×10^1	3.3360×10^1	2.2776×10^{-12}	4.3730×10^{-13}
f_{10}	Best	4.3875×10^0	4.1052×10^0	8.8818×10^{-16}	4.4409×10^{-16}
	Worst	5.5574×10^0	7.0695×10^0	1.4984×10^{-7}	4.9674×10^{-11}
	Avg	4.9694×10^0	5.4362×10^0	4.9956×10^{-9}	1.3645×10^{-12}
	Std	3.0500×10^{-1}	7.6760×10^{-1}	2.7357×10^{-8}	7.1312×10^{-12}
	Best	6.1492×10^0	1.5479×10^0	0.0000×10^0	0
f_{11}	Worst	1.0079×10^1	9.9589×10^0	6.2061×10^{-14}	7.6383×10^{-14}
	Avg	8.3353×10^0	2.9901×10^0	2.1427×10^{-15}	1.7741×10^{-15}
	Std	9.5600×10^{-1}	1.5860×10^0	1.1320×10^{-14}	1.0861×10^{-14}
	Best	3.0565×10^0	3.9423×10^0	7.9800×10^{-2}	5.7522×10^{-8}
	Worst	8.6318×10^0	2.2947×10^1	8.1560×10^{-1}	6.6587×10^{-6}
f_{12}	Avg	5.4697×10^0	8.7237×10^0	2.1890×10^{-1}	4.2548×10^{-7}
	Std	1.4374×10^0	3.4332×10^0	1.3430×10^{-1}	1.0904×10^{-6}
	Best	7.7039×10^1	7.0222×10^1	8.8830×10^0	4.8954×10^{-6}
	Worst	2.1652×10^3	2.0390×10^3	3.2611×10^1	0.0124
	Avg	2.3802×10^2	1.9408×10^2	1.3338×10^1	0.0016
f_{13}	Std	4.2503×10^2	3.5310×10^2	6.2886×10^0	0.0039

Overall, the evaluation results validate the efficacy of the CBSA-GA algorithm, showcasing its superior performance in finding optimal solutions for the benchmark functions considered. The algorithm's remarkable exploration capabilities, demonstrated by its convergence to global optima, further solidify its status as a promising approach for solving optimization problems in diverse application domains.

3.2. Engineering optimization problems

This section aims to assess the effectiveness of the proposed algorithm by applying it to two practical optimization problems in the field of power engineering. Specifically, the problems considered are economic dispatch and the determination of the optimal location and size of distributed generation (DG) resources in the distribution system. The details of these problems, including the objective function, the

number of decision variables, and the governing constraints, are provided below.

The economic dispatch problem involves optimizing the allocation of power generation among various generating units to meet the electricity demand while minimizing the total cost. The objective function is typically formulated to minimize the fuel cost, taking into account the operational constraints of the generating units. The decision variables include the power output levels of each unit. The constraints may include power balance equations, generation limits, ramp rate limits, and other operational limits imposed by the power system.

The second problem focuses on determining the optimal location and size of DG resources in the distribution system. The objective is to maximize system performance by strategically placing DG units at suitable

locations and optimizing their capacities. The decision variables include the locations of the DG units and their respective sizes. The constraints may involve technical limitations such as voltage limits, power flow constraints, and capacity limits.

To evaluate the efficiency of the proposed algorithm, a comparative analysis has been performed with other popular optimization algorithms including GA, PSO, PSO-GA, and CBSA. The results obtained using the proposed algorithm have been compared against the results obtained by these alternative algorithms. By conducting this comprehensive evaluation, we can gain insights into the performance and superiority of the proposed algorithm in solving these power engineering optimization problems.

By assessing the proposed algorithm against other established methods, we can determine its effectiveness, robustness, and potential advantages in addressing these real-world power engineering challenges. This comparative analysis serves as a valuable benchmark for evaluating the proposed algorithm's performance and its applicability in practical scenarios.

