

Received 1 September 2023, accepted 13 September 2023, date of publication 18 September 2023,
date of current version 27 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3316619

RESEARCH ARTICLE

A New Metaphor-Free Metaheuristic Approach Based on Complex Networks and Bezier Curves

KARLA AVILA¹, ERIK CUEVAS¹, MARCO PEREZ¹, (Senior Member, IEEE),
AND RAM SARKAR², (Senior Member, IEEE)

¹Department of Electro-Photonic Engineering, University of Guadalajara, Guadalajara 44430, Mexico

²Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

Corresponding author: Karla Avila (karla.acardenas@academicos.udg.mx)

ABSTRACT A metaheuristic method is an optimization technique that is generally inspired by natural or physical processes. The use of metaphors has created a tendency to reproduce existing algorithms with slight modifications or variations rather than encouraging the development of novel algorithmic techniques and principles. On the other hand, a complex network is a mathematical structure whose main characteristic is the ability to capture and analyze the intricate patterns and properties that emerge from the interactions between the elements that it connects. In this paper, a new metaphor-free metaheuristic algorithm based on complex networks and Bezier curves is presented. In this approach, candidate solutions are represented as nodes in a graph, whereas the connections between nodes or edges reflect the differences in their objective function values. Therefore, the graph provides a higher-level representation that captures the essential relationships and dependencies among the solutions. Once the graph is generated, the shortest path between each solution and the best solution is obtained. Then, the nodes obtained from this process are used as control points in the Bezier equation to generate the new agent position. Therefore, during the optimization process, the graph is continuously modified based on the evaluation of new candidate solutions and their objective function values, producing trajectories that allow the exploration and exploitation of the search space. The experimental results demonstrated the effectiveness of our approach by achieving competitive results compared to other well-known metaheuristic algorithms on various benchmark functions.

INDEX TERMS Complex networks, metaheuristics, Bezier curves, optimization, complex systems, graphs theory.

I. INTRODUCTION

Optimization [1] is the process of determining the best solution for a problem within a given set of constraints. This involves maximizing or minimizing an objective function while satisfying specific conditions or limitations. Optimization methods can be broadly classified into two categories: classical and metaheuristic. Classical optimization methods refer to a set of well-established mathematical techniques that aim to find the optimal solution through systematic mathematical analysis and calculations. Metaheuristic optimization methods [2] are general-purpose algorithms inspired by natural or physical processes or social behavior. Metaheuristic techniques offer several advantages [3] over classical opti-

mization methods. They do not require explicit mathematical models and can handle complex, large-scale, or nondifferentiable problems.

Metaheuristic methods can be classified into several categories [2] based on their underlying principles, sources of inspiration, or search strategies. Nevertheless, the most common classification divides these techniques into evolution-based, swarm intelligence, and physics-based algorithms [4], [6]. Evolution-based algorithms simulate biological evolution by employing selection, crossover, mutation, and reproduction operators to generate better candidate solutions. The most popular algorithms in this category are Genetic Algorithm (GA) [7], Genetic Programming [8], Tabu Search (TS) [9], Differential Evolution (DE) [10], Evolutionary Programming [11], and Evolutionary Strategies [12]. Swarm intelligence algorithms are based on the

The associate editor coordinating the review of this manuscript and approving it for publication was Nazar Zaki¹.

collective social behavior of animals and insects. In this case, candidate solutions are obtained through the interactions between individuals and their environments. The most important metaheuristic algorithms based on swarm intelligence include Particle Swarm Optimization (PSO) [13], Bat Algorithm (BA) [14], Cuckoo Search Algorithm (CS) [15], crown search algorithm (CSA) [16], and Grey Wolf Optimization (GWO) [17]. Finally, physics-based algorithms use the universe's physical rules to define the search operators and find candidate solutions. Examples of such algorithms include simulated annealing (SA) [18], Harmony Search (HS) [19], Sine Cosine Algorithm (SCA) [20], and state-of-matter search (SMS) [21].

Metaheuristic algorithms have demonstrated their superiority in several real-world applications where classical techniques cannot be used [22], [23], [24], [25], [26], [27], [28]. However, it is impossible that all metaheuristic techniques can solve all problems competitively due to the inherent complexity and diversity of optimization problems [4]. Each problem has unique characteristics and requires specific strategies to achieve optimal solutions. Therefore, a single metaheuristic algorithm cannot be universally effective for all problem types. Additionally, many metaheuristic algorithms may have some limitations, such as slow convergence rate, difficulty handling high-dimensionality, parameters and noise sensibility, and convergence to local optima. Under such conditions, introducing new methods is necessary to expand the capabilities of metaheuristics and improve their applicability to various problem domains [5]. By developing novel techniques, researchers attempt to address a higher number of problems and enhance the effectiveness and efficiency of optimization algorithms. New approaches can provide novel ways to solve the limitations of existing metaheuristic algorithms, improving the convergence rate, solution quality, and robustness.

Traditional metaheuristic algorithms incorporate metaphors or analogies from various natural or physical systems to guide the search process. Metaphors have played a significant role in developing metaheuristic algorithms and have provided valuable insights and inspiration. However, potential drawbacks and limitations are associated with the use of metaphors in this context [29]. The exclusive use of metaphor-based principles may hinder innovation in the design of algorithms. It can create a tendency to reproduce existing algorithms with slight modifications or variations rather than encouraging the exploration of novel algorithmic techniques and principles [30]. This limits the potential for algorithmic breakthroughs and advancements. In contrast, a metaphor-free metaheuristic algorithm [31] is an optimization algorithm that does not rely on any specific metaphor or analogy to guide the search process. A metaphor-free metaheuristic algorithm is designed to avoid any direct reference to particular metaphors or analogies. Instead, they focus on developing algorithmic mechanisms based on the combination of mathematical and computational principles [32]. The

most important advantage of metaphor-free metaheuristic algorithms over classical metaheuristics is their potential for novel and creative problem-solving approaches. Metaphor-free methods can explore unconventional search operators based on mature mathematical computational principles that have already been successful in various scientific areas.

Complex networks [33] are mathematical and computational models used to represent and analyze relationships or interactions between entities or components in a system. They present a structure consisting of nodes (also called vertices) and edges. Nodes represent individual entities or elements, and edges represent the connections or relationships between them. The main characteristic of complex networks compared to other modeling techniques, is their ability to capture and analyze the intricate patterns and properties that emerge from the relationships and interactions between components in a system. In complex networks, one notable characteristic is that the number of nodes and edges can dynamically change over time [34]. This dynamic nature of complex networks enables the representation of systems that undergo adaptation or reconfiguration. Complex networks, as a modeling technique, have found notable applications in various fields. Some examples include social networks [35], [36], [37] and economic analysis [38], [39] and telecommunications and energy networks [40], [41], [42], [43], [44].

Bezier curves are polynomial curves used for geometry approximation [45]. The Bezier curves are defined by a set of control points that influence their shapes. The control points act as handles that guide the curve, allowing precise control over the curvature and direction. In the trajectory, each point is computed considering Bernstein polynomials, which calculate the position of the point according to any given number of control points. By varying the positions and number of control points, different Bezier curves can be created, ranging from straight lines to complex curves. Bezier curves have been combined with metaheuristic methods to create interesting applications [46], [47], [48], [49], [50], [51], [52]. In these approaches, metaheuristic techniques have been employed to manipulate the control points of Bezier curves, resulting in meaningful shapes within a specific domain. The curves can be tailored to satisfy certain criteria or constraints by optimizing the positions of the control points. However, the proposed method uses a different approach. Instead of using metaheuristic techniques to modify the control points of the Bezier curves for shape manipulation, we utilize the trajectories generated by the Bezier curves to create new candidate solutions that explore and exploit the search space. In this method, each control point in the Bezier curve corresponds to a candidate solution, and the curve defines a trajectory that guides the movement of each agent in the search space.

In this paper, a new metaphor-free metaheuristic approach based on complex networks and Bezier curves is presented. This method generates graphs that represent the fitness relationships between the possible solutions. The shortest path between each solution and the best solution is then obtained.

The nodes obtained from this process are used as control points in the Bezier equation to generate feasible agent trajectories. During the optimization process, the graph is continuously modified based on the evaluation of new candidate solutions and their objective function values, producing trajectories that allow the exploration and exploitation of the search space. This methodology reduces the complexity of high-dimensional functions by modeling only the relationships between search agents using graphs. A set of benchmark functions, including multimodal, unimodal, and hybrid functions, was used to compare the performance of the proposed approach numerically and statistically with that of several state-of-the-art metaheuristic algorithms. The experimental results indicated that the proposed approach produces competitive results in terms of accuracy and robustness.

The remainder of this paper is organized as follows: In Section II, the preliminary concepts of complex networks, shortest path problem, and Bezier curves are presented. The proposed method is explained in Section III. The experimental results and comparative analysis are presented in Section IV. Finally, in Section V, conclusions are offered.

II. PRELIMINARY CONCEPTS

The main objective of this section is to provide an overview and discussion of the most important concepts that form the foundation of the proposed approach. By addressing these concepts, we aim to establish a common understanding of the fundamental elements and principles that underlie the proposed method.

A. COMPLEX NETWORKS

Complex networks refer to the study and analysis of systems composed of interconnected elements, represented as nodes or vertices, and the relationships between them, expressed as edges or links. In a complex network, the nodes can represent various elements, such as individuals, organizations, or candidate solutions, while the edges capture the connections, interactions, or dependencies between these elements. Complex networks, as a modeling technique, possess notable characteristics that enable the analysis of interconnected systems. They capture the intricate patterns of connectivity and topology, revealing features such as distributions and clusters. With unique structural properties, complex networks facilitate efficient information flow and exhibit power-law degree distributions. They offer various network analysis techniques, including metrics, centrality measures, and community detection algorithms. Moreover, the dynamic nature of complex networks [53] allows for modeling growth, adaptation, and reconfiguration, enabling the study of evolving systems and dynamical phenomena. Complex networks also represent important mechanisms to simplify the complexity of high-dimensional problems by representing them as configurations of nodes and edges. In high-dimensional problems, the number of variables or dimensions involved can be overwhelming, making analysis and understanding challenging. However, by employing complex networks, these

problems can be transformed into a network representation where nodes represent the variables or components, and edges denote the relationships or interactions between them.

A simple network (or graph) consists of a set of nodes (or vertices) and a set of edges (or links) that connect nodes. The mathematical definition of a network is shown in equation (1). Where V is a finite set of nodes, $E \subseteq V \otimes V = \{e_1, e_2, \dots, e_m\}$ is a set of links, and f is a mapping which associates some elements of E to a pair of elements of V , such as that if $v_i \in V$ and $v_j \in V$, then $f: e_p \rightarrow [v_i, v_j]$ and $f: e_q \rightarrow [v_j, v_i]$ [33].

$$G = (V, E, f) \quad (1)$$

Graphs can maintain weights in the links. They are known as weighted graphs [34]. In a weighted graph, each edge or link between nodes is assigned a numerical value called a weight. These weights represent some measure of importance, distance, cost, or any other relevant quantity associated with the node connection.

Graphs can be arranged in a matrix known as an adjacency matrix. It should be mentioned that a vertex is adjacent to another node if there is an edge to it from that vertex. The adjacency matrix A is defined as the $|V| \times |V|$ matrix, where V is the number of nodes, and each element A_{ij} contains the mapping relation between nodes i and j (usually represented by their weights). If the network has no self-loops, then the diagonal elements of A are zero.

B. SHORTEST PATH PROBLEMS

The shortest-path problem (SPP) is a well-studied topic in computer science. In graph theory, communication between two non-adjacent nodes depends on the path connection. A path is defined as a sequence of nodes in which each successive node is adjacent to its predecessor [53]. In other words, given a set of vertices V , a source vertex u , a destination vertex v , where $u, v \in V$, and a set of weighted edges E over set V , the shortest path is the minimum length criterion sum $S(u, v)$ from source to destination (see Equation 2), where $d(u, v)$ denotes the edge weights of the nodes [54].

$$S(u, v) = \sum_{u, v \in V(G)} d(u, v) \quad (2)$$

Dijkstra's algorithm [55], [56], [57] is a popular method for finding the shortest path between two nodes in a weighted graph. The Dijkstra's algorithm is specifically designed to find the path with the minimal cost in terms of the weights of the edges in a graph. This algorithm employs a greedy approach to iteratively determine the shortest paths from a source node to all other nodes in the graph. It considers two types of vertices: solved and unsolved vertices. First, a source vertex is defined and marked as the solution. Then, all other edges (through unsolved nodes) connected to the source vertex are checked. Once the algorithm identified the shortest link, it added the corresponding vertex to the solved node list. The algorithm iterates until all vertices are solved

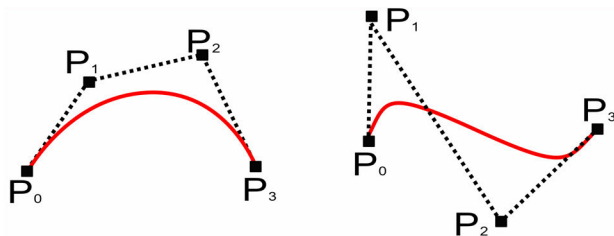


FIGURE 1. Examples of BC using four different control points.

or the destination node is reached. However, it does not need to analyze all links.

C. BEZIER CURVES

Bezier curves (BC) are a well-known standard tool in computer graphic modeling [45]. They are polynomial approximations based on a set of points known as control points. Figure 1 shows some examples of BC using different control points. As illustrated in the image, the curve is constructed by mapping the polynomial straight lines. Where P_0 and P_3 are the initial and final points on the curve, respectively. In addition, the curve does not pass through P_1 or P_2 , because these points are used to provide directional information. Depending on the number of control points, various curve shapes can be obtained.

BC uses a linear combination of Bernstein polynomials (see equation (3)). Where $B_i^n(x)$ are elements of a binomial distribution and c_i are the approximate function values generated based on the range $[a, b]$. This is mathematically described by equations (4) and (5): Thus, a BC of degree n can be defined as shown in equation (6). Where P_i represents the control points. BC present the following characteristics: 1) convex hull, which is bounded by the convex hull of the control points; 2) symmetry, the opposite order of the control points produces the same curve with reverse parameterization; 3) affine invariant, any move operation (translate, scale, rotate, or skew) affects the entire curve; 4) any subdivision of a BC retains its properties; and 5) easy programming, which can be subdivided into simple recursive steps to compute high-order BC [58].

$$P(x) = \sum_{i=0}^n c_i B_i^n(x) \quad (3)$$

$$B_i^n(x) = \binom{n}{i} x^i (1-t)^{n-i} \quad (4)$$

$$c_i = f\left(i \frac{b-a}{n} + a\right) \quad (5)$$

$$B(t) = \sum_{i=0}^n \binom{n}{i} P_i (1-t)^{n-i} t^i \quad (6)$$

III. COMPLEX NETWORKS AND BEZIER CURVES AS A METAHEURISTIC METHOD

In this paper, a new metaphor-free metaheuristic algorithm that combines complex networks and Bezier curves is pre-

sented. For this reason, this approach is called through the manuscript CNBC. The objective of the algorithm is to find the global solution for a nonlinear problem based on the formulation of an optimization problem described as follows:

$$\begin{aligned} & \text{Maximize/ minimize} \\ & \text{Subject to } J(\mathbf{x}) \\ & \mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d \\ & \mathbf{x} \in X \end{aligned} \quad (7)$$

where $J: \mathbb{R}^d \rightarrow \mathbb{R}$ is a d -dimensional nonlinear function and X represent the search space ($\mathbf{x} \in \mathbb{R}^d \mid l_i \leq x_i \leq u_i, i=1, \dots, d$) defined by lower (l_i) and upper (u_i) bounds. Regardless of whether a metaheuristic method is classical or metaphor-free, they typically maintain a similar structure consisting of three main elements: initialization, movement operators, and a selection mechanism. In this section, we will explain each of these elements for the proposed approach.

A. INITIALIZATION

Initialization is the first operation of the algorithm, and its objective is to generate an initial population of N agents, $Ag = \{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ where each agent \mathbf{a}_i represents the combination of decision variables $\mathbf{a}_i = \{a_{i,1}, \dots, a_{i,d}\}$. They can be initialized based on the specifications of a defined problem; however, their initial positions are typically set randomly. To make this possible, the positions for each decision variable $a_{i,j}$, where ($i \in 1, \dots, N; j \in 1, \dots, d$), are established with a numerical value uniformly distributed between the lower (l_i) and upper (u_i) bounds, as shown in Equation 8.

$$a_{i,j} = l_i + \text{rand}(0, 1) \cdot (u_i - l_i) \quad (8)$$

B. MOVEMENT OPERATORS

Once the population is initialized, an iterative process is executed to produce a new agent position in each iteration. In metaheuristics, exploration and exploitation are two important behaviors. The first focuses on an examination of the search space, looking for new candidate solutions. The second refines the search process to obtain a more accurate solution. For CNBC, both effects are produced by considering two main processes: the creation of a complex network and the generation of trajectory.