3.2.1. Economic dispatch

In the problem of economic dispatch, the purpose is to solve the problem of optimal operation of three power plant units as described in equation (1):

$$\begin{aligned} C_1 &= 0.0015P_1^2 + 8P_1 + 300, & 50 \\ &\leq P_1 \leq 400 \\ C_2 &= 0.0005P_2^2 + 8P_2 + 450, & 50 \\ &\leq P_2 \leq 800 \\ C_3 &= 0.001P_3^2 + 7.5P_3 + 700, & 50 \\ &\leq P_3 \leq 1000 \end{aligned} \quad (1)$$

Table 5.

Table 6 indicates the performance of the CBSA-GA on this problem. This table shows

In other words, the purpose is to determine the optimal production power of each of the power plants in such a way that the objective function, which is the total cost of production defined by equation 2, is minimized, and on the other hand, the total demand (P_L) is equal to the total production of all generators.

$$\text{Min. Cost} = C_1 + C_2 + C_3$$

S. t.

$$\sum_{i=1}^3 P_i = P_L \quad (2)$$

The important point in solving this optimization problem is the entering the load equality condition in the objective function. To do this, the difference between the total demand and the generated values must be multiplied by a specific number (λ) and that value must be added to the objective function. In other words, the objective function of this problem is rewritten as follows:

$$\begin{aligned} \text{fit} &= C_1 + C_2 + C_3 \\ &+ \lambda \times \left(P_L - \sum_{i=1}^3 P_i \right) \end{aligned} \quad (3)$$

In this problem, the value of λ is considered equal to 10. Now, for three different states that have different total demand, the problem of optimal operation of these three power plant units must be solved. The simulation results of this problem are given in

Table 6 indicates the performance of the CBSA-GA on this problem. This table shows the optimal values of the decision variables and the constraint values corresponding to the best solution obtained by CBSA-GA over 50 independent runs.

the optimal values of the decision variables and the constraint values corresponding to the best solution obtained by CBSA-GA over 50 independent runs.

Table 5. Result of economic dispatch problem with 100 dimensions (50 runs).

P_L	Algorithm	Best	Worst	Avg.	Std.
240	GA	3328.6322	3357.1371	3339.1898	6.0167

1200	PSO	3324.6230	6123.6614	3915.5006	875.4737
	CBSA	3324.6000	3324.6095	3324.6002	0.0014
	CBSA-GA	3324.6000	3324.6000	3324.6000	4.6120e-09
	GA	11238.0790	11248.2036	11234.5309	2.1792
	PSO	11233.6364	11233.6364	11233.6364	3.1506
	CBSA	11233.6364	11233.8138	11233.6566	0.0352
	CBSA-GA	11233.6364	11233.6364	11233.6364	2.6870e-07
	GA	18249.1514	18282.3188	18258.2140	8.7673
2000	PSO	18249.0338	18322.2682	18258.1523	15.4142
	CBSA	18249.0000	18249.0227	18249.0009	0.0033
	CBSA-GA	18249.0000	18249.0000	18249.0000	6.3662e-08

Table 6. The best solution obtained by CBSA for economic dispatch problem

P_D	P_1	P_2	P_3	Fit
240	50	50	140	3324.6000
1200	172.7370	518.1654	509.0975	11233.6364
2000	380.0024	800	819.9976	18249.0000

As it is clear from the results, the solution of the above problem using the CBSA-GA method has less deviations. In other words, the probability of converging the problem to a near-optimal solution using the CBSA-GA method is much higher than other methods.

The convergence curve of all expressed optimization algorithms for finding the

economic dispatch of generator for different load is shown in the Fig. 5 to Fig. 7. It can be seen that the convergence rate of CBSA-GA is good. As can be seen, the CBSA-GA has undergone fewer iteration to reach its near-global optimum. In other words, it converges faster to the near-global optimum point.

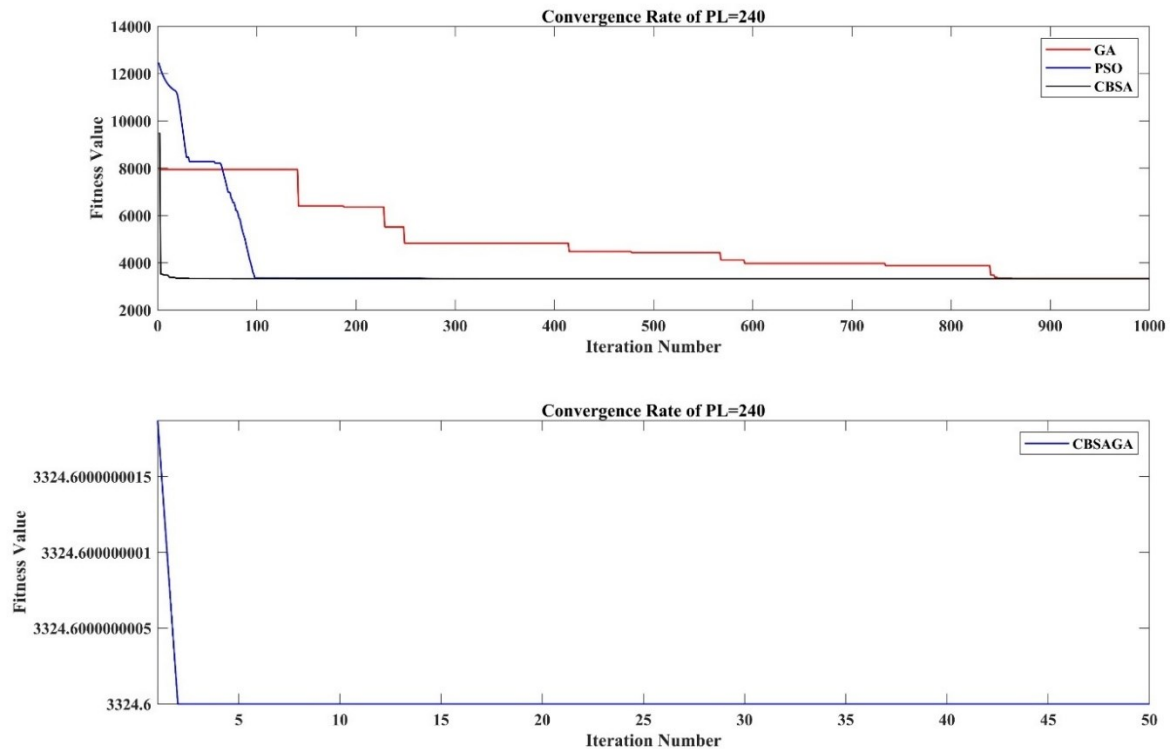


Fig. 5. Convergence rate of all optimization method for finding the best solution of economic dispatch problem for $P_L=240$.