1) CREATION OF A COMPLEX NETWORK

In the proposed approach, the complex network is structured in a way that each node in the network represents an agent or a candidate solution from the population. These nodes capture the individual solutions that are being explored and evaluated. On the other hand, the edges in the network represent the relationships between pairs of agents in terms of their fitness values. Specifically, if we consider the node n_i , it corresponds to the candidate solution \mathbf{a}_i ($i \in 1, \dots, N$). The edge or link $fitDif_{i,j}$ (see equation (9)) connecting the node n_i to node n_j ($j \in 1, \dots, N$) in the network represent the absolute difference

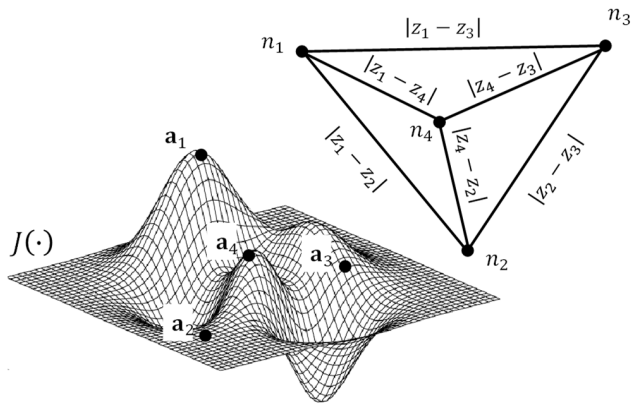


FIGURE 2. Example where the agents or particles in the search space are associated with the construction of a complex network.

between the fitness or objective values of the solutions associated with agents \mathbf{a}_i and \mathbf{a}_j . Therefore, the value of $fitDif_{i,j}$ is computed as follows:

$$fitDif_{i,j} = abs(z_i - z_j) \tag{9}$$

where $z_i = J(\mathbf{a}_i)$ and $z_j = J(\mathbf{a}_j)$ symbolize the values in terms of the objective function $J(\cdot)$ produced by agents \mathbf{a}_i and \mathbf{a}_j , respectively. In Figure 2, we can observe an example where the agents or particles in the search space are associated with the construction of a complex network. The figure displays four agents $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$ distributed in different positions within the search space. The graph visualizes how each agent is represented as a node $\{n_1, n_2, n_3, n_4\}$ in the complex network. In this representation, the nodes of the graph correspond to the individual agents, and the edges connecting the nodes capture the relationships between them. The weights assigned to these edges reflect the differences observed among the particles regarding their fitness values. In Algorithm 1, the creation of the complex network is presented in lines 6 and 7.

Once the complex network has been constructed, the next step in the proposed approach is to identify the node (agent) with the best value in terms of the objective function. This node n_B represents the solution \mathbf{a}_B that currently has the highest fitness or optimality within the population. By identifying the best node n_B , the algorithm determines the reference point or benchmark for evaluating the other candidate solutions. It serves as a guide for exploring and exploiting the search space.

After identifying the best node n_B , the proposed algorithm calculates the new position for each particle or agent in the population. Therefore, for each node n_i or agent \mathbf{a}_i (excluding the node n_B that represents the solution with the best value), it is applied the Dijkstra’s algorithm to find the set of nodes $SN_{i,B}$ that form the shortest path connecting node n_i with the best node n_B in the network (see Algorithm 1 line 9). In Figure 3, we can observe an example that demonstrates the application of Dijkstra’s algorithm in a specific configuration of a complex network. The complex network

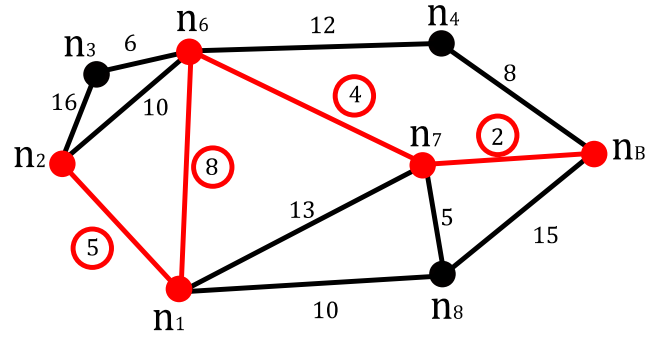


FIGURE 3. Application of Dijkstra’s algorithm in a specific configuration of a complex network.

comprises eight nodes, and the values of the edges represent the relationships between the differences in fitness values for pairs of nodes. Each edge indicates the absolute difference in fitness between the corresponding nodes. To determine the new position of agent \mathbf{a}_2 (or node n_2), the algorithm utilizes Dijkstra’s algorithm. This algorithm is applied to find the set of nodes $S_{2,B}$ that form the shortest path between n_2 and the best node n_B in the network. By identifying this shortest path, we can determine the trajectory that agent \mathbf{a}_2 should follow to approach the optimal solution. Upon applying Dijkstra’s algorithm, the resulting set of nodes that form the shortest path between n_2 and n_B consists of five nodes $SN_{2,B} = \{n_2, n_1, n_6, n_7, n_B\}$. In Figure 3, these five nodes are highlighted in red. As observed in Figure 3, the Dijkstra’s algorithm effectively determines the path with the minimum cost or weight in terms of the fitness differences among the candidate solutions. By following the principle of selecting the path with the minimal cost, Dijkstra’s algorithm ensures that the trajectory chosen for a specific agent optimally balances the exploration and exploitation of the search space. The algorithm prioritizes the edges with lower weights, indicating smaller differences in fitness values, as it seeks to find the most efficient route towards the best solution.

2) GENERATION OF TRAJECTORY

For the generation of trajectories, for each agent \mathbf{a}_i from the population Ag , the set $SN_{i,B}$ of nodes obtained Dijkstra’s algorithm are used in Eq. (6) as control points. In this step, it is generated a series of points that lie along the curve. The number of the points N_{pts} generated along the Bezier curve will determine the granularity of the trajectory. A higher density of points will result in a smoother trajectory but may also increase the computational complexity. In this paper, the value of N_{pts} has been set to 100.

In the proposed method, updating the position of agent \mathbf{a}_i involves dividing the trajectory into four segments, namely A, B, C, and D. Each segment corresponds to a quarter of the complete trajectory, containing 25 points. The purpose of dividing the trajectory into segments is to introduce variability in the agent’s movement and exploration. When updating the position of agent \mathbf{a}_i , two different positions

\mathbf{p}_1 and \mathbf{p}_2 from the trajectory are considered. These positions are selected based on the results of the search strategy. The specific segments from which the positions are chosen depending on the algorithm's performance. To keep track of the algorithm's performance, a counter called *Cont* is implemented. *Cont* is incremented each time the algorithm fails to find a better solution than the current best solution. This counter serves as a measure of stagnation or lack of progress in the search. By monitoring the value of *Cont*, the algorithm can adapt its behavior and explore different segments of the trajectory to potentially discover new and better solutions. In Algorithm 1, the generation of the trajectory is represented by *GetBezierCurve* function (line 10).

In the proposed method, the selection of points \mathbf{p}_1 and \mathbf{p}_2 from the trajectory is determined based on the performance of the algorithm, as indicated by the value of the counter *Cont*. Four different cases are considered, each corresponding to a different scenario or behavior of the algorithm during the search. The cases are determined by the value of the counter *Cont* and dictate which of trajectory segments, namely A, B, C, or D, are used to select the points \mathbf{p}_1 and \mathbf{p}_2 . The selection of the segments is important as it influences the exploration-exploitation balance of the algorithm. If the algorithm frequently finds better solutions (case I), indicating a successful search strategy, the points \mathbf{p}_1 and \mathbf{p}_2 are randomly selected from the segment A. This case represents a scenario where the algorithm is actively exploring and discovering new, improved solutions. If the counter *cont* indicates that the algorithm takes a bit longer to find good solutions (case II), the points \mathbf{p}_1 and \mathbf{p}_2 are selected from both segments A and B. This case reflects a slightly slower search strategy, where the algorithm explores more within the initial segments of the trajectory. Similarly, if the algorithm takes even longer to find good solutions (case III), the points \mathbf{p}_1 and \mathbf{p}_2 are randomly selected from segments A, B, and C. This case represents a scenario where the algorithm explores more extensively throughout the trajectory, including the earlier segments. Finally, if the algorithm struggles significantly in finding good solutions (case IV), the points \mathbf{p}_1 and \mathbf{p}_2 are randomly selected from all four segments, A, B, C, and D. This case reflects a more exploratory behavior, where the algorithm extensively explores different parts of the to overcome stagnation and find better solutions.

The selection of the appropriate case and segments for choosing points \mathbf{p}_1 and \mathbf{p}_2 is determined by the value of the counter *cont*. This adaptive approach allows the method to adjust its behavior dynamically based on the success or difficulty encountered in the search process. Equation (10) specifies the different cases and the corresponding segments used for selecting points \mathbf{p}_1 and \mathbf{p}_2 .

$$Case = \begin{cases} I & 0 \leq Cont < 30 \\ II & 30 \leq Cont < 70 \\ III & 70 \leq Cont < 120 \\ IV & Cont \geq 120 \end{cases} \quad (10)$$

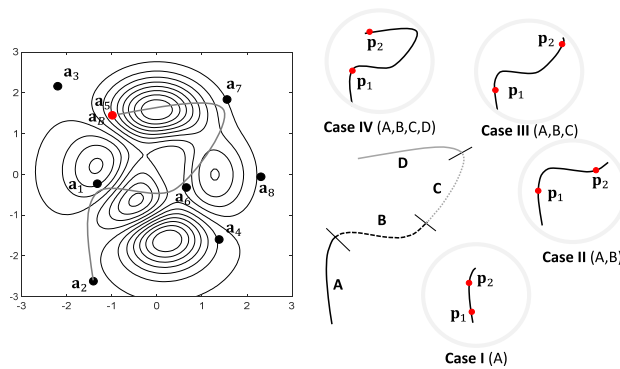


FIGURE 4. Trajectory example generated by using the proposed approach.

Once the points \mathbf{p}_1 and \mathbf{p}_2 have been determined based on the corresponding case, the new position of the agent $\mathbf{a}_i(k+1)$ is computed using equation (11). This equation defines the calculation of the new position considering the actual agent $\mathbf{a}_i(k)$ and the middle point obtained by the selected points \mathbf{p}_1 and \mathbf{p}_2 computed by a scaling factor *rand* (0, 1). The computational implementation of this step is shown in Algorithm 1, line 11.

$$\mathbf{ua}_i(k+1) = \mathbf{a}_i(k) + rand(0, 1) \cdot [(\mathbf{p}_1 + \mathbf{p}_2) / 2] \quad (11)$$

A bound checking is also implemented inside the *UpdatePosition* function (see Algorithm1, line 11) to ensure that the updated point $\mathbf{ua}_i(k+1)$ is within the dimensional search space. equation 12 shows the bound-checking method.

$$\begin{aligned} a_{i,j} &= l_i + rand(0, 1) \cdot (u_i - l_i), \\ a_{i,j} &< l_i \vee a_{i,j} > u_i \end{aligned} \quad (12)$$

Figure 4 illustrates an example trajectory generated using the proposed approach. The trajectory aims to connect agent \mathbf{a}_2 with the best element \mathbf{a}_B identified in the complex network. By applying Dijkstra's algorithm, the set of nodes $SN_{2,B}$ comprising the agents \mathbf{a}_2 , \mathbf{a}_1 , \mathbf{a}_6 , \mathbf{a}_7 and \mathbf{a}_B is determined as the shortest path. Using the Bezier curve, the trajectory is calculated and depicted in the figure. The trajectory is divided into four distinct parts: A, B, C, and D, each representing a quarter of the complete trajectory. These segments are indicated in the figure. Additionally, the figure shows the different cases that arise during the selection of points \mathbf{p}_1 and \mathbf{p}_2 for determining the new position of agent \mathbf{a}_2 . The cases depend on the performance of the algorithm and are reflected in the selection of the corresponding segments of the curve. If the algorithm consistently finds better solutions, the points \mathbf{p}_1 and \mathbf{p}_2 are selected randomly from segment A. If the algorithm takes slightly longer to find improved solutions, the two points are selected from segments A and B. This pattern continues for the other cases involving different combinations of segments. The figure also illustrates the hypothetical positions of points \mathbf{p}_1 and \mathbf{p}_2 for each case, provides a visual representation and understanding of the selection process.

3) SELECTION MECHANISM

In the proposed approach, the position of each agent is updated based on two conditions. The first condition evaluates the fitness of the new agent after its position is changed. If the new fitness value is better than the previous value, indicating an improvement, then the position is updated.

However, the second condition allows the possibility of updating the agent's position even if it does not result in a better fitness value. This condition is determined by an updating probability Up . If the updating probability is met, the agent's position is changed regardless of the fitness improvement (see Algorithm 1 lines 14-20). This introduces an element of exploration in the search process, allowing the agent to potentially explore new areas of the search space that may lead to better solutions. In this work, the value of Up has been set to 0.3.

By incorporating these two conditions, the algorithm balances exploitation (improving current solutions) and exploration (searching for potentially better solutions) during the optimization process. Agents whose fitness improves will update their positions. In contrast, agents with lower fitness values may still have a chance to explore new regions of the search space based on the moving probability.

4) COMPUTATIONAL PROCEDURE

Algorithm 1 summarizes the operations of the proposed method as pseudo-code. This approach considers as input values: the number of agents N , the maximum number of access functions af , and the updating probability Up .

First, the proposed method generates a set of N evenly distributed agents (line 2). These represent the initial population $\{\mathbf{a}_1, \dots, \mathbf{a}_N\}$. Subsequently, the best agent \mathbf{a}_B from $\{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ is selected, and the iterative process begins based on the maximum number of access functions af . At each iteration, the fitness differences between all elements of $\{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ are calculated (line 6), producing a preprocessing adjacent matrix $fitDif$ that satisfies Dijkstra's algorithm requirements (see Section II-B). Subsequently, a graph was generated (line 7). This network captures the relationships among all the agents through their fitness values. Subsequently, Dijkstra's algorithm is applied (line 9) to find the shortest path through each agent and the best path (\mathbf{a}_B). The path nodes obtained from this process are used as control points for the Bezier curve equation to generate feasible agent trajectories (line 10). To increase the randomness and diversity of solutions and to generate a non-deterministic new agent position, two points \mathbf{p}_1 and \mathbf{p}_2 from each trajectory are operated based on the $Cont$ value (which changes based on the necessity of the method to explore or exploit the search space) (line 11). Finally, all new agent positions are evaluated and updated based on two cases: 1) direct updating for all solutions that improve the previous solution and 2) probability-based updating for the rest of the solutions (lines 14-20).

Algorithm 1 Pseudo-Code for the CNBC Model

1	Inputs: N, Up, af
2	$\{\mathbf{a}_1, \dots, \mathbf{a}_N\} \leftarrow \text{Initialize}(N)$;
3	$\mathbf{a}_B \leftarrow \text{SelectBestAgent}(\mathbf{a}_1, \dots, \mathbf{a}_N)$;
4	$k = 1$;
5	While $k \leq af$ do
6	$fitDif \leftarrow \text{CalculateFitnessDifferences}(\mathbf{a}_1, \dots, \mathbf{a}_N)$;
7	$G \leftarrow \text{GenerateGraph}(fitDif)$;
8	For each agent \mathbf{a}_i where $i = 1, \dots, N$
9	$SN_{i,B} \leftarrow \text{GetShortestPath}(G, \mathbf{a}_i, \mathbf{a}_B)$;
10	$[case, \mathbf{p}_1, \mathbf{p}_2] \leftarrow \text{GetBezierCurve}(SN_{i,B})$;
11	$ua_i(k+1) \leftarrow \text{UpdatePosition}(\mathbf{a}_i(k), \mathbf{p}_1, \mathbf{p}_2)$;
12	End for
13	$\mathbf{a}_B(k+1) \leftarrow \text{SelectBestAgent}(ua_1(k+1), \dots, ua_N(k+1))$;
14	If $J(ua(k+1)) < J(a(k))$
15	$\mathbf{a}(k+1) = ua(k+1)$;
16	else
17	If $(rand(0, 1) > Up)$
18	$\mathbf{a}(k+1) = \mathbf{a}(k)$;
19	End if
20	End else
21	$k = k + 1$;
22	End while
23	Output: \mathbf{a}_B

An example is implemented to illustrate the CNBC approach. The objective is to detect the minimum value of the two-dimensional objective function defined in equation (13), as illustrated in figure 5(a).

$$\begin{aligned}
 f(x_1, x_2) &= 3(1 - x_1)^2 e^{-(x_1^2 - x_2^2)} \\
 &\quad - 10 \left(\frac{x_1}{5} - x_1^3 - x_2^5 \right) e^{-(x_1^2 - x_2^2)} - 1/3 e^{-(x_1+1)^2 - x_2^2}
 \end{aligned} \tag{13}$$

Figure 5(b) shows the first step of CNBC initialization. Figures 5(c) and 5(d) illustrate the creation of a complex network process. The generation of trajectories is presented in figures 5(e) and 5(f), where the square elements represent the points \mathbf{p}_1 and \mathbf{p}_2 used in the trajectories. Next, the new position of each agent was calculated using Eq. 11. Finally, the selected mechanism was applied to update the agent's position. These operations are applied sequentially until the number of access functions af is reached.