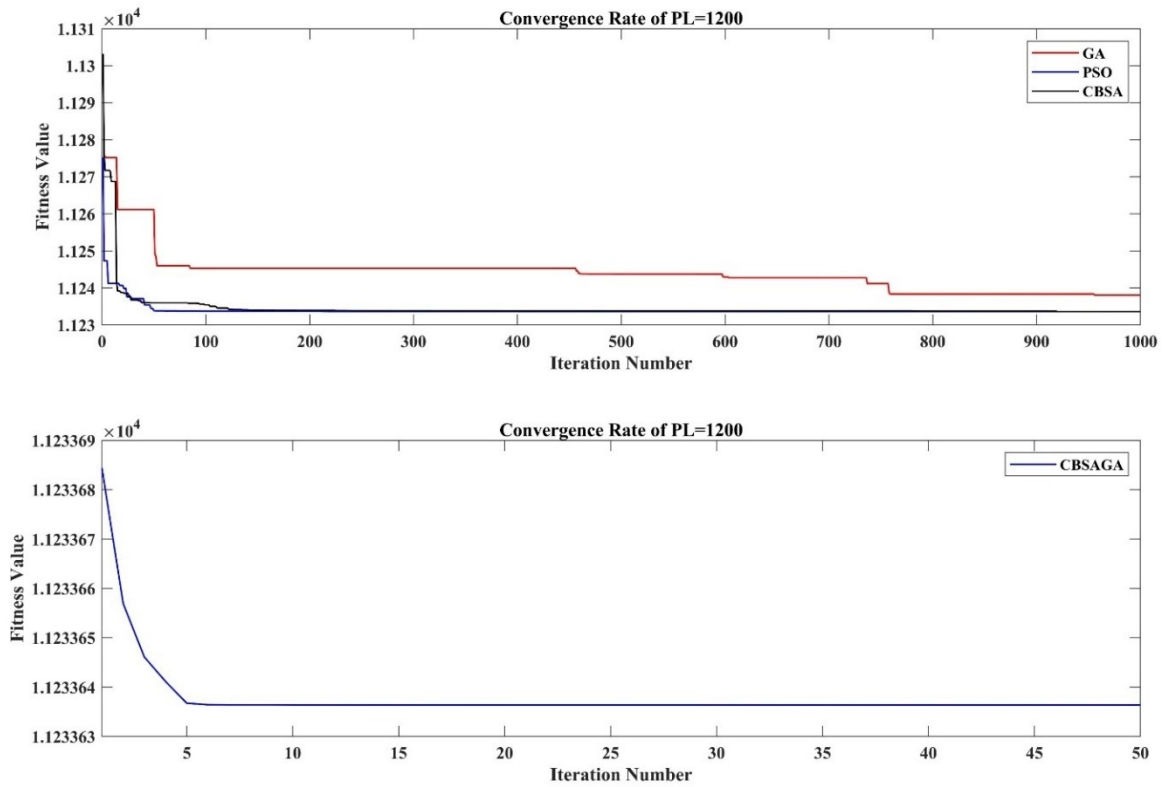


Fig. 6. Convergence rate of all optimization method for finding the best solution of economic dispatch problem for $P_L = 1200$.

3.2.2. Optimal location and size of DG

The main problem in finding the optimal size and location of DG units in the distribution system depends on the performance of the distribution network and the limitations of DG performance. In addition to the optimal choice of installation location, the desired amount of DG units will also have a great impact on the parameters of voltage, line current and even network power losses. In this section some of optimization method is presented for solving

optimal placement, and size of DG in distribution network. The objective function of proposed method in this paper for optimal placement of DG is total active power system losses. To show the efficiency of proposed optimization algorithm for solving the problem of optimal size, and location, of DG, IEEE 33 bus distribution systems have been used as case study. This problem has been modelled as a single objective optimization problem. The objective is the system real power losses minimization.