5) COMPUTATIONAL COST

This subsection considers the computational cost associated with our proposed approach. Metaheuristic algorithms, by nature, encompass intricate structures entailing stochastic components. Consequently, undertaking a conventional complexity analysis for such systems becomes unviable. The execution duration of an algorithm is susceptible to the impact of various factors, rendering a traditional assessment of complexity impracticable. In light of these considerations, the standard methodology is to employ Big-O notation [59]. This notation provides an essential structure for evaluating the algorithmic cost quantifying the count of operations needed for successful execution.

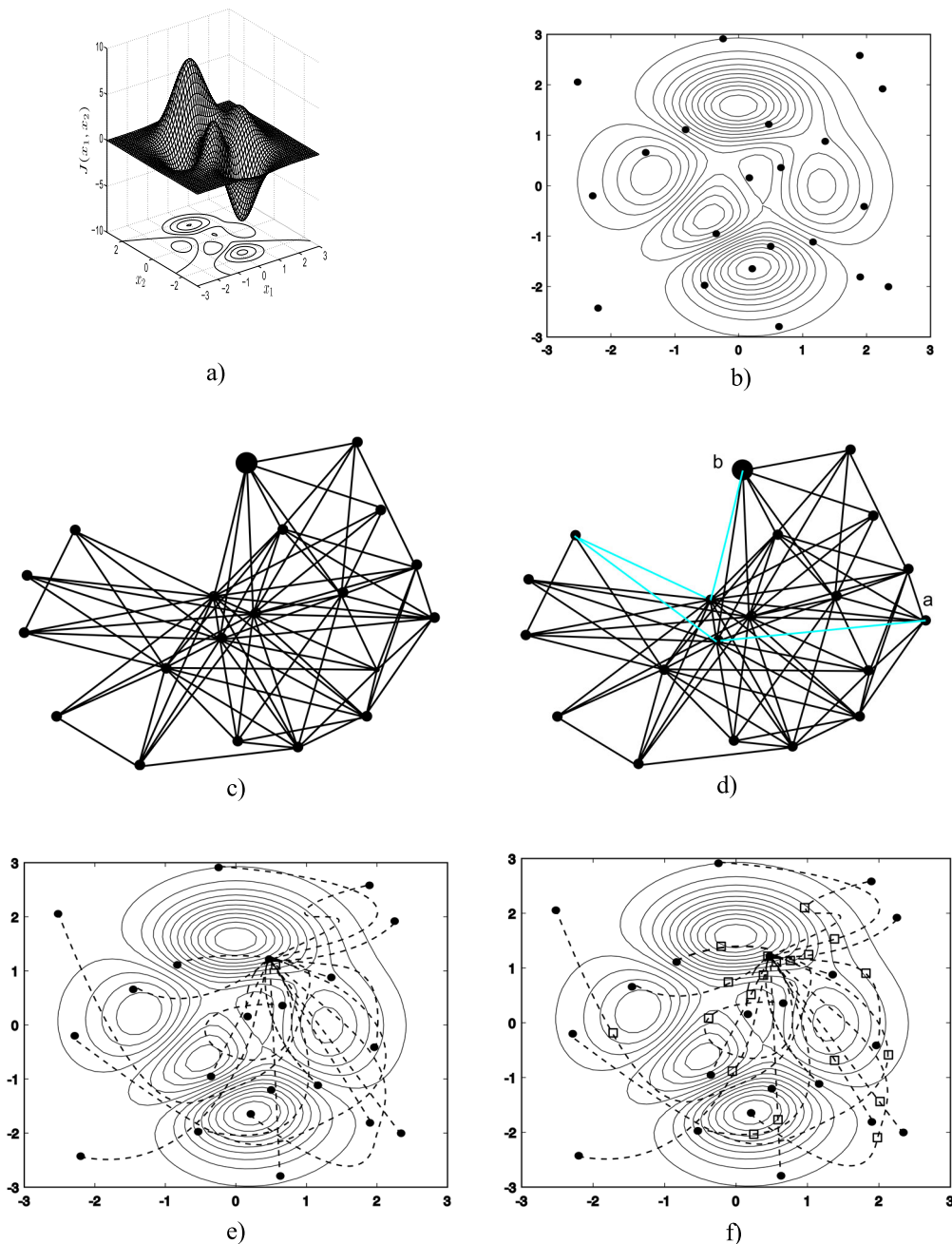


FIGURE 5. Graphical example of CNBC approach.

The evaluation of the computational cost of our method has been conducted utilizing the foundational principles of the Big-O notation introduced in [60]. Table 1 provides the partial computational costs attributed to our approach. Within the table, we adopt the assumption that N denotes the number of candidate solutions subject to algorithmic operations, while $MAXITER$ represents the upper limit of iterations within the methodology. For the sake of clarity, the table also offers a breakdown of the Algorithm 1 operations, aligning them with respective partial computational costs.

Since the steps 2-7 of Table 1 are executed $MAXITER$ iterations, the total cost $O(N)$ is formulated as follows:

$$O(N) = MAXITER \cdot [N^2 + 2 \cdot N + (N \cdot \log(N))(N + 1) + SN_{i,B} \cdot 100] \tag{14}$$

Under such conditions, the Big-O notation of the proposed method is polynomial (quadratic complexity N^2), which means that the algorithm's time complexity grows at a polynomial rate as the input size of N increases.

TABLE 1. Partial computational cost in terms of the Big-O notation attributed to our approach.

	Process	Big-O
1	$\{a_1, \dots, a_N\} \leftarrow \text{Initialize}(N)$ The N different solutions are initialized.	N
2	$a_B \leftarrow \text{SelectBestAgent}(a_1, \dots, a_N)$ Select the best solution.	$N \cdot \log(N)$
3	$fitDif \leftarrow \text{CalculateFitnessDifferences}(a_1, \dots, a_N)$ Determine the differences among all N solutions.	$N \cdot N$
4	$G \leftarrow \text{GenerateGraph}(fitDif)$ Build the graph.	N
5	$SN_{i,B} \leftarrow \text{GetShortestPath}(G, a_i, a_B)$ Complexity attributed to the Dijkstra's algorithm	$N^2 \cdot \log(N)$
6	$[case, p_1, p_2] \leftarrow \text{GetBezierCurve}(SN_{i,B})$ The time complexity of determining a Bezier curve depends on the number of control points $SN_{i,B}$ and the number of evaluation points (100).	$SN_{i,B} \cdot 100$
7	$a_i(k+1) \leftarrow \text{UpdatePosition}(a_i(k), p_1, p_2)$ Update N different solutions.	N

TABLE 2. Parameter configuration for each evolutionary method.

Algorithm	Parameter(s)	Reference
BA	The loudness is set as $A = 0.9$ and the pulse rate $r = 0.2$	[14]
CS	The discover probability is set to $pa = 0.25$	[15]
CSA	The awareness probability is set as $AP = 0.1$ and the flight length $fl = 2$	[16]
DE	The variant considered is DE/rand/bin, where crossover probability $cr = 0.5$ and differential weight $dw = 0.2$	[10]
GWO	The population size is $n = 50$	[17]
HS	The parameters considered are harmony memory size $HMS = 50$, harmony memory consideration rate $HMCR = 0.7$, and pitch adjustment rate $PAR = 0.3$	[19]
PSO	The parameters are set to $c_1 = 2$ and $c_2 = 2$ with linearly decreasing weight factor from 0.9 to 0.2	[13]
SA	The parameter beta is set to $beta = 0.98$	[18]
SCA	The implementation considers the guidelines described by the author.	[20]
SMS	The source code has been obtained by its reference.	[21]

IV. EXPERIMENTAL RESULTS

To assess the performance of the proposed method, a comprehensive set of 23 benchmark functions has been chosen. These benchmark functions represent various types of optimization problems, including unimodal, multimodal, and

hybrid functions. By applying the proposed method to these benchmark functions, we can quantitatively measure its effectiveness in finding high-quality solutions, handling different types of objective landscapes, and adapting to various problem complexities. The mathematical description of the test functions is shown in Table 17, available in Appendix A. Where n corresponds to the n -dimensional vector at which the test functions are evaluated, $f(x^*)$ represents the optimal value of a given function evaluated at position x^* , and S corresponds to the lower and upper limits of the search space.

The results obtained from the proposed approach have been compared to a set of algorithms, including classical and state-of-the-art metaheuristic methods. These algorithms have been chosen to represent a diverse range of optimization techniques. The comparison provides a comprehensive evaluation of the proposed approach's performance against well-established and widely used methods. The set of algorithms used for comparison includes the Bat Algorithm (BA) [14], Cuckoo Search Algorithm (CS) [15], Crow Search Algorithm (CSA) [16], Differential Evolution (DE) [10], Grey Wolf Optimization (GWO) [17], Harmony Search (HS) [19], Particle Swarm Optimization (PSO) [13], Simulated Annealing (SA) [18], Sine Cosine Algorithm (SCA) [20], and State of Matter Search (SMS) [21]. Each of these algorithms presents interesting characteristics in their search strategies. To ensure a comprehensive evaluation, the comparison has been conducted on each of the benchmark functions selected for the study. The evaluation considers the performance of the algorithms across various dimensions, specifically 30, 100, and 200 dimensions. Additionally, a high-dimension comparison analysis has been made between the proposed method and Dynamic Stochastic Search (DSS) [61] to evaluate the CNBC performance at 300 and 3000 dimensions. All experiments were carried out utilizing a computer equipped with an AMD Ryzen 3 processor with Radeon Vega Graphics 3.50Ghz and 8 GB of memory. Additionally, the proposed method has been programmed using MATLAB's language (M language) in the integrated development environment MATLAB R2022b.

This section is divided into four sub-sections, each addressing a specific aspect of the evaluation and analysis of the proposed approach. The first subsection (A) provides details about the setup and configuration of the test environment. In the second subsection (B), the performance of the proposed approach is evaluated and compared with other popular metaheuristic algorithms. The evaluation involves running the algorithms on the benchmark functions and collecting relevant performance metrics. Statistical techniques are employed to compare the results obtained by the proposed approach with those of the other algorithms, allowing for a quantitative assessment of its performance. The third subsection (C) validates the proposed approach's ability to converge towards an accurate solution. This analysis examines the algorithm's behavior over iterations or gener-

TABLE 3. Minimization results with $n = 30$. Benchmark functions (available on Appendix A).

		BA	CS	CSA	DE	GWO	HS	PSO	SA	SCA	SMS	CNBC
f_1	f_{Best}	1.37E-05	5.73E+02	3.71E+01	7.28E+03	1.05E-75	8.55E+00	8.23E+02	1.51E+02	9.45E-07	3.49E+02	5.72E-138
	\bar{f}	1.41E-05	5.80E+02	4.93E+01	1.74E+04	9.08E-73	1.26E+01	3.32E+03	2.32E+02	9.22E-02	3.79E+02	1.45E-101
	σ_f	1.58E-06	3.16E+01	6.01E+00	1.23E+04	2.51E-72	2.09E+00	1.66E+03	3.89E+01	6.52E-03	1.92E+01	7.78E-101
	f_{Worst}	1.62E-05	6.48E+02	5.97E+01	2.38E+03	1.26E-71	1.89E+01	6.65E+02	3.18E+02	1.83E-02	4.18E+02	4.27E-100
f_2	f_{Best}	3.08E+06	7.41E+07	2.51E+06	3.99E+01	4.29E-70	4.99E+05	2.31E+01	1.60E+07	2.43E-09	4.41E+07	7.12E-71
	\bar{f}	2.88E+06	7.46E+07	4.06E+06	6.60E+01	7.23E-68	6.90E+05	1.03E+02	2.91E+07	6.03E-05	5.00E+07	1.07E-52
	σ_f	2.10E+06	2.31E+06	7.81E+05	5.20E+01	1.90E-67	9.40E+04	4.16E+01	5.71E+06	6.33E-06	3.32E+06	5.84E-52
	f_{Worst}	7.52E+06	7.94E+07	5.16E+06	6.63E+00	9.10E-67	8.89E+05	1.89E+01	4.72E+07	1.29E-05	5.97E+07	3.20E-51
f_3	f_{Best}	1.01E+01	3.40E+04	2.05E+03	4.51E+04	8.24E-74	3.43E+02	2.13E+03	8.93E+03	4.36E+01	2.11E+04	1.63E-123
	\bar{f}	4.63E-04	3.42E+04	2.52E+03	7.93E+04	3.34E-71	5.25E+02	2.03E+04	1.40E+04	1.10E+04	2.32E+04	1.27E-93
	σ_f	3.58E+01	1.56E+03	2.83E+02	5.79E+04	8.54E-71	7.89E+01	7.96E+03	2.78E+03	3.48E+03	1.35E+03	6.95E-93
	f_{Worst}	1.72E+02	3.73E+04	3.21E+03	9.38E+03	4.54E-70	6.87E+02	3.64E+03	1.98E+04	3.42E+03	2.70E+04	3.81E-92
f_4	f_{Best}	4.35E+01	8.87E+01	4.18E+01	5.95E+01	1.04E-18	3.66E+01	1.50E+01	8.65E+01	1.20E+00	1.00E+02	2.19E-67
	\bar{f}	4.35E+01	8.85E+01	5.02E+01	7.65E+01	2.44E-17	4.08E+01	3.50E+01	9.41E+01	2.50E+01	1.00E+02	2.63E-57
	σ_f	7.79E+00	1.09E+00	3.79E+00	6.86E+01	4.66E-17	1.69E+00	2.23E+01	3.20E+00	1.15E+01	6.60E-03	1.41E-56
	f_{Worst}	5.63E+01	9.09E+01	5.63E+01	3.73E+00	2.09E-16	4.38E+01	4.30E+00	9.80E+01	6.61E+00	1.00E+02	7.75E-56
f_5	f_{Best}	1.38E+03	5.12E+01	7.06E+04	1.61E+01	8.49E-73	1.48E+04	4.11E-12	4.33E+05	8.26E-09	9.14E+05	6.29E-138
	\bar{f}	5.02E+04	7.84E+01	1.16E+05	2.30E+01	1.08E-69	2.28E+04	2.86E+03	6.10E+05	9.44E-03	9.89E+05	2.99E-101
	σ_f	3.63E+04	2.24E+01	1.71E+04	4.90E+00	1.67E-69	3.27E+03	5.33E+03	9.87E+04	3.66E-02	4.70E+04	1.29E-100
	f_{Worst}	1.39E+05	1.27E+02	1.52E+05	3.51E+01	6.17E-69	3.13E+04	2.15E+04	7.89E+05	2.01E-01	1.10E+06	6.79E-100
f_6	f_{Best}	1.76E+04	2.40E+05	1.37E+04	8.45E+03	0.00E+00	3.51E+03	7.65E+02	6.48E+04	0.00E+00	1.26E+05	0.00E+00
	\bar{f}	1.75E+04	2.43E+05	1.92E+04	1.70E+04	0.00E+00	4.83E+03	3.34E+03	9.76E+04	2.00E+00	1.45E+05	0.00E+00
	σ_f	4.72E+03	7.11E+03	2.23E+03	1.19E+04	0.00E+00	7.27E+02	1.93E+03	1.77E+04	6.67E-02	9.54E+03	0.00E+00
	f_{Worst}	3.07E+04	2.48E+05	2.27E+04	2.24E+03	0.00E+00	6.79E+03	6.78E+02	1.35E+05	3.65E-01	1.67E+05	0.00E+00
f_7	f_{Best}	1.28E+01	4.57E+02	1.79E+01	3.56E+00	7.69E+00	1.20E+01	2.16E+00	1.19E+02	3.22E-03	3.74E+02	1.59E-05
	\bar{f}	2.25E+01	6.22E+02	2.48E+01	1.57E+01	8.85E+00	1.43E+01	1.14E+01	2.19E+02	1.12E-01	4.36E+02	1.29E-04
	σ_f	7.48E+00	8.68E+01	3.10E+00	9.27E+00	5.20E-01	9.04E-01	5.38E+00	6.21E+01	2.48E-02	4.01E+01	7.94E-05
	f_{Worst}	4.33E+01	7.48E+02	2.92E+01	2.57E+00	9.54E+00	1.56E+01	2.34E+00	3.35E+02	2.21E-02	5.49E+02	2.99E-04
f_8	f_{Best}	5.99E-18	1.77E-09	2.45E-02	2.83E-08	4.90E-173	3.38E-03	0.00E+00	1.01E+00	7.21E-17	1.31E+01	0.00E+00
	\bar{f}	4.34E-17	2.18E-08	1.19E-01	1.48E-07	2.38E-164	1.11E-02	0.00E+00	5.06E+00	7.63E-07	1.59E+01	3.75E-267
	σ_f	2.10E-17	1.96E-08	4.32E-02	1.03E-07	0.00E+00	3.71E-03	0.00E+00	1.99E+00	3.27E-06	1.88E+00	0.00E+00
	f_{Worst}	1.06E-16	8.00E-08	2.07E-01	4.23E-07	5.29E-163	1.90E-02	0.00E+00	8.84E+00	1.72E-05	2.14E+01	1.12E-265
f_9	f_{Best}	7.14E-02	3.55E-01	1.45E+01	3.42E+00	7.13E-02	1.28E+00	6.05E-01	1.61E-01	2.16E-01	5.16E+01	7.13E-02
	\bar{f}	7.14E-02	4.75E-01	1.83E+01	6.50E+00	7.57E-02	1.92E+00	2.60E+00	3.86E-01	6.32E-01	6.06E+01	7.13E-02
	σ_f	3.89E-06	8.75E-02	1.37E+00	4.49E+00	3.49E-03	3.11E-01	1.52E+00	1.62E-01	3.57E-01	5.16E+00	1.62E-10
	f_{Worst}	7.14E-02	6.95E-01	2.11E+01	7.30E-01	8.31E-02	2.72E+00	6.29E-01	7.67E-01	1.12E-01	7.80E+01	7.13E-02
f_{10}	f_{Best}	7.92E+01	8.45E+02	2.16E+02	2.45E+02	1.04E-18	6.46E+01	9.56E+01	2.30E+02	5.95E-06	6.38E+02	0.00E+00
	\bar{f}	7.26E+01	8.68E+02	2.62E+02	3.06E+02	2.44E-17	9.53E+01	2.42E+02	4.06E+02	8.81E+01	6.81E+02	0.00E+00
	σ_f	3.03E+01	3.91E+01	1.22E+01	2.75E+02	4.66E-17	1.20E+01	1.74E+02	5.31E+01	2.02E+01	2.60E+01	0.00E+00
	f_{Worst}	1.52E+02	8.68E+02	2.78E+02	1.47E+01	2.09E-16	1.21E+02	3.12E+01	4.71E+02	2.75E+01	7.50E+02	0.00E+00
f_{11}	f_{Best}	1.53E+01	2.00E+01	1.64E+01	1.59E+01	7.11E-15	1.09E+01	7.60E+00	1.99E+01	4.16E-05	2.14E+01	8.88E-16
	\bar{f}	1.55E+01	2.00E+01	1.73E+01	1.77E+01	1.24E-14	1.23E+01	1.47E+01	2.05E+01	2.02E+01	2.15E+01	8.88E-16
	σ_f	1.04E+00	7.23E-15	3.72E-01	1.67E+01	2.91E-15	5.47E-01	1.13E+01	2.93E-01	1.64E+01	6.47E-02	0.00E+00
	f_{Worst}	1.68E+01	2.00E+01	1.79E+01	4.70E-01	1.42E-14	1.30E+01	1.66E+00	2.10E+01	7.50E+00	2.16E+01	8.88E-16
f_{12}	f_{Best}	7.85E+01	1.99E+03	1.12E+02	7.27E+01	0.00E+00	3.34E+01	8.75E+00	5.74E+02	2.26E-06	1.17E+03	0.00E+00
	\bar{f}	1.50E+02	2.29E+03	1.73E+02	1.79E+02	2.28E-03	4.38E+01	2.77E+01	8.70E+02	6.93E-01	1.30E+03	3.15E-03
	\bar{f}	4.97E+01	9.70E+01	2.23E+01	1.12E+02	6.04E-03	4.80E+00	1.73E+01	1.31E+02	1.75E-01	6.82E+01	1.05E-04