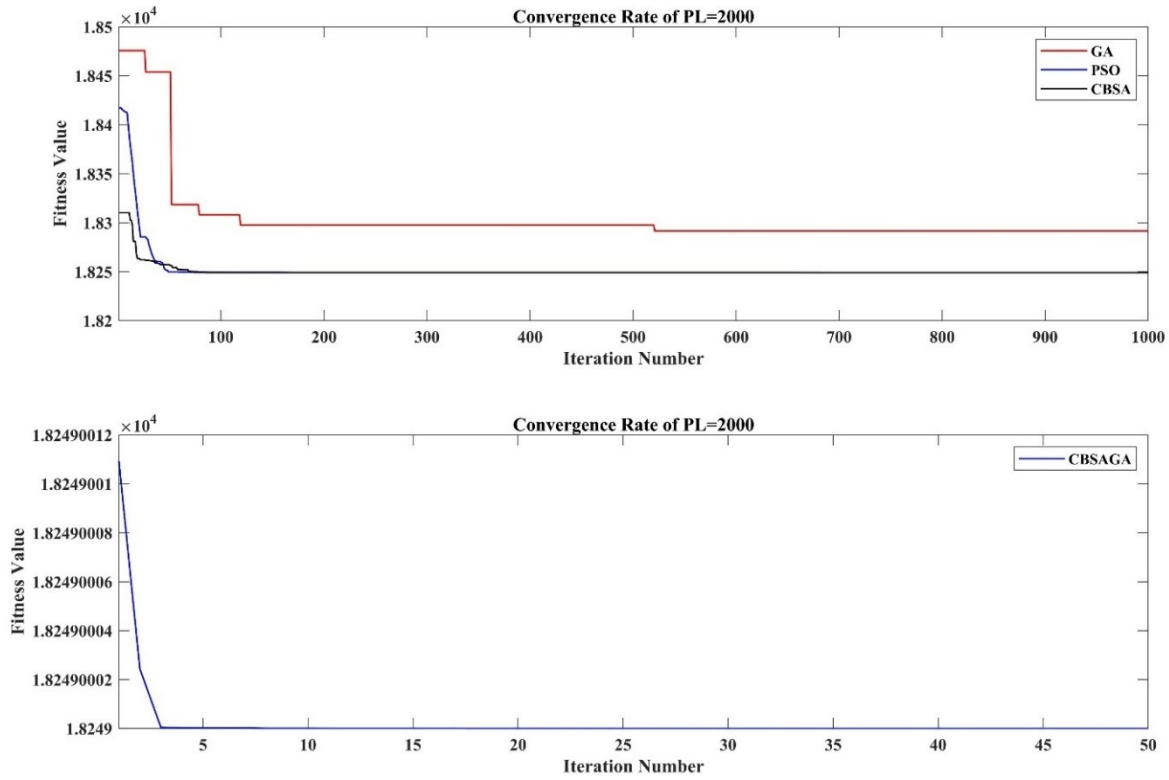


Fig. 7. Convergence rate of all optimization method for finding the best solution of economic dispatch problem for $P_L=2000$.

Power losses occur at all levels of the power system. Large amounts of losses occur in distribution systems. Therefore, reducing losses is a priority for distribution network operators. There are various methods to reduce power losses in distribution networks. One of these methods is to determine the optimal size and location of DGs in the distribution network. In other words, with the expansion of the use of DG at the distribution level on the one hand and on the other hand the priority of distribution network operators to reduce power losses, determining the optimal size and location of DG to reduce power losses is essential. Therefore, the objective function of DG placement and sizing is minimizing system real power losses which are expressed as follows:

$$fit = \min(P_{Loss}) \quad (4)$$

Where P_{Loss} is the total active power system losses and expressed as follow:

$$P_{Loss} = Real \left[\sum_{l=1}^{N_{line}} \left(\frac{|V_l^i - V_l^j|}{Z_l} \right)^2 * R_l \right] \quad (5)$$

Where in equation (5), V_l^i is the voltage of line l and bus i , V_l^j is the voltage of line l and bus j , Z_l is the impedance of line l , and R_l is the resistance of line l .

Proper operating conditions of the network are achieved when initially the governing constraints of network are satisfied and then the objective functions will be optimized. Constraints of the DG placement and sizing such as bus magnitude of bus voltage, line current for each line of the system, DG capacity, DG power factor are as follows:

$$V_l^{min} \leq V_i \leq V_l^{max} \quad (6)$$

Where V_i^{min} and V_i^{max} are the minimum and maximum voltages at each bus, respectively.

$$I_l \leq I_l^{max} \quad (7)$$

Where I_l is the line current of the l th line and I_l^{max} is the maximum capacity of the l th line of the distribution network.

$$\begin{aligned} P_{DG}^{min} &\leq P_{DG} \\ &\leq P_{DG}^{max} \end{aligned} \quad (8)$$

where P_{DG}^{min} and P_{DG}^{max} are the minimum and maximum power output limits of the DG.

The IEEE 33 bus distribution systems is considered as case study to demonstrate the performance of CBSA-GA algorithm in finding the optimal location, and size of the DG. The IEEE 33 bus distribution system has a nominal active and reactive load of 3.715 MW, and 2.3 MVar. Also, the power losses of this network are 210.99 kW and 143.13 kVar [31]. Fig. 8 shows the IEEE 33 bus distribution system.

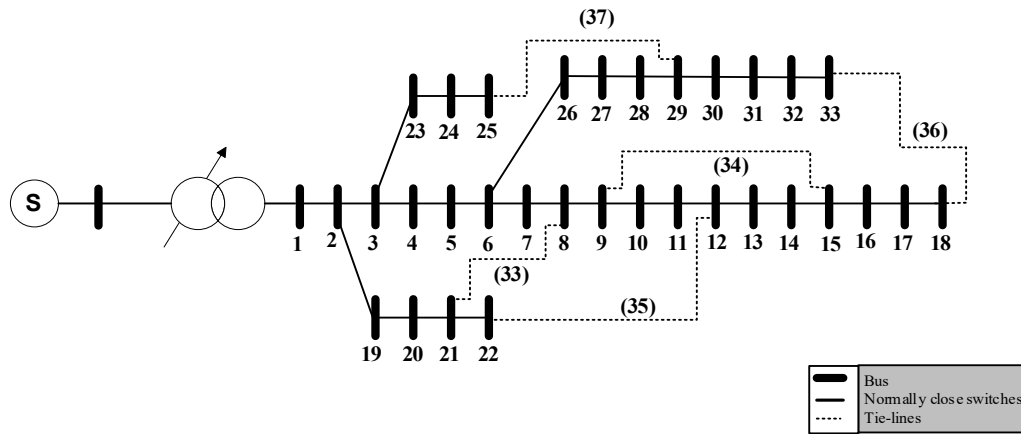


Fig. 8. One-line diagram of the IEEE 33-bus distribution system

The results of the simulation by considering three DG are presented in Table 7. In this case, the total power loss is reduced to 71.50 kW, which is the lowest one compared to other mentioned method in this table. Also, the percentage of reduction of power loss is 66.11%, which is the highest reduction percentage among the methods presented in this table.

Table 7. Comparison of results for IEEE 33-bus system with three DG units

Algorithm	Best (MW)	Worst (MW)	Avg. (MW)	Std. (MW)
GA	0.0726	0.1072	0.0869	0.0067

PSO	0.0720	0.0847	0.0742	0.0035
CBSA	0.0728	0.0785	0.0733	0.0016
CBSA-GA	0.0715	0.0768	0.0716	7.5664e-04

The convergence curve of all expressed optimization algorithms for optimal location and size of DG problem is shown in the Fig. 9. It can be seen that the convergence rate of CBSA-GA is good. As can be seen, the CBSA-GA has undergone fewer iteration to reach its near-global optimum. In other words, it converges faster to the near-global optimum point.

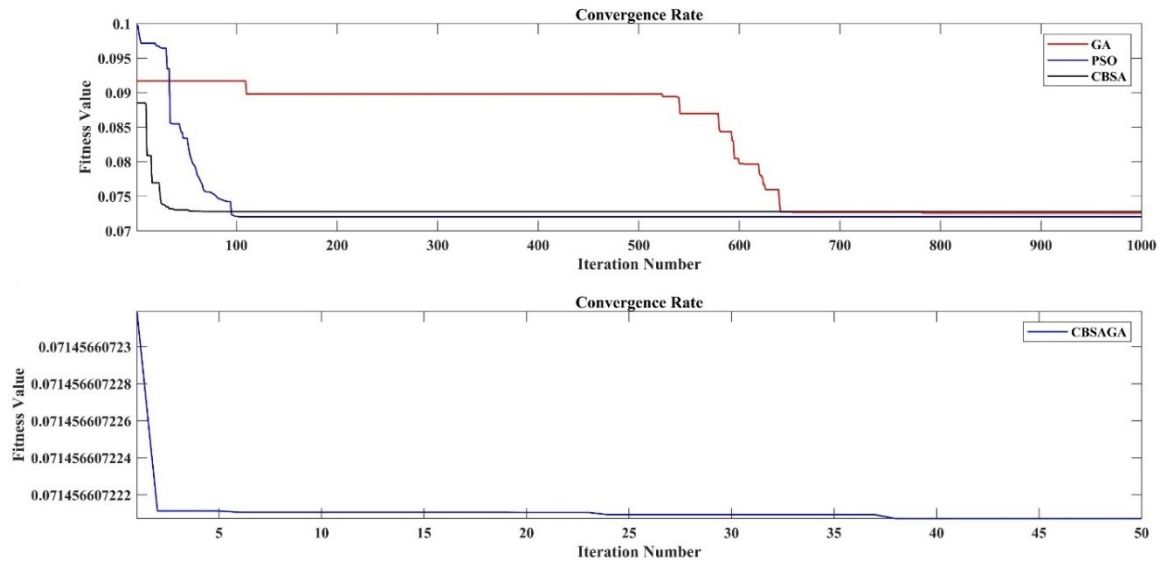


Fig. 9: Convergence rate of all optimization method for finding the best solution of optimal location and size of DG problem

As can be seen, with the installation of DG, power loss has been significantly reduced. The

Table 8. In this case when DGs at unit PF is installed in the network power loss decreased by 66.11%. Therefore, it can be concluded that by installing three DG units and determining

results are for when three DG are installed in the distribution network are presented in

their optimal location and size with CBSA-GA the network is in a very good condition. power loss have been significantly reduced.