TABLE 3. (Continued.) Minimization results with $n = 30$. Benchmark functions (available on Appendix A).

	f_{Worst}	3.23E+02	2.36E+03	2.15E+02	2.24E+01	2.81E-02	5.29E+01	4.68E+00	1.09E+03	2.00E-01	1.44E+03	5.75E-04
f_{13}	f_{Best}	-2.41E+00	1.67E+01	1.17E+00	5.39E-02	-3.00E+00	-1.84E+00	-3.24E-02	5.24E+00	-3.00E+00	1.86E+01	-3.00E+00
	\bar{f}	-1.08E+00	2.01E+01	1.54E+00	1.44E+00	-3.00E+00	-1.54E+00	3.92E+00	7.95E+00	-3.00E+00	2.16E+01	-3.00E+00
	σ_f	5.70E-01	1.88E+00	1.98E-01	7.96E-01	0.00E+00	1.56E-01	1.71E+00	1.23E+00	-3.00E+00	1.66E+00	1.62E-08
	f_{Worst}	2.51E-01	2.43E+01	2.01E+00	3.31E-01	-3.00E+00	-1.18E+00	1.01E+00	1.04E+01	2.65E-05	2.58E+01	-3.00E+00
f_{14}	f_{Best}	2.75E+00	1.90E+02	1.53E+01	3.59E+01	2.75E+00	3.78E+00	3.34E+00	2.72E+00	2.75E+00	1.09E+04	2.72E+00
	\bar{f}	2.75E+00	4.02E+02	3.72E+01	1.27E+01	2.75E+00	4.77E+00	2.80E-00	5.67E+01	2.75E+00	1.31E+04	2.72E+00
	σ_f	4.12E-03	1.10E+02	1.13E+01	1.49E+01	1.56E-04	5.82E-01	8.86E-01	8.64E+01	2.55E-04	1.48E+03	5.59E-03
	f_{Worst}	2.77E+00	6.15E+02	6.74E+01	6.25E+01	2.75E+00	6.32E+00	7.68E+00	2.95E+02	2.75E+00	1.71E+04	2.75E+00
f_{15}	f_{Best}	9.67E-06	7.34E+01	5.16E+00	3.10E+00	1.05E-75	9.37E-01	1.13E+00	2.34E+01	2.93E-10	5.50E+01	-1.04E+00
	\bar{f}	1.27E-05	8.51E+01	7.65E+00	7.40E+00	4.80E-74	2.02E+00	3.93E+00	3.51E+01	3.03E-06	6.40E+01	1.33E-14
	σ_f	1.50E-06	4.75E+00	1.01E+00	5.05E+00	7.97E-74	4.14E-01	2.21E+00	5.69E+00	3.35E-07	3.81E+00	-3.45E-02
	f_{Worst}	1.64E-05	9.35E+01	9.32E+00	9.97E-01	3.49E-73	2.72E+00	7.48E-01	4.87E+01	7.04E-07	7.00E+01	1.89E-01
f_{16}	f_{Best}	2.71E-03	5.39E-01	1.12E+03	2.72E+02	8.93E-08	2.58E+02	3.22E-03	7.78E-13	1.29E-05	1.17E+05	1.63E-139
	\bar{f}	7.62E-03	1.05E+00	1.86E+03	4.03E+02	4.95E-06	4.84E+02	1.92E+02	3.67E+03	1.29E+00	1.48E+05	2.69E-99
	σ_f	2.47E-03	3.83E-01	4.43E+02	6.67E+01	7.17E-06	1.00E+02	1.18E+02	3.94E+03	5.24E+00	2.26E+04	1.39E-98
	f_{Worst}	1.50E-02	1.98E+00	2.75E+03	5.47E+02	3.17E-05	6.63E+02	5.42E+02	1.29E+04	2.87E+01	2.18E+05	7.61E-98
f_{17}	f_{Best}	1.59E+01	7.69E+02	3.61E+01	1.36E+06	6.37E-01	7.10E+00	1.31E+01	1.51E+02	3.03E-01	5.46E+02	1.57E-32
	\bar{f}	1.36E+01	7.62E+02	5.74E+01	4.05E+07	1.01E+00	1.18E+01	3.42E+03	3.01E+02	1.11E+01	6.22E+02	1.12E-09
	σ_f	8.27E+00	7.47E+01	9.22E+00	1.64E+07	1.55E-01	1.40E+00	3.69E+02	7.21E+01	1.20E+00	4.95E+01	4.17E-11
	f_{Worst}	5.04E+01	8.83E+02	7.30E+01	1.08E+07	1.27E+00	1.44E+01	9.05E+02	4.47E+02	1.94E+00	7.74E+02	2.05E-10
f_{18}	f_{Best}	7.66E+06	6.76E+09	4.13E+07	7.68E+06	1.09E+02	4.65E+06	6.98E+01	1.19E+09	2.10E+00	3.27E+09	1.35E-32
	\bar{f}	4.90E+06	6.88E+09	7.69E+07	1.02E+08	9.79E+04	8.44E+06	2.83E+05	1.99E+09	5.04E+00	3.73E+09	1.56E-11
	σ_f	9.05E+06	5.33E+08	2.13E+07	4.89E+07	6.30E+04	2.14E+06	6.62E+04	5.03E+08	2.73E+00	2.09E+08	5.20E-13
	f_{Worst}	3.19E+07	7.87E+09	1.23E+08	1.87E+07	2.14E+05	1.34E+07	7.98E+04	3.04E+09	5.80E-01	4.16E+09	2.84E-12
f_{19}	f_{Best}	2.92E+04	1.15E+28	1.48E+04	5.22E+03	5.81E-42	3.12E+03	1.21E+03	7.06E+04	1.09E-06	1.43E+20	7.86E-71
	\bar{f}	2.48E+04	2.45E+26	2.04E+04	1.89E+04	1.31E-40	4.44E+03	2.23E+04	1.02E+05	8.34E-03	1.26E+23	3.36E-57
	σ_f	1.79E+04	1.75E+28	3.10E+03	1.35E+04	1.23E-40	7.13E+02	4.79E+03	1.90E+04	1.77E-03	4.92E+23	1.66E-56
	f_{Worst}	7.46E+04	4.69E+28	2.53E+04	3.10E+03	4.60E-40	5.83E+03	4.07E+03	1.43E+05	2.11E-03	2.65E+24	9.10E-56
f_{20}	f_{Best}	4.95E+02	3.50E+03	5.52E+02	5.15E+02	2.90E+01	2.96E+02	2.33E+02	2.01E+03	2.90E+01	1.43E+20	2.90E+01
	\bar{f}	4.80E+02	3.55E+03	7.26E+02	7.81E+02	2.90E+01	3.40E+02	4.66E+02	2.85E+03	2.90E+01	1.26E+23	2.90E+01
	σ_f	1.45E+02	9.54E+01	5.44E+01	6.30E+02	5.32E-05	2.41E+01	3.54E+02	3.71E+02	4.52E-03	4.92E+23	6.18E-05
	f_{Worst}	9.75E+02	3.57E+03	8.31E+02	5.96E+01	2.90E+01	4.02E+02	5.38E+01	3.43E+03	2.90E+01	2.65E+24	2.90E+01
f_{21}	f_{Best}	2.48E+07	7.36E+09	1.97E+07	4.96E+07	3.20E+01	3.92E+06	1.12E+05	1.03E+09	3.13E+02	3.24E+09	3.20E+01
	\bar{f}	1.64E+07	7.35E+09	6.50E+07	1.72E+07	3.20E+01	6.55E+06	1.64E+05	2.11E+09	6.28E+02	3.77E+09	4.20E+01
	σ_f	2.64E+07	4.48E+08	2.33E+07	2.10E+07	1.05E-05	1.83E+06	1.84E+03	5.03E+08	3.20E+01	3.08E+08	3.07E+02
	f_{Worst}	1.14E+08	8.36E+09	1.22E+08	7.77E+07	3.20E+01	1.03E+07	6.82E+05	3.19E+09	3.55E+03	4.67E+09	5.02E+01
f_{22}	f_{Best}	2.64E+03	1.54E+28	6.89E+02	5.45E+02	2.90E+01	2.98E+02	3.00E+02	1.80E+03	2.90E+01	4.36E+20	2.90E+01
	\bar{f}	1.10E+03	2.81E+27	9.13E+02	9.86E+02	2.90E+01	3.38E+02	2.97E+03	3.02E+03	2.90E+01	6.00E+27	2.90E+01
	σ_f	5.91E+03	1.78E+28	1.29E+02	7.46E+02	1.00E-14	2.48E+01	7.10E+02	4.87E+02	3.79E-03	3.29E+28	0.00E+00
	f_{Worst}	2.63E+04	4.43E+28	1.35E+03	1.07E+02	2.90E+01	3.77E+02	5.79E+02	3.95E+03	2.90E+01	1.80E+29	2.90E+01
f_{23}	f_{Best}	1.09E+03	5.06E+25	4.28E+06	1.87E+06	-7.54E+01	5.26E+04	1.40E+03	5.87E+08	-3.23E+01	2.18E+20	-8.10E+01
	\bar{f}	4.39E+07	7.67E+27	1.82E+07	2.82E+07	-6.68E+01	4.93E+05	1.56E+06	1.13E+09	5.25E+02	2.15E+22	-8.10E+01
	σ_f	1.16E+08	1.46E+28	1.02E+07	9.86E+06	5.30E+00	4.16E+05	5.81E+04	3.09E+08	6.67E+00	3.68E+22	1.08E-04
	f_{Worst}	6.02E+08	5.62E+28	3.88E+07	6.42E+06	-5.56E+01	2.16E+06	2.83E+05	1.68E+09	9.99E+01	1.80E+23	-8.10E+01

ations, tracking the convergence of the objective function values. By analyzing the convergence patterns, the effec-

tiveness of the proposed approach in reaching optimal or near-optimal solutions is assessed. In the final subsection (D),

TABLE 4. Minimization results with $n = 100$. Benchmark functions (available on Appendix A).