Table 8. IEEE 33 bus distribution system results

Algorithm	Number of DG	Location	Size (kW)	Power factor	Total DG Size (kW)	Fit (MW)	Fit improvement (%)
CBSA-GA	Three DG	14	0.754	1	2.9248	71.5	66.1121
		24	1.0994	1			
		30	1.0714	1			
SFS [32]	Three DG	13	0.802	1	2.947	72.79	65.50
		24	1.091	1			
		30	1.054	1			
SA [33]	Three DG	6	0.632	1	2.4677	82.03	61.12
		18	0.487	1			
		30	0.550	1			
KHA [34]	Three DG	13	0.8107	1	2.4885	75.412	64.26

		25	0.8368	1			
		30	0.8410	1			
		12	0.8808	1			
QOTLBO [35]	Three DG	24	1.0592	1	3.0114	74.101	64.88
		29	1.0714	1			
		32	1.2000	1			
GA-PSO [36]	Three DG	16	0.8630	1	2.9880	103.40	50.99
		11	0.9250	1			

4. Conclusion

In this paper, we have developed an efficient hybrid technique called CBSA-GA, which demonstrates its effectiveness in finding optimal solutions for engineering optimization problems. The algorithm has been thoroughly evaluated through implementations on 13 benchmark functions and two real engineering problems in the field of power engineering. These problem instances encompass both constrained and unconstrained optimization scenarios. The results obtained from the proposed CBSA-GA algorithm showcase its superiority over other well-known optimization methods such as GA, PSO, and CBSA. Across the 13 benchmark functions, the CBSA-GA algorithm has consistently achieved excellent performance, converging to the absolute optimal value in three of the functions. Even in the remaining functions, the algorithm has delivered highly competitive results, approaching the optimal solutions closely.

A notable advantage of the CBSA-GA algorithm is its resilience to dimensionality. Increasing the dimensions of the optimization problem did not have a detrimental effect on the algorithm's performance. This robustness is a crucial attribute, as real-world engineering problems often involve complex high-dimensional search spaces. The CBSA-GA algorithm's ability to maintain its efficiency and accuracy across different problem

dimensions further strengthens its applicability in practical engineering scenarios. Considering the comprehensive evaluation and comparisons performed in this study, it is evident that the presented CBSA-GA algorithm exhibits commendable performance in solving both constrained and unconstrained optimization problems in the field of engineering. Its superior performance, especially when compared to other established optimization techniques, highlights its potential as an effective tool for tackling a wide range of engineering optimization challenges.

Future research directions could explore the application of the CBSA-GA algorithm to additional engineering domains and the investigation of its performance on larger-scale optimization problems. Moreover, incorporating problem-specific features and constraints into the algorithm could further enhance its capabilities in addressing real-world engineering optimization tasks. Overall, the CBSA-GA algorithm presents a promising approach that contributes to the advancement of optimization methodologies in engineering applications.

Reference

- [1] Lv, X. W., Huang, S., Hui, Z. W., & Ji, H. J. (2018). Test cases generation for multiple paths based on PSO algorithm with metamorphic relations. *Iet Software*, 12(4), 306-317.