		BA	CS	CSA	DE	GWO	HS	PSO	SA	SCA	SMS	CNBC
f_1	f_{Best}	1.83E-04	4.95E+00	2.62E+02	3.03E+02	1.14E-38	2.25E+02	1.35E-01	7.26E+02	2.23E+02	1.03E+03	1.27E-139
	\bar{f}	1.97E-02	7.04E+00	2.83E+02	3.41E+02	4.61E-37	2.54E+02	5.65E+00	8.62E+02	1.25E+04	1.07E+03	2.33E-100
	σ_f	1.06E-01	1.10E+00	9.84E+00	2.07E+01	6.40E-37	1.28E+01	1.06E+01	8.02E+01	3.39E+03	2.93E+01	1.28E-99
	f_{Worst}	5.83E-01	9.10E+00	3.07E+02	3.82E+02	3.34E-36	2.72E+02	2.66E+01	1.05E+03	3.17E+03	1.16E+03	6.99E-99
f_2	f_{Best}	2.61E+07	4.41E+06	2.30E+08	1.44E+08	3.88E-32	1.91E+08	1.07E+07	8.64E+08	9.23E-03	1.36E+09	6.09E-69
	\bar{f}	1.83E+08	5.66E+06	2.83E+08	1.68E+08	3.63E-31	2.22E+08	3.98E+07	1.06E+09	6.47E+00	1.47E+09	5.30E-56
	σ_f	7.43E+07	6.93E+05	2.20E+07	1.08E+07	4.20E-31	1.73E+07	1.93E+07	1.35E+08	8.98E-01	5.51E+07	1.76E-55
	f_{Worst}	3.32E+08	7.53E+06	3.14E+08	1.99E+08	1.73E-30	2.46E+08	8.52E+07	1.43E+09	1.36E+00	1.55E+09	8.04E-55
f_3	f_{Best}	1.05E-01	7.35E+02	4.50E+04	3.79E+04	5.44E-36	3.53E+04	5.50E+02	1.37E+05	9.75E+04	1.98E+05	2.65E-138
	\bar{f}	3.29E+03	1.11E+03	4.97E+04	4.44E+04	8.04E-35	4.12E+04	4.77E+03	1.59E+05	2.49E+05	2.15E+05	8.59E-94
	σ_f	3.67E+03	1.43E+02	2.95E+03	3.21E+03	6.39E-35	2.74E+03	2.93E+03	1.47E+04	1.69E+05	9.05E+03	4.70E-93
	f_{Worst}	1.73E+04	1.41E+03	5.41E+04	5.09E+04	2.53E-34	4.59E+04	1.16E+04	1.92E+05	4.46E+04	2.39E+05	2.58E-92
f_4	f_{Best}	4.01E+01	2.29E+01	6.49E+01	9.24E+01	1.28E-06	7.54E+01	5.18E+01	9.34E+01	7.66E+01	1.00E+02	1.55E-70
	\bar{f}	5.44E+01	2.70E+01	7.15E+01	9.52E+01	2.91E-04	7.86E+01	6.27E+01	9.67E+01	9.11E+01	1.00E+02	3.05E-55
	σ_f	8.03E+00	2.23E+00	2.24E+00	1.14E+00	1.09E-03	1.19E+00	5.18E+00	1.19E+00	8.50E+01	0.00E+00	1.22E-54
	f_{Worst}	6.89E+01	3.15E+01	7.51E+01	9.68E+01	6.04E-03	8.03E+01	7.26E+01	9.85E+01	3.29E+00	1.00E+02	6.10E-54
f_5	f_{Best}	2.23E+05	3.48E+04	1.94E+06	1.63E+06	2.30E-34	1.51E+06	8.59E+03	5.51E+06	7.45E+03	8.65E+06	4.62E-134
	\bar{f}	1.02E+06	5.25E+04	2.17E+06	1.92E+06	2.20E-33	1.79E+06	2.14E+05	6.90E+06	5.69E+04	9.10E+06	6.88E-93
	σ_f	3.41E+05	9.99E+03	1.47E+05	1.25E+05	1.86E-33	1.18E+05	1.23E+05	6.24E+05	3.73E+04	2.34E+05	3.77E-92
	f_{Worst}	1.59E+06	8.60E+04	2.42E+06	2.13E+06	7.70E-33	1.96E+06	5.20E+05	7.70E+06	1.38E+05	9.51E+06	2.06E-91
f_6	f_{Best}	4.29E+04	2.16E+03	9.55E+04	1.16E+05	0.00E+00	8.14E+04	1.24E+02	2.67E+05	4.81E+02	3.89E+05	0.00E+00
	\bar{f}	6.81E+04	3.16E+03	1.06E+05	1.30E+05	1.43E+00	9.58E+04	2.04E+03	3.28E+05	1.18E+04	4.05E+05	0.00E+00
	σ_f	1.68E+04	4.31E+02	5.37E+03	6.98E+03	1.07E+00	5.25E+03	3.75E+03	3.15E+04	4.20E+03	1.07E+04	0.00E+00
	f_{Worst}	1.01E+05	3.91E+03	1.18E+05	1.43E+05	4.00E+00	1.05E+05	1.04E+04	3.98E+05	3.19E+03	4.34E+05	0.00E+00
f_7	f_{Best}	9.77E+01	4.30E+01	3.27E+02	6.07E+02	3.73E+01	3.18E+02	5.48E+01	1.66E+03	4.70E+00	3.56E+03	7.62E-06
	\bar{f}	1.96E+02	4.65E+01	4.07E+02	7.28E+02	3.95E+01	3.94E+02	7.44E+01	2.55E+03	1.13E+02	3.84E+03	1.26E-04
	σ_f	8.67E+01	1.59E+00	4.67E+01	5.36E+01	9.79E-01	3.39E+01	2.12E+01	4.23E+02	4.86E+01	1.60E+02	9.97E-05
	f_{Worst}	4.80E+02	4.95E+01	5.08E+02	8.42E+02	4.14E+01	4.59E+02	1.50E+02	3.34E+03	3.03E+01	4.22E+03	3.96E-04
f_8	f_{Best}	1.65E-14	9.58E-05	1.53E+00	7.25E+00	1.11E-85	1.83E+00	0.00E+00	8.75E+00	3.00E-01	3.59E+01	0.00E+00
	\bar{f}	3.54E-14	2.55E-04	2.09E+00	1.01E+01	1.56E-81	2.49E+00	0.00E+00	1.77E+01	1.00E+00	4.08E+01	2.24E-275
	σ_f	1.24E-14	1.41E-04	2.82E-01	1.28E+00	3.25E-81	3.00E-01	0.00E+00	4.59E+00	5.63E-01	3.33E+00	0.00E+00
	f_{Worst}	6.74E-14	6.60E-04	2.65E+00	1.19E+01	1.17E-80	2.94E+00	0.00E+00	3.03E+01	2.47E+00	5.08E+01	6.63E-274
f_9	f_{Best}	2.08E-02	2.42E+00	8.48E+01	3.32E-02	2.46E-02	3.45E+01	2.00E+01	5.96E-02	9.74E-02	1.72E+02	2.04E-02
	\bar{f}	2.08E-02	5.31E+00	9.11E+01	3.51E-02	2.62E-02	3.91E+01	2.91E+01	1.43E-01	4.80E-01	1.82E+02	2.04E-02
	σ_f	3.44E-05	1.42E+00	2.11E+00	1.15E-03	9.95E-04	1.65E+00	6.30E+00	4.49E-02	1.87E-01	7.10E+00	1.45E-10
	f_{Worst}	2.09E-02	7.82E+00	9.38E+01	3.69E-02	2.81E-02	4.15E+01	4.00E+01	2.58E-01	7.26E-02	1.94E+02	2.04E-02
f_{10}	f_{Best}	1.81E+02	5.33E+02	1.06E+03	1.08E+03	0.00E+00	8.40E+02	3.61E+02	1.13E+03	1.05E+01	2.01E+03	0.00E+00
	\bar{f}	2.57E+02	6.27E+02	1.13E+03	1.12E+03	1.14E-13	8.87E+02	4.69E+02	1.49E+03	5.13E+02	2.08E+03	0.00E+00
	σ_f	4.81E+01	3.76E+01	2.88E+01	2.12E+01	1.12E-13	2.36E+01	6.12E+01	1.47E+02	2.08E+02	5.19E+01	0.00E+00
	f_{Worst}	3.55E+02	6.88E+02	1.19E+03	1.16E+03	4.55E-13	9.34E+02	5.66E+02	1.73E+03	1.05E+02	2.23E+03	0.00E+00
f_{11}	f_{Best}	1.42E+01	9.36E+00	1.85E+01	1.93E+01	6.39E-14	1.83E+01	3.19E+00	2.02E+01	6.52E+00	2.13E+01	8.88E-16
	\bar{f}	1.59E+01	1.11E+01	1.88E+01	1.94E+01	6.87E-14	1.86E+01	5.84E+00	2.07E+01	2.06E+01	2.13E+01	8.88E-16
	σ_f	8.52E-01	1.31E+00	1.43E-01	8.27E-02	4.60E-15	1.24E-01	2.84E+00	1.97E-01	1.85E+01	4.70E-02	0.00E+00
	f_{Worst}	1.81E+01	1.44E+01	1.91E+01	1.96E+01	7.82E-14	1.88E+01	1.31E+01	2.10E+01	4.63E+00	2.15E+01	8.88E-16
f_{12}	f_{Best}	4.11E+02	2.16E+01	8.47E+02	1.05E+03	0.00E+00	7.53E+02	1.30E+00	2.72E+03	6.96E+00	3.42E+03	0.00E+00
	\bar{f}	5.84E+02	2.65E+01	9.61E+02	1.18E+03	1.44E-03	8.59E+02	2.95E+01	3.03E+03	1.31E+02	3.65E+03	0.00E+00
	\bar{f}	1.48E+02	4.14E+00	5.92E+01	5.85E+01	4.61E-03	4.47E+01	4.83E+01	2.21E+02	4.14E+01	1.29E+02	0.00E+00
	f_{Worst}	1.07E+03	3.75E+01	1.06E+03	1.27E+03	2.02E-02	9.15E+02	1.82E+02	3.56E+03	3.45E+01	3.94E+03	0.00E+00
f_{13}	f_{Best}	-5.86E+00	-5.53E+00	9.24E+00	1.16E+01	-1.00E+01	6.56E+00	-8.29E+00	2.21E+01	-9.97E+00	5.84E+01	-1.00E+01

TABLE 4. (Continued.) Minimization results with $n = 100$. Benchmark functions (available on Appendix A).

	\bar{f}	-3.43E+00	-3.77E+00	1.00E+01	1.36E+01	-1.00E+01	7.57E+00	-7.12E+00	3.04E+01	-7.39E+00	6.24E+01	-1.00E+01
	σ_f	1.42E+00	5.28E-01	5.21E-01	7.38E-01	0.00E+00	5.66E-01	7.30E-01	3.08E+00	-9.09E+00	2.64E+00	0.00E+00
	f_{Worst}	-2.51E-01	-2.93E+00	1.11E+01	1.47E+01	-1.00E+01	8.58E+00	-5.28E+00	3.48E+01	8.30E-01	6.80E+01	-1.00E+01
f_{14}	f_{Best}	2.73E+00	5.58E+00	4.05E+02	5.15E+02	2.73E+00	2.38E+02	5.62E+01	2.72E+00	5.19E+00	2.76E+04	2.72E+00
	\bar{f}	2.73E+00	7.17E+00	6.71E+02	7.81E+02	2.73E+00	4.48E+02	2.58E+02	4.10E+02	2.42E+02	3.26E+04	2.72E+00
	σ_f	1.62E-07	8.61E-01	1.35E+02	1.25E+02	1.27E-05	6.75E+01	1.25E+02	6.17E+02	5.77E+01	2.74E+03	1.22E-10
	f_{Worst}	2.73E+00	8.95E+00	1.03E+03	1.04E+03	2.73E+00	5.80E+02	5.80E+02	2.12E+03	6.60E+01	4.08E+04	2.72E+00
f_{15}	f_{Best}	1.59E-04	6.02E-01	3.57E+01	4.78E+01	9.51E-39	3.59E+01	1.16E-02	1.06E+02	1.11E-01	1.69E+02	-1.02E+00
	\bar{f}	1.96E-04	1.08E+00	4.24E+01	5.33E+01	8.20E-38	3.95E+01	4.55E-01	1.30E+02	5.85E+00	1.84E+02	-8.23E-01
	σ_f	1.45E-05	1.78E-01	2.85E+00	2.58E+00	5.78E-38	1.67E+00	1.21E+00	1.17E+01	1.47E+00	7.67E+00	3.80E-01
	f_{Worst}	2.34E-04	1.44E+00	4.56E+01	5.81E+01	2.00E-37	4.25E+01	4.06E+00	1.59E+02	1.42E+00	1.98E+02	9.90E-117
f_{16}	f_{Best}	7.88E-02	1.83E+02	1.29E+04	3.91E+04	3.32E-07	1.90E+04	3.71E+02	3.12E-02	7.67E+00	3.03E+05	6.60E-128
	\bar{f}	1.02E+01	2.58E+02	1.92E+04	4.80E+04	7.57E-06	2.58E+04	5.51E+03	1.25E+04	1.82E+03	3.44E+05	5.64E-103
	σ_f	1.88E+01	5.61E+01	2.64E+03	3.77E+03	8.10E-06	3.11E+03	3.48E+03	1.32E+04	1.82E+03	3.45E+04	2.98E-102
	f_{Worst}	7.09E+01	4.05E+02	2.37E+04	5.52E+04	3.75E-05	3.23E+04	1.33E+04	6.50E+04	9.32E+03	4.75E+05	1.63E-101
f_{17}	f_{Best}	2.58E+01	4.28E+01	3.20E+02	5.87E+02	5.49E+00	2.60E+02	3.47E+01	9.51E+02	2.51E+07	1.57E+03	8.56E-01
	\bar{f}	4.54E+01	6.85E+01	3.66E+02	6.60E+02	6.10E+00	2.94E+02	6.08E+01	1.14E+03	3.11E+08	1.73E+03	1.16E+00
	σ_f	1.40E+01	1.30E+01	1.89E+01	3.56E+01	3.80E-01	1.39E+01	1.52E+01	1.21E+02	1.16E+08	9.43E+01	1.12E-01
	f_{Worst}	7.59E+01	9.75E+01	3.95E+02	7.40E+02	6.96E+00	3.21E+02	1.10E+02	1.43E+03	6.55E+07	1.91E+03	1.33E+00
f_{18}	f_{Best}	6.64E+05	3.22E+06	7.30E+08	1.45E+09	1.75E+06	7.94E+08	4.77E+05	5.28E+09	4.98E+07	9.45E+09	2.96E-10
	\bar{f}	1.12E+08	3.88E+06	8.65E+08	2.56E+09	2.22E+06	9.64E+08	1.52E+06	7.18E+09	5.70E+08	1.03E+10	3.42E-08
	σ_f	1.34E+08	3.89E+05	8.46E+07	2.99E+08	2.90E+05	7.60E+07	9.45E+05	9.12E+08	2.48E+08	5.56E+08	3.83E-08
	f_{Worst}	6.51E+08	4.56E+06	1.07E+09	3.08E+09	2.80E+06	1.07E+09	5.28E+06	9.46E+09	1.23E+08	1.16E+10	1.32E-07
f_{19}	f_{Best}	3.62E+04	1.00E+10	1.40E+13	2.10E+05	9.27E-21	8.11E+04	2.06E+04	2.77E+05	2.14E+02	4.57E+62	8.03E-73
	\bar{f}	1.19E+24	1.00E+10	2.13E+19	2.40E+05	2.08E-20	9.89E+04	6.30E+04	3.37E+05	1.23E+04	2.23E+67	4.82E-54
	σ_f	4.82E+24	0.00E+00	7.31E+19	2.34E+04	8.11E-21	6.74E+03	4.10E+04	2.06E+04	4.57E+03	4.70E+67	2.62E-53
	f_{Worst}	2.41E+25	1.00E+10	3.87E+20	3.08E+05	4.19E-20	1.10E+05	1.67E+05	3.72E+05	3.31E+03	2.10E+68	1.43E-52
f_{20}	f_{Best}	1.01E+03	8.79E+02	3.08E+03	4.68E+03	9.90E+01	3.11E+03	4.57E+02	8.03E+03	1.41E+02	1.22E+04	9.90E+01
	\bar{f}	2.24E+03	9.48E+02	3.64E+03	5.12E+03	9.90E+01	3.44E+03	6.09E+02	9.41E+03	9.59E+02	1.29E+04	9.90E+01
	σ_f	4.99E+02	3.23E+01	1.62E+02	2.21E+02	4.21E-04	1.18E+02	1.80E+02	9.42E+02	5.03E+02	5.01E+02	1.72E-04
	f_{Worst}	3.22E+03	1.02E+03	3.90E+03	5.48E+03	9.90E+01	3.60E+03	1.29E+03	1.14E+04	2.06E+02	1.43E+04	9.90E+01
f_{21}	f_{Best}	9.47E+05	1.66E+04	5.36E+08	1.85E+09	1.09E+02	7.99E+08	1.99E+04	5.32E+09	3.61E+07	9.32E+09	1.09E+02
	\bar{f}	1.94E+08	6.09E+04	8.51E+08	2.52E+09	1.09E+02	9.32E+08	1.97E+05	7.07E+09	7.69E+08	1.02E+10	1.09E+02
	σ_f	1.67E+08	3.45E+04	1.36E+08	2.81E+08	6.37E-05	9.22E+07	1.47E+05	9.55E+08	2.56E+08	5.28E+08	2.18E-06
	f_{Worst}	5.36E+08	1.42E+05	1.03E+09	3.04E+09	1.09E+02	1.17E+09	6.21E+05	9.33E+09	1.47E+08	1.14E+10	1.09E+02
f_{22}	f_{Best}	2.44E+03	1.00E+10	7.51E+11	6.29E+03	9.90E+01	3.12E+03	2.68E+03	9.10E+03	1.26E+02	6.04E+63	9.90E+01
	\bar{f}	7.13E+20	1.00E+10	6.90E+18	1.14E+04	9.90E+01	3.63E+03	4.31E+03	1.08E+04	1.02E+03	5.27E+67	9.90E+01
	σ_f	3.54E+21	0.00E+00	3.35E+19	1.56E+04	4.34E-14	1.83E+02	7.26E+02	1.08E+03	3.66E+02	1.91E+68	0.00E+00
	f_{Worst}	1.93E+22	1.00E+10	1.84E+20	9.37E+04	9.90E+01	4.09E+03	5.78E+03	1.33E+04	2.31E+02	1.05E+69	9.90E+01
f_{23}	f_{Best}	1.47E+07	1.00E+10	2.94E+10	2.36E+09	-1.67E+02	3.81E+08	6.85E+07	3.00E+09	4.90E+05	6.35E+63	-2.87E+02
	\bar{f}	6.95E+18	1.00E+10	1.09E+18	3.56E+09	-1.46E+02	5.09E+08	8.24E+08	4.23E+09	1.43E+08	1.24E+69	-2.87E+02
	σ_f	3.80E+19	0.00E+00	2.81E+18	4.20E+08	1.28E+01	7.35E+07	4.06E+08	7.24E+08	2.67E+07	5.62E+69	4.08E-06
	f_{Worst}	2.08E+20	1.00E+10	1.08E+19	4.58E+09	-1.22E+02	6.47E+08	1.65E+09	5.69E+09	3.13E+07	3.08E+70	-2.87E+02

a comparative analysis was carried out over three engineering design problems to evaluate the performance of the proposed method over real-world applications.

A. CONFIGURATION OF THE TEST ENVIRONMENT

In this study, a set of 23 benchmark functions has been employed to evaluate and compare the performance of the

TABLE 5. Minimization results with $n = 200$. Benchmark functions (available on Appendix A).

		BA	CS	CSA	DE	GWO	HS	PSO	SA	SCA	SMS	CNBC
f_1	f_{Best}	2.62E+01	8.91E-04	5.85E+02	1.35E+03	2.02E-26	7.01E+02	3.12E+01	1.51E+03	1.02E+04	1.93E+03	1.10E-131
	\bar{f}	3.19E+01	3.95E+00	6.49E+02	1.43E+03	6.55E-26	7.76E+02	7.88E+01	1.72E+03	2.59E+04	2.03E+03	2.51E-104
	σ_f	3.63E+00	8.29E+00	2.25E+01	3.47E+01	4.29E-26	2.45E+01	3.30E+01	1.15E+02	1.03E+04	4.86E+01	1.22E-103
	f_{Worst}	4.16E+01	3.19E+01	6.86E+02	1.50E+03	1.78E-25	8.18E+02	1.43E+02	2.03E+03	5.07E+04	2.15E+03	6.64E-103
f_2	f_{Best}	1.01E+08	5.18E+08	2.63E+09	4.31E+09	5.54E-20	2.84E+09	2.96E+08	7.52E+09	1.25E+00	1.05E+10	1.08E-66
	\bar{f}	1.21E+08	1.62E+09	2.96E+09	4.67E+09	1.75E-19	3.01E+09	5.32E+08	8.77E+09	1.30E+01	1.09E+10	5.12E-54
	σ_f	1.51E+07	4.52E+08	1.59E+08	1.40E+08	1.07E-19	1.13E+08	1.34E+08	6.89E+08	9.73E+00	2.35E+08	2.07E-53
	f_{Worst}	1.58E+08	2.30E+09	3.26E+09	4.91E+09	5.47E-19	3.30E+09	8.78E+08	9.94E+09	4.29E+01	1.12E+10	1.09E-52
f_3	f_{Best}	8.24E+03	2.46E+02	2.15E+05	4.52E+05	3.28E-24	2.53E+05	1.87E+04	5.61E+05	5.74E+05	7.64E+05	1.05E-125
	\bar{f}	1.06E+04	2.78E+04	2.35E+05	4.67E+05	1.86E-23	2.69E+05	4.27E+04	6.43E+05	8.36E+05	8.05E+05	3.98E-100
	σ_f	1.29E+03	2.36E+04	9.23E+03	1.13E+04	1.20E-23	8.94E+03	1.22E+04	4.66E+04	1.57E+05	2.85E+04	1.52E-99
	f_{Worst}	1.30E+04	8.90E+04	2.49E+05	5.02E+05	4.83E-23	2.82E+05	6.97E+04	7.66E+05	1.14E+06	8.93E+05	6.34E-99
f_4	f_{Best}	2.97E+01	4.51E+01	7.59E+01	9.62E+01	2.38E-01	8.60E+01	8.39E+01	9.58E+01	9.29E+01	1.00E+02	5.10E-70
	\bar{f}	3.25E+01	6.17E+01	7.99E+01	9.78E+01	3.57E+00	8.87E+01	9.58E+01	9.72E+01	9.50E+01	1.00E+02	2.87E-52
	σ_f	1.64E+00	8.30E+00	1.51E+00	6.63E-01	2.66E+00	7.12E-01	2.98E+00	5.94E-01	1.03E+00	0.00E+00	1.53E-51
	f_{Worst}	3.66E+01	7.90E+01	8.21E+01	9.87E+01	1.13E+01	8.96E+01	9.82E+01	9.83E+01	9.69E+01	1.00E+02	8.41E-51
f_5	f_{Best}	3.43E+05	8.57E+05	9.27E+06	1.83E+07	1.45E-22	1.03E+07	9.76E+05	2.38E+07	1.72E+05	3.22E+07	1.87E-136
	\bar{f}	4.59E+05	4.50E+06	1.03E+07	1.97E+07	6.93E-22	1.15E+07	1.95E+06	2.80E+07	1.02E+06	3.36E+07	7.70E-96
	σ_f	5.94E+04	1.62E+06	4.19E+05	5.34E+05	4.63E-22	3.75E+05	5.10E+05	1.83E+06	5.20E+05	9.56E+05	4.22E-95
	f_{Worst}	6.28E+05	8.49E+06	1.11E+07	2.04E+07	1.86E-21	1.21E+07	2.89E+06	3.20E+07	2.23E+06	3.58E+07	2.31E-94
f_6	f_{Best}	1.04E+04	9.02E+04	2.32E+05	5.08E+05	2.00E+00	2.73E+05	1.51E+04	5.80E+05	8.32E+03	7.39E+05	0.00E+00
	\bar{f}	1.31E+04	1.39E+05	2.47E+05	5.42E+05	7.07E+00	2.97E+05	3.49E+04	6.55E+05	2.73E+04	7.77E+05	0.00E+00
	σ_f	1.17E+03	3.29E+04	6.58E+03	1.11E+04	4.17E+00	8.10E+03	1.31E+04	3.46E+04	1.53E+04	2.34E+04	0.00E+00
	f_{Worst}	1.59E+04	2.40E+05	2.65E+05	5.55E+05	1.90E+01	3.12E+05	6.24E+04	7.22E+05	6.71E+04	8.61E+05	0.00E+00
f_7	f_{Best}	1.00E+02	3.73E+02	1.57E+03	6.79E+03	8.22E+01	2.38E+03	2.17E+02	7.58E+03	4.84E+02	1.32E+04	1.06E-05
	\bar{f}	1.07E+02	7.75E+02	2.04E+03	7.87E+03	8.59E+01	2.97E+03	3.35E+02	1.07E+04	9.02E+02	1.41E+04	1.32E-04
	σ_f	3.28E+00	3.58E+02	1.67E+02	4.08E+02	1.67E+00	1.74E+02	8.65E+01	1.35E+03	2.41E+02	3.89E+02	1.17E-04
	f_{Worst}	1.15E+02	2.22E+03	2.40E+03	8.65E+03	8.87E+01	3.18E+03	5.25E+02	1.37E+04	1.56E+03	1.50E+04	5.13E-04
f_8	f_{Best}	9.43E-04	9.16E-13	4.85E+00	3.07E+01	1.20E-61	1.07E+01	0.00E+00	2.56E+01	2.60E+00	7.06E+01	0.00E+00
	\bar{f}	2.35E-03	1.49E-12	6.34E+00	4.10E+01	6.07E-58	1.21E+01	0.00E+00	3.81E+01	6.34E+00	7.46E+01	1.52E-243
	σ_f	8.54E-04	3.49E-13	6.33E-01	3.11E+00	2.05E-57	5.53E-01	0.00E+00	5.87E+00	2.26E+00	3.32E+00	0.00E+00
	f_{Worst}	4.12E-03	2.35E-12	7.43E+00	4.70E+01	1.08E-56	1.32E+01	0.00E+00	5.04E+01	1.11E+01	8.30E+01	4.56E-242
f_9	f_{Best}	8.82E+01	-4.76E+11	1.88E+02	2.05E+00	2.02E+00	1.19E+02	1.08E+02	2.17E+00	9.35E+01	2.32E+02	1.01E-02
	\bar{f}	3.72E+01	1.24E-02	1.96E+02	1.04E+00	1.44E-02	1.14E+02	8.40E+01	1.38E-02	4.91E+02	3.29E+02	1.01E-02
	σ_f	4.63E+01	1.32E-02	4.93E+00	1.72E+00	1.52E-02	1.20E+02	1.05E+02	8.13E-02	2.10E+02	3.49E+02	4.27E-11
	f_{Worst}	4.10E+00	3.70E-04	2.06E+02	3.22E-01	4.95E-04	3.24E+00	1.16E+01	2.99E-02	8.79E+02	1.06E+01	1.01E-02
f_{10}	f_{Best}	1.38E+03	2.42E+02	2.30E+03	2.89E+03	2.27E-13	2.23E+03	1.05E+03	2.36E+03	2.06E+01	3.89E+03	0.00E+00
	\bar{f}	1.51E+03	4.53E+02	2.44E+03	2.99E+03	1.09E+00	2.32E+03	1.19E+03	2.98E+03	2.07E+01	4.03E+03	0.00E+00
	σ_f	5.66E+01	1.00E+02	3.89E+01	4.26E+01	2.91E+00	3.51E+01	7.83E+01	2.66E+02	3.46E-02	6.66E+01	0.00E+00
	f_{Worst}	1.62E+03	7.19E+02	2.50E+03	3.10E+03	1.13E+01	2.38E+03	1.37E+03	3.51E+03	2.07E+01	4.17E+03	0.00E+00
f_{11}	f_{Best}	9.89E+00	1.39E+01	1.91E+01	1.98E+01	4.30E-13	1.95E+01	1.18E+01	1.96E+01	3.42E+01	2.12E+01	8.88E-16
	\bar{f}	1.24E+01	1.58E+01	1.93E+01	1.98E+01	6.50E-13	1.97E+01	1.43E+01	2.05E+01	2.39E+02	2.13E+01	8.88E-16
	σ_f	1.13E+00	8.18E-01	7.67E-02	1.15E-02	1.22E-13	8.60E-02	1.17E+00	2.93E-01	1.14E+02	3.42E-02	0.00E+00
	f_{Worst}	1.49E+01	1.77E+01	1.94E+01	1.99E+01	8.81E-13	1.98E+01	1.67E+01	2.09E+01	4.59E+02	2.14E+01	8.88E-16
f_{12}	f_{Best}	9.75E+01	6.76E+02	2.05E+03	4.69E+03	1.11E-16	2.47E+03	1.47E+02	5.17E+03	4.15E+08	6.61E+03	0.00E+00
	\bar{f}	1.15E+02	1.13E+03	2.23E+03	4.93E+03	7.67E-04	2.69E+03	3.08E+02	5.97E+03	9.32E+08	6.94E+03	0.00E+00
	\bar{f}	1.20E+01	3.08E+02	6.65E+01	1.02E+02	4.20E-03	7.40E+01	1.13E+02	3.43E+02	2.22E+08	1.69E+02	0.00E+00
	f_{Worst}	1.43E+02	1.83E+03	2.34E+03	5.12E+03	2.30E-02	2.84E+03	5.62E+02	6.62E+03	1.30E+09	7.27E+03	0.00E+00

TABLE 5. (Continued.) Minimization results with $n = 200$. Benchmark functions (available on Appendix A).

f_{13}	f_{Best}	-6.38E+00	-1.05E+01	2.13E+01	5.02E+01	-2.00E+01	2.41E+01	-8.32E+00	5.58E+01	5.78E+08	1.08E+02	-2.00E+01
	\bar{f}	-4.58E+00	-5.74E+00	2.38E+01	5.54E+01	-2.00E+01	2.73E+01	-5.38E+00	6.36E+01	1.56E+09	1.18E+02	-2.00E+01
	σ_f	8.13E-01	3.00E+00	8.64E-01	1.98E+00	4.57E-15	1.11E+00	1.69E+00	4.19E+00	4.95E+08	3.83E+00	0.00E+00
	f_{Worst}	-3.38E+00	2.41E+00	2.56E+01	5.90E+01	-2.00E+01	2.92E+01	-2.77E-01	7.32E+01	2.47E+09	1.28E+02	-2.00E+01
f_{14}	f_{Best}	1.72E+01	2.72E+00	1.71E+03	6.97E+03	2.72E+00	2.30E+03	6.25E+02	2.72E+00	9.98E-01	5.15E+04	2.72E+00
	\bar{f}	2.31E+01	2.72E+00	2.32E+03	8.64E+03	2.72E+00	2.92E+03	1.47E+03	5.62E+02	1.26E+00	5.88E+04	2.72E+00
	σ_f	3.28E+00	2.04E-06	2.57E+02	8.17E+02	3.55E-06	2.66E+02	5.43E+02	8.89E+02	6.86E-01	3.20E+03	1.49E-10
	f_{Worst}	3.15E+01	2.72E+00	2.85E+03	9.87E+03	2.72E+00	3.32E+03	2.51E+03	3.03E+03	2.98E+00	6.45E+04	2.72E+00
f_{15}	f_{Best}	4.09E+00	7.95E-04	8.80E+01	2.07E+02	1.98E-27	1.11E+02	5.52E+00	2.25E+02	3.39E-04	3.34E+02	-1.01E+00
	\bar{f}	4.89E+00	9.59E-04	9.89E+01	2.18E+02	1.41E-26	1.19E+02	1.25E+01	2.59E+02	9.38E-04	3.48E+02	-3.34E-01
	σ_f	4.94E-01	5.38E-05	3.44E+00	5.59E+00	1.95E-26	3.85E+00	4.44E+00	1.89E+01	3.74E-04	9.71E+00	4.42E-01
	f_{Worst}	6.07E+00	1.06E-03	1.04E+02	2.28E+02	1.08E-25	1.27E+02	2.07E+01	3.01E+02	1.44E-03	3.68E+02	1.17E-103
f_{16}	f_{Best}	1.06E+03	6.12E-01	3.72E+04	1.50E+05	8.40E-07	9.16E+04	9.81E+03	2.16E-15	3.25E+03	5.04E+05	9.97E-128
	\bar{f}	1.37E+03	6.00E+01	5.63E+04	1.77E+05	1.02E-05	9.97E+04	1.74E+04	3.34E+04	1.09E+04	5.99E+05	5.00E-99
	σ_f	2.16E+02	1.27E+02	5.36E+03	1.27E+04	1.25E-05	4.64E+03	5.66E+03	3.01E+04	3.47E+03	5.63E+04	2.73E-98
	f_{Worst}	2.01E+03	6.26E+02	6.46E+04	2.09E+05	6.71E-05	1.09E+05	2.73E+04	8.78E+04	1.83E+04	8.12E+05	1.50E-97
f_{17}	f_{Best}	1.21E+02	5.10E+01	7.86E+02	1.91E+03	1.36E+01	8.50E+02	1.87E+02	1.95E+03	3.97E+08	2.93E+03	1.03E+00
	\bar{f}	1.61E+02	8.80E+01	8.83E+02	2.08E+03	1.45E+01	9.55E+02	2.56E+02	2.36E+03	9.42E+08	3.17E+03	1.17E+00
	σ_f	2.50E+01	2.22E+01	3.48E+01	7.35E+01	3.97E-01	4.05E+01	5.18E+01	2.04E+02	3.13E+08	1.06E+02	4.38E-02
	f_{Worst}	2.36E+02	1.45E+02	9.36E+02	2.22E+03	1.52E+01	1.03E+03	4.29E+02	2.75E+03	1.68E+09	3.34E+03	1.25E+00
f_{18}	f_{Best}	1.28E+07	1.04E+07	1.75E+09	1.04E+10	5.04E+06	3.56E+09	3.69E+07	1.17E+10	3.00E+00	1.76E+10	1.31E-09
	\bar{f}	1.60E+07	3.80E+08	2.37E+09	1.16E+10	6.57E+06	4.04E+09	1.09E+08	1.43E+10	3.00E+00	1.87E+10	6.39E-08
	σ_f	1.55E+06	2.41E+08	1.91E+08	5.60E+08	5.34E+05	1.99E+08	1.28E+08	1.42E+09	1.01E-05	7.71E+08	1.08E-07
	f_{Worst}	1.85E+07	1.08E+09	2.61E+09	1.25E+10	7.43E+06	4.41E+09	4.94E+08	1.74E+10	3.00E+00	2.04E+10	5.60E-07
f_{19}	f_{Best}	1.00E+10	6.55E+08	1.66E+30	6.84E+56	5.96E-14	3.39E+05	2.55E+05	5.56E+05	2.49E+03	6.87E+12	4.79E-73
	\bar{f}	1.00E+10	7.11E+53	2.88E+48	1.42E+64	9.50E-14	1.58E+15	3.56E+05	6.71E+05	2.16E+04	3.96E+13	1.41E-52
	σ_f	0.00E+00	3.90E+54	1.07E+49	5.85E+64	3.25E-14	5.38E+15	3.96E+04	4.65E+04	1.07E+04	2.16E+13	6.73E-52
	f_{Worst}	1.00E+10	2.13E+55	5.42E+49	3.04E+65	1.86E-13	2.47E+16	4.31E+05	7.73E+05	5.93E+04	1.18E+13	3.67E-51
f_{20}	f_{Best}	2.09E+03	2.69E+03	7.60E+03	1.58E+04	1.99E+02	9.22E+03	1.94E+03	1.63E+04	9.99E+02	2.31E+04	1.99E+02
	\bar{f}	2.28E+03	4.68E+03	8.18E+03	1.65E+04	1.99E+02	9.74E+03	2.59E+03	1.89E+04	2.14E+03	2.45E+04	1.99E+02
	σ_f	7.60E+01	9.02E+02	2.88E+02	3.41E+02	6.99E-04	1.67E+02	3.19E+02	1.21E+03	6.64E+02	9.44E+02	1.72E-04
	f_{Worst}	2.42E+03	6.79E+03	8.78E+03	1.72E+04	1.99E+02	1.00E+04	3.32E+03	2.15E+04	3.37E+03	2.74E+04	1.99E+02
f_{21}	f_{Best}	5.41E+05	8.61E+07	2.06E+09	1.02E+10	2.19E+02	3.52E+09	3.01E+07	1.17E+10	-8.11E+00	1.71E+10	2.19E+02
	\bar{f}	1.30E+06	4.46E+08	2.45E+09	1.14E+10	2.21E+02	4.05E+09	1.39E+08	1.49E+10	-3.32E+00	1.84E+10	2.19E+02
	σ_f	6.51E+05	3.19E+08	2.05E+08	5.73E+08	3.32E+00	1.95E+08	1.70E+08	1.44E+09	2.25E+00	5.71E+08	7.29E-07
	f_{Worst}	3.52E+06	1.41E+09	2.87E+09	1.26E+10	2.31E+02	4.41E+09	4.86E+08	1.79E+10	-4.97E-01	1.98E+10	2.19E+02
f_{22}	f_{Best}	1.00E+10	3.84E+03	5.65E+38	2.52E+54	1.99E+02	1.12E+04	1.04E+04	1.92E+04	-7.46E+00	1.86E+12	1.99E+02
	\bar{f}	1.00E+10	1.51E+37	9.68E+47	2.11E+66	1.99E+02	1.12E+14	1.29E+04	2.18E+04	-4.85E+00	2.47E+13	1.99E+02
	σ_f	0.00E+00	8.02E+37	2.22E+48	9.37E+66	2.11E-13	4.66E+14	1.29E+03	1.77E+03	1.60E+00	1.35E+13	0.00E+00
	f_{Worst}	1.00E+10	4.40E+38	9.02E+48	5.01E+67	1.99E+02	2.46E+15	1.67E+04	2.58E+04	-9.09E-01	7.41E+13	1.99E+02
f_{23}	f_{Best}	1.00E+10	5.06E+07	7.39E+41	1.26E+53	-2.47E+02	3.76E+09	3.02E+09	7.07E+09	-8.34E+00	6.90E+12	-5.82E+02
	\bar{f}	1.00E+10	5.71E+52	2.29E+48	3.79E+64	-2.14E+02	2.78E+15	4.67E+09	8.53E+09	-4.83E+00	4.76E+13	-5.82E+02
	σ_f	0.00E+00	3.13E+53	9.80E+48	1.35E+65	1.65E+01	1.49E+16	9.15E+08	9.02E+08	1.88E+00	2.60E+13	1.60E-06
	f_{Worst}	1.00E+10	1.71E+54	5.37E+49	7.00E+65	-1.82E+02	8.16E+16	6.61E+09	1.07E+10	-9.43E-01	1.43E+13	-5.82E+02