- [2] Avila-George, H., Torres-Jimenez, J., Gonzalez-Hernandez, L., & Hernández, V. (2013). Metaheuristic approach for constructing functional test-suites. *Iet Software*, 7(2), 104-117.
- [3] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- [4] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- [5] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.
- [6] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, 76(2), 60-68.
- [7] Colnari, A., Dorigo, M., & Maniezzo, V. (1991, December). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life* (Vol. 142, pp. 134-142).
- [8] Glover, F. (1990). Tabu search—part II. *ORSA Journal on computing*, 2(1), 4-32.
- [9] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65-74.
- [10] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided design*, 43(3), 303-315.
- [11] Chu, C. W., Lin, M. D., Liu, G. F., & Sung, Y. H. (2008). Application of immune algorithms on solving minimum-cost problem of water distribution network. *Mathematical and Computer Modelling*, 48(11-12), 1888-1900.
- [12] Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *2007 IEEE congress on evolutionary computation* (pp. 4661-4667). Ieee.
- [13] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232-2248.
- [14] Naruei, I., & Keynia, F. (2021). A new optimization method based on COOT bird natural life model. *Expert Systems with Applications*, 183, 115352.
- [15] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & structures*, 169, 1-12.
- [16] Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. (2020). Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 94, 103731.
- [17] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702-713.
- [18] Punathanam, V., & Kotecha, P. (2016). Yin-Yang-pair Optimization: A novel lightweight optimization algorithm. *Engineering Applications of Artificial Intelligence*, 54, 62-79.
- [19] He, S., Wu, Q. H., & Saunders, J. R. (2009). Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE transactions on evolutionary computation*, 13(5), 973-990.
- [20] Moghdani, R., Elaziz, M. A., Mohammadi, D., & Neggaz, N. (2021). An improved volleyball premier league algorithm based on sine cosine algorithm for global optimization problem. *Engineering with Computers*, 37, 2633-2662.
- [21] Hasanazadeh, M. R., Keynia, F., & Hashemipour, M. (2022). A new index-based hyper-heuristic algorithm for global optimisation problems. *IET Software*, 16(5), 493-515.
- [22] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*, 2(2), 78-84.
- [23] Zhao, S., Zhang, T., Ma, S., & Chen, M. (2022). Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. *Engineering Applications of Artificial Intelligence*, 114, 105075.
- [24] Sulaiman, M. H., Mustaffa, Z., Saari, M. M., & Daniyal, H. (2020). Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 87, 103330.
- [25] Wang, T. L., Gu, S. W., Liu, R. J., Chen, L. Q., Wang, Z., & Zeng, Z. Q. (2025). Cuckoo catfish optimizer: a new meta-heuristic optimization algorithm. *Artificial Intelligence Review*, 58(10), 326.
- [26] Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2023). Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, 35(5), 4099-4131.
- [27] Lang, Y., & Gao, Y. (2025). Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. *Computer Methods in Applied Mechanics and Engineering*, 436, 117718.
- [28] Abdel-Basset, M., Mohamed, R., Jameel, M., & Abouhawwash, M. (2023). Spider wasp optimizer: A novel meta-heuristic optimization algorithm. *Artificial Intelligence Review*, 56(10), 11675-11738.
- [29] Mohammed, B. O., Aghdasi, H. S., & Salehpour, P. (2025). Dhole optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Cluster Computing*, 28(7), 430.
- [30] Memarzadeh, G., & Keynia, F. (2021). A new optimal energy storage system model for wind power producers based on long short term memory and Coot Bird Search Algorithm. *Journal of Energy Storage*, 44, 103401.
- [31] Memarzadeh, G., & Keynia, F. (2020). A new index-based method for optimal DG placement in distribution networks. *Engineering Reports*, 2(10), e12243.

- [32] Nguyen, T. P., Tran, T. T., & Vo, D. N. (2019). Improved stochastic fractal search algorithm with chaos for optimal determination of location, size, and quantity of distributed generators in distribution systems. *Neural Computing and Applications*, 31, 7707-7732.
- [33] Injeti, S. K., & Kumar, N. P. (2013). A novel approach to identify optimal access point and capacity of multiple DGs in a small, medium and large scale radial distribution systems. *International Journal of Electrical Power & Energy Systems*, 45(1), 142-151.
- [34] Sultana, S., & Roy, P. K. (2016). Krill herd algorithm for optimal location of distributed generator in radial distribution system. *Applied Soft Computing*, 40, 391-404.
- [35] Sultana, S., & Roy, P. K. (2014). Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems. *International Journal of Electrical Power & Energy Systems*, 63, 534-545.
- [36] Moradi, M. H., & Abedini, M. (2012). A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems. *International Journal of Electrical Power & Energy Systems*, 34(1), 66-74.