CNBC approach with other metaheuristic methods. The evaluation focuses on the fitness values achieved by the algorithms. The optimization process aims to minimize these

fitness values. To control the optimization process, a maximum number of function evaluations, denoted as af , has been set as a stop criterion. In this case, the value of af is

TABLE 6. Wilcoxon p-values obtained over 30-dimensional results.

CNBC vs	BA	CS	CSA	DE	GWO	HS	PSO	SA	SCA	SMS
f_1	3.005E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_2	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	4.311E-08▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_3	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_4	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_5	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	2.975E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_6	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.000E+00▶	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.000E+00▶	1.212E-12▲
f_7	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_8	2.114E-11▲	2.114E-11▲	2.114E-11▲	2.113E-11▲	2.114E-11▲	2.114E-11▲	1.000E+00▶	2.114E-11▲	2.114E-11▲	2.114E-11▲
f_9	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.608E-01▶	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{10}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	2.113E-11▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{11}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	3.121E-13▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{12}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	2.158E-02▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{13}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.000E+00▶	1.212E-12▲	1.212E-12▲	1.212E-12▲	5.841E-09▲	9.330E-13▲
f_{14}	8.867E-12▲	8.867E-12▲	8.867E-12▲	8.867E-12▲	8.867E-12▲	8.867E-12▲	8.867E-12▲	1.526E-09▲	8.867E-12▲	8.867E-12▲
f_{15}	3.003E-11▲	3.010E-11▲	3.020E-11▲	3.020E-11▲	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	2.875E-11▲
f_{16}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{17}	3.020E-11▲	9.069E-03▲	3.020E-11▲	3.020E-11▲	2.921E-02▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	1.907E-01▶	3.020E-11▲
f_{18}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{19}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{20}	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲
f_{21}	2.398E-11▲	2.398E-11▲	2.398E-11▲	2.398E-11▲	2.936E-11▲	2.398E-11▲	2.398E-11▲	2.398E-11▲	2.398E-11▲	2.398E-11▲
f_{22}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.000E+00▶	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{23}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
▲	23	23	23	23	19	23	22	23	21	23
▼	0	0	0	0	0	0	0	0	0	0
▶	0	0	0	0	4	0	1	0	2	0

fixed at 5000. This ensures that the algorithms are given a limited number of function evaluations to find the optimal or near-optimal solutions. To account for the stochastic nature of the algorithms, each benchmark function is executed independently 30 times. By conducting multiple runs, the effects of randomness and variability in the algorithms’ performance can be better understood and statistically analyzed. For the experimental test, a population size of 50 individuals has been set for each metaheuristic approach. The population size determines the number of candidate solutions that are generated and evaluated in each iteration of the algorithm. Furthermore, the evaluation is conducted on three different dimensional search spaces: 30, 100, and 200 dimensions. This allows for the assessment of the scalability of the algorithms, examining their performance across different levels of problem complexity. Table 2 summarizes the parameter values used for each metaheuristic algorithm in the comparative test.

It should be noted that these parameter values correspond to the settings recommended by the original authors of each approach. The authors have selected these parameter values to obtain the best possible performance from their respective algorithms. By utilizing this consistent experimental setup with predefined parameter values, population size, and dimensional search spaces, a fair and controlled compari-

son of the algorithms can be conducted. This facilitates a meaningful assessment of their performance, highlighting the strengths and weaknesses of each approach.

B. PERFORMANCE COMPARISON

The performance results of each method are presented in Tables 3-5. These tables contain numerical indexes that are used to evaluate the performance of the algorithms. The indexes used are the Best Fitness Value, the Average Fitness, the Standard Deviation and the Worst Value. The Best Fitness Value (f_{Best}) represents the lowest fitness value achieved by each method. It indicates the quality of the solutions obtained, with lower values indicating better performance. The Average Fitness (\bar{f}) indicates the average fitness value across all runs of each method. It provides an overall measure of the algorithm’s performance and stability. The Standard Deviation (σ_f) quantifies the variability or dispersion of the fitness values obtained by each method. A lower standard deviation indicates more consistent performance. The Worst Value (f_{Worst}) represents the highest fitness value achieved by each method. It indicates the poorest quality solutions obtained, with higher values indicating poorer performance. The best performance entries in the tables are highlighted in boldface, making it easier to identify the methods that achieved the

TABLE 7. Wilcoxon p-values obtained over 100-dimensional results.

CNBC vs	BA	CS	CSA	DE	GWO	HS	PSO	SA	SCA	SMS
f_1	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_2	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_3	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_4	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	1.212E-12▲
f_5	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_6	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	4.806E-09▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_7	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_8	1.617E-11▲	1.618E-11▲	1.618E-11▲	1.618E-11▲	1.618E-11▲	1.618E-11▲	1.000E+00☐	1.618E-11▲	1.618E-11▲	1.618E-11▲
f_9	1.720E-12▲	1.720E-12▲	1.720E-12▲	1.720E-12▲	1.720E-12▲	1.720E-12▲	1.617E-12▲	1.720E-12▲	1.720E-12▲	1.713E-12▲
f_{10}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.046E-07▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{11}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	8.604E-13▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{12}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	8.152E-02☐	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{13}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.000E+00☐	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.076E-12▲
f_{14}	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	9.989E-11▲	1.015E-11▲	1.015E-11▲
f_{15}	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.001E-11▲
f_{16}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{17}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{18}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{19}	3.020E-11▲	1.212E-12▲	3.018E-11▲	3.020E-11▲	3.016E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.018E-11▲
f_{20}	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	1.000E+00☐	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲
f_{21}	2.506E-11▲	2.506E-11▲	2.506E-11▲	2.506E-11▲	2.505E-11▲	2.506E-11▲	2.506E-11▲	2.506E-11▲	2.506E-11▲	2.506E-11▲
f_{22}	1.212E-12▲	1.685E-14▲	1.212E-12▲	1.212E-12▲	1.000E+00☐	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{23}	3.005E-11▲	1.204E-12▲	3.005E-11▲	3.005E-11▲	3.005E-11▲	3.005E-11▲	3.005E-11▲	3.005E-11▲	3.005E-11▲	3.005E-11▲
▲	23	23	23	23	19	23	22	23	23	23
▼	0	0	0	0	0	0	0	0	0	0
☐	0	0	0	0	4	0	1	0	0	0

lowest fitness values. The tables present the minimization results obtained for three different dimensional search spaces: 30, 100, and 200 dimensions. Each table provides a comprehensive overview of the performance of the algorithms across these dimensions, allowing for a comparison of their effectiveness in solving optimization problems of varying complexity.

The CNBC approach presented in Table 3 shows better performance than its competitors for most of the benchmark functions ($f_1 - f_5, f_7, f_{10}, f_{11}, f_{15} - f_{19}$, and f_{23}). These results indicate that CNBC can find optimal solutions even in the presence of multiple optima. It also shows that the proposed approach produces similar results to GWO in functions $f_6, f_9, f_{12}, f_{13}, f_{20}, f_{21}$, and f_{22} . In addition, CNBC obtains the same fitness value as PSO and SA for functions f_8 and f_6 . In addition, SCA and CNBC share similar results for f_6, f_{13}, f_{20} , and f_{22} . Based on these numerical results, it can be seen that the proposed method shares the same results with GWO and SCA in the unimodal function f_6 , hybrid functions f_{20} and f_{22} , and multimodal function f_{13} . This condition can be produced because both methods consider the influence of several solutions in guiding the search process. In the proposed method, the trajectories produced by Bezier curves are influenced by a set of particles that represent the shortest path to the best solution. In GWO, the positions of the wolves are updated

based on the position of several kinds of wolves, which represent different types of solutions. The similarities in some results between the proposed method and SCA can also be explained by their coding mechanisms. The proposed method represents candidate solutions using a graph, where nodes represent the solutions and edges capture the differences in fitness values between solutions. Similarly, SCA utilizes a vector representation known as sine-cosine vectors to encode each solution. Then, there are five other functions ($f_8, f_9, f_{12}, f_{14}, f_{21}$) in which only one method (GWO, PSO, or SA) yields similar results to CNBC. Of these three algorithms, GWO appears most frequently. This reinforces the claim that GWO has a similar performance (owing to its search mechanism) for some of the benchmark functions, as in the proposed method. However, in the remaining 14 functions ($f_1 - f_5, f_7, f_{10}, f_{11}, f_{15} - f_{19}$, and f_{23}), the CNBC approach outperforms all other algorithms. This demonstrates the effectiveness of the proposed method in terms of accuracy and stability.

To evaluate the scalability of the proposed Complex Network-based method (CNBC), additional tests were conducted in higher-dimensional search spaces. Specifically, tests were performed in 100 and 200 dimensions. Table 4 presents the results obtained for the 100-dimensional test. The table clearly demonstrates that the CNBC approach outperforms its competitors on most of the benchmark functions.

TABLE 8. Wilcoxon p-values obtained over 200-dimensional results.

CNBC vs	BA	CS	CSA	DE	GWO	HS	PSO	SA	SCA	SMS
f_1	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	5.573E-10▲	3.020E-11▲	3.020E-11▲	3.016E-11▲
f_2	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.016E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_3	3.018E-11▲	3.018E-11▲	3.018E-11▲	3.018E-11▲	3.018E-11▲	3.018E-11▲	3.018E-11▲	3.018E-11▲	3.018E-11▲	3.018E-11▲
f_4	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	2.982E-11▲	3.020E-11▲	1.212E-12▲
f_5	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_6	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.142E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_7	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_8	1.613E-11▲	1.618E-11▲	1.618E-11▲	1.618E-11▲	1.616E-11▲	1.618E-11▲	1.000E+00☐	1.618E-11▲	1.618E-11▲	1.618E-11▲
f_9	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.204E-12▲
f_{10}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.057E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{11}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.208E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{12}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	2.708E-14▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{13}	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.000E+00☐	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.135E-12▲
f_{14}	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	1.015E-11▲	2.884E-04▲	6.082E-07▲	1.015E-11▲
f_{15}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.012E-11▲	3.020E-11▲	3.020E-11▲	2.986E-11▲
f_{16}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{17}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{18}	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.988E-04▲	3.020E-11▲	3.020E-11▲	3.020E-11▲
f_{19}	3.020E-11▲	1.212E-12▲	3.020E-11▲	3.018E-11▲	3.020E-11▲	3.020E-11▲	3.020E-11▲	3.003E-11▲	3.020E-11▲	3.020E-11▲
f_{20}	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲	2.262E-11▲
f_{21}	1.748E-11▲	1.748E-11▲	1.748E-11▲	1.748E-11▲	1.748E-11▲	1.748E-11▲	2.529E-02▲	1.748E-11▲	1.748E-11▲	1.748E-11▲
f_{22}	1.212E-12▲	1.685E-14▲	1.212E-12▲	1.212E-12▲	1.000E+00☐	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲	1.212E-12▲
f_{23}	2.982E-11▲	1.192E-12▲	2.982E-11▲	2.982E-11▲	2.982E-11▲	2.982E-11▲	2.982E-11▲	2.982E-11▲	2.982E-11▲	2.982E-11▲
▲	23	23	23	23	21	23	22	23	23	23
▼	0	0	0	0	0	0	0	0	0	0
☐	0	0	0	0	2	0	1	0	0	0

TABLE 9. Results obtained over 300-dimensional results.

Function		DSS	CNBC
Sphere	\bar{f}	0.00E+00	9.73E-84
	σ_f	0.00E+00	3.92E-65
Step	\bar{f}	4.56E-02	0.00E+00
	σ_f	1.14E-02	0.00E+00
Ackley	\bar{f}	8.88E-16	8.88E-16
	σ_f	0.00E+00	0.00E+00
Generalized Penalized	\bar{f}	6.67E-03	6.37E-06
	σ_f	2.01E-03	7.40E-06
Griewank	\bar{f}	0.00E+00	0.00E+00
	σ_f	0.00E+00	0.00E+00

TABLE 10. Results obtained over 3000-dimensional results.

Function		DSS	CNBC
Sphere	\bar{f}	0.00E+00	2.17E-70
	σ_f	0.00E+00	1.05E-69
Step	\bar{f}	5.74E-02	0.00E+00
	σ_f	1.52E-02	0.00E+00
Ackley	\bar{f}	8.88E-16	8.88E-16
	σ_f	0.00E+00	0.00E+00
Generalized Penalized	\bar{f}	1.10E-01	5.80E-05
	σ_f	1.20E-02	8.17E-05
Griewank	\bar{f}	0.00E+00	0.00E+00
	σ_f	0.00E+00	0.00E+00

Unlike the other algorithms, there are no instances in this test where two or more algorithms exhibit better performance than CNBC. This highlights the scalability of the proposed method, as its effectiveness is maintained even as the dimensionality increases. The results indicate that CNBC is capable of handling high-dimensional problems and offers superior performance compared to its counterparts. Furthermore, the GWO generates the same fitness value as the CNBC for $f_6, f_{10}, f_{12}, f_{13}, f_{20},$ and f_{22} . When analyzing the statistical values, it can be seen that CNBC is more robust than its principal

competitor (GWO) because CNBC’s mean values for $f_6, f_{10}, f_{12},$ and f_{20} are equal to the best fitness obtained, and their standard deviation values are smaller than those obtained by GWO. In addition, PSO exhibits similar performance to that of CNBC for functions $f_8,$ and SA has the same fitness value as CNBC in function f_{14} . However, the proposed approach is more consistent with SA because CNBC’s standard deviation of CNBC is smaller than that of SA.

It should be mentioned that the results obtained by CNBC for the 30 and 100-dimension tests are interesting because

they are very similar even when the dimensionality increases. This demonstrates the scalability of the proposed method. In addition, to include higher dimensionality in the optimization procedure and validate its effectiveness, the performance results based on a 200-dimensional search space are presented in Table 5.

This experiment illustrates the scalability and robustness of mixing Bezier curves and complex networks to obtain optimal solutions, even in high-dimensional search spaces. For most benchmark functions ($f_1 - f_7$, $f_9 - f_{12}$, $f_{15} - f_{19}$, and f_{23}), the proposed CNBC approach outperforms the other metaheuristic methodologies considered in the numerical experiment presented in Table 5. The performance of GWO is similar to that of CNBC in f_{13} , f_{14} , f_{20} , f_{21} , and f_{22} . As can be observed, since the 30-dimensional experiment (Table 3), GWO presents a performance similar to that of CNBC for some of the benchmark functions. However, the number of functions in which this occurs is reduced when the search space dimensions are increased. This result supports the scalability of the proposed approach. In addition, even when the GWO shows the same fitness and mean value, the CNBC has a smaller standard deviation, ensuring the method's effectiveness. In addition, CS and SA generate the same fitness value as CNBC in function f_{14} ; however, as mentioned previously, the standard deviation of CNBC is smaller. Finally, the PSO and proposed methods showed similar performances in f_8 .

Consequently, it is evident that the proposed approach generates exceptional performance in higher-dimensional search spaces. The Bezier curves used for generating feasible motion trajectories allow us to explore and explode the search space satisfactorily, particularly in higher-dimensional spaces. The numerical results obtained in higher-dimensional search spaces indicate that the representation of fitness relationships based on complex networks in the optimization process produces a higher level of scalability because it outperforms the rest of the tested methods for most benchmark functions.

The Wilcoxon signed-rank test [62] is commonly used to evaluate metaheuristic algorithms to assess their statistical significance and compare their performance against each other or against a baseline method. Metaheuristic algorithms are stochastic optimization methods that rely on randomness to explore and search for optimal solutions in complex problem spaces. Due to their stochastic nature, the performance of these algorithms can vary across different runs or datasets. The Wilcoxon test helps determine whether the observed differences in performance between two algorithms are statistically significant or simply due to chance. The Wilcoxon test is a non-parametric test that does not make any assumptions about the underlying distribution of the data. It compares the ranks of paired observations, typically the performance results obtained by two algorithms on the same set of benchmark functions. By comparing the ranks, the Wilcoxon test provides a p -value that indicates the likelihood of observing the observed differences in performance by chance. If the p -value is below a predetermined significance level (e.g.,

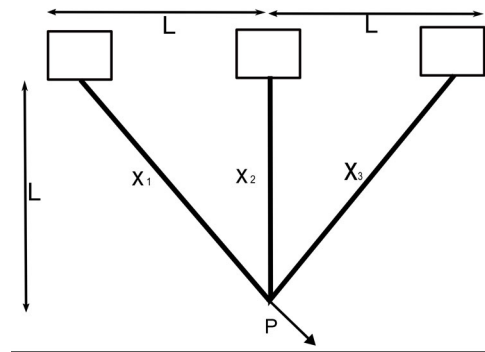


FIGURE 6. Graphical description of the three-bar truss design problem.

0.05), it suggests that the observed differences are statistically significant, indicating that one algorithm performs significantly better or worse than the other.

The p -values obtained from the Wilcoxon test are listed in Table 6. These statistical results were based on the performance results presented in Table 3 (where $n = 30$). On the other hand, Table 7 shows the p -values obtained by the rank-sum test considering the results of Table 4 (where $n = 100$). Finally, Table 8 presents the p -values obtained by the Wilcoxon test. The performance results for this case are listed in Table 5 (where $n = 200$). These tables present a pairwise comparison between CNBC and the rest of the tested algorithms to validate the experiments implemented statistically. Tables 6-8 employ the symbols \blacktriangle , \blacktriangledown , and \blacktriangleright . When CNBC achieves remarkably better results than a given adversary algorithm, the symbol \blacktriangle is used. Then, the symbol \blacktriangledown is utilized when CNBC generates a worse result than its opponents. Finally, the symbol \blacktriangleright is employed in the case when the Wilcoxon test cannot discriminate between the numerical results. This symbology can help to visualize the results clearly.

Based on the p -values in Table 6, it can be suggested that the Wilcoxon test cannot identify a significant difference between CNBC and GWO for functions f_6 , f_9 , f_{13} , and f_{22} . In addition, it can be confirmed that the proposed method produces a result similar to that of the SCA for functions f_6 , and f_{17} . Additionally, under the pairwise comparison of CNBC and PSO, it can be deduced that both algorithms show similar performance for the function f_8 . These results corroborate those presented in Table 3. In this analysis, the Wilcoxon test based on Table 3 demonstrated the effectiveness of the method in terms of accuracy and sturdiness. There are only seven comparative cases in which the rank sum is unable to distinguish between the numerical results. In addition, it is noticed that the three algorithms (GWO, PSO, and SCA) generate similar results to CNBC in six functions over twenty-three benchmark functions.

According to Table 7, (where the 100-dimensional results are exposed), the CNBC approach produces similar results to GWO for functions f_{12} , f_{13} , f_{20} , and f_{22} . Additionally, in function f_8 , the p -values obtained in the comparison between

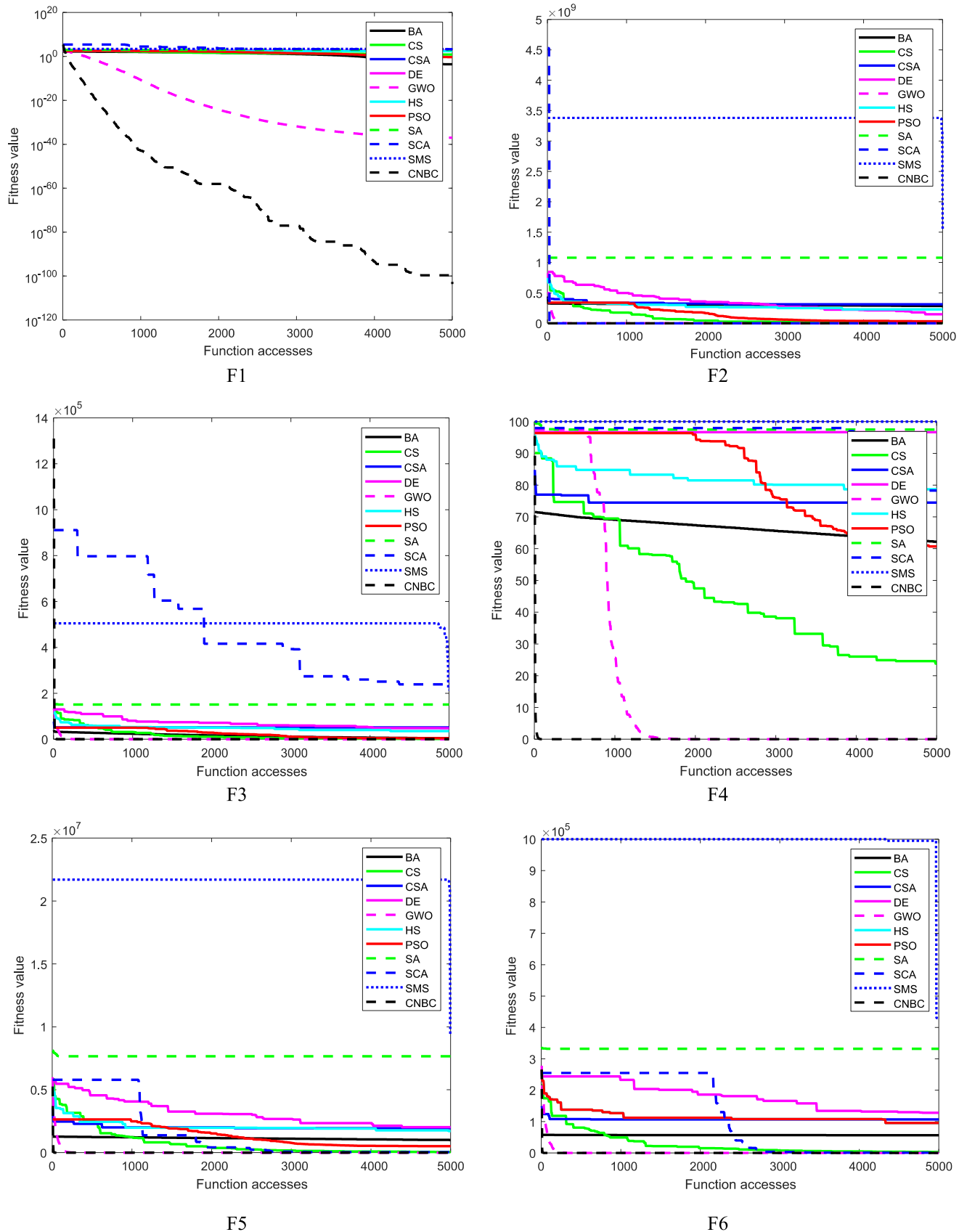


FIGURE 7. Convergence graphs from Table 4.

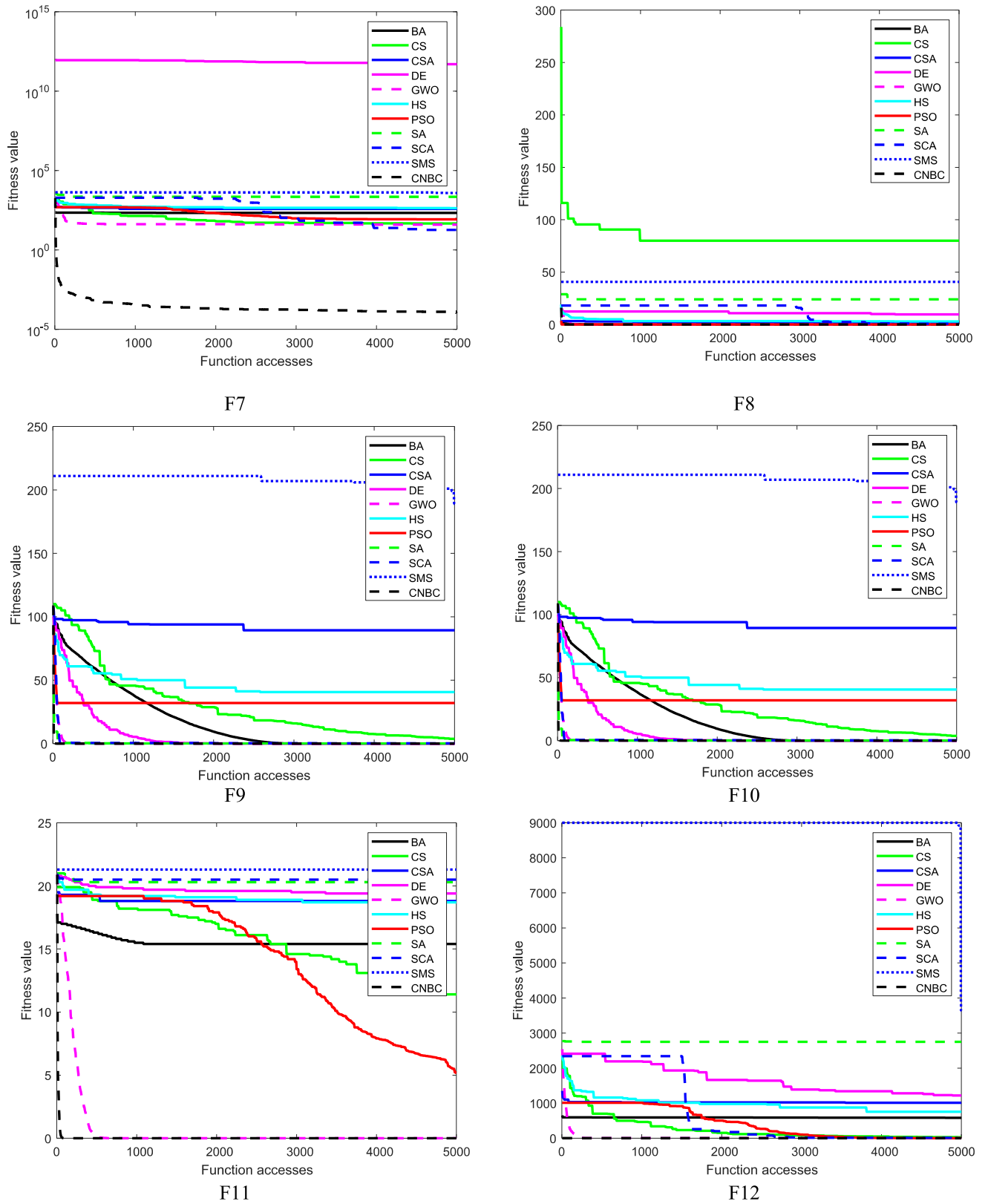


FIGURE 7. (Continued.) Convergence graphs from Table 4.

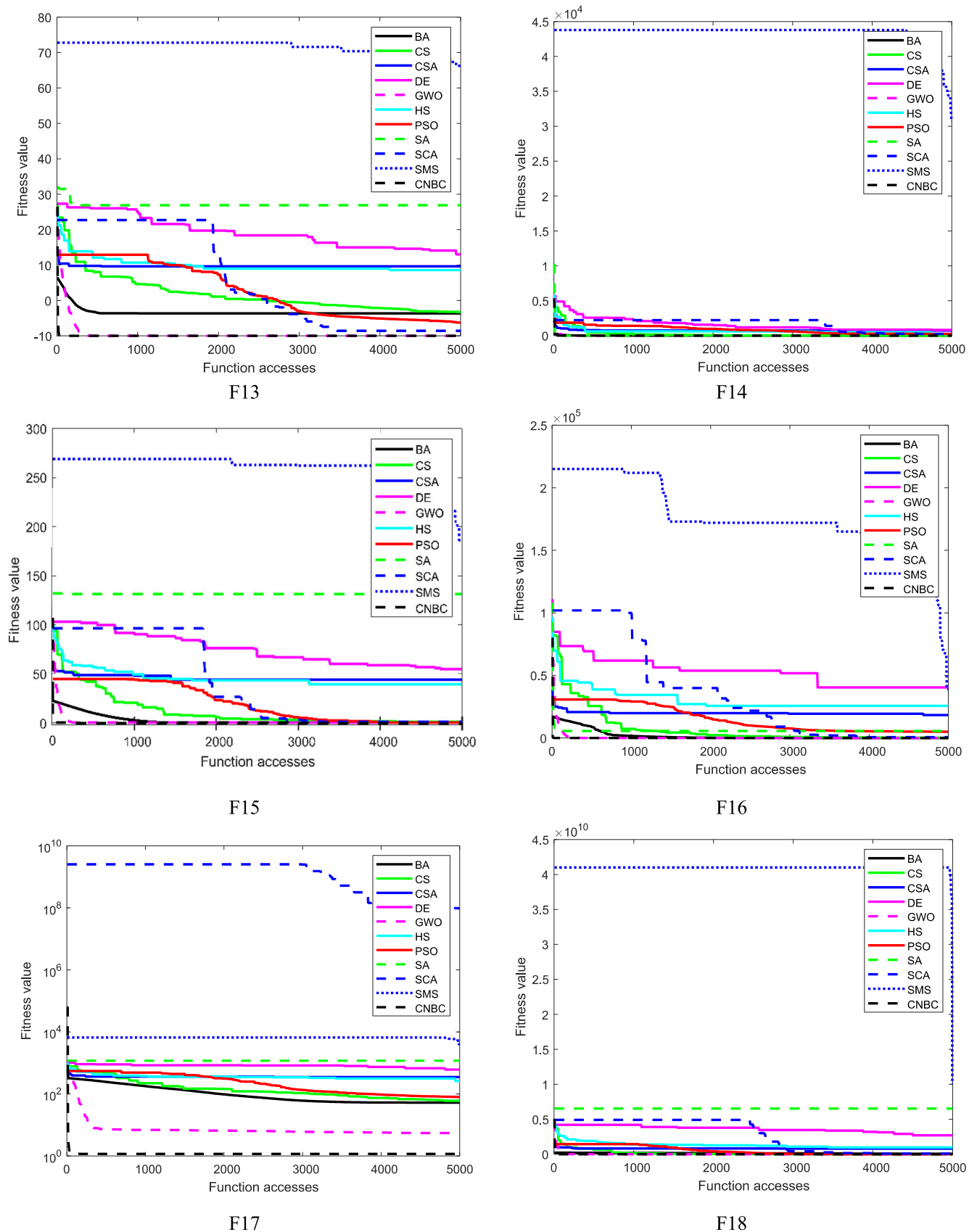


FIGURE 7. (Continued.) Convergence graphs from Table 4.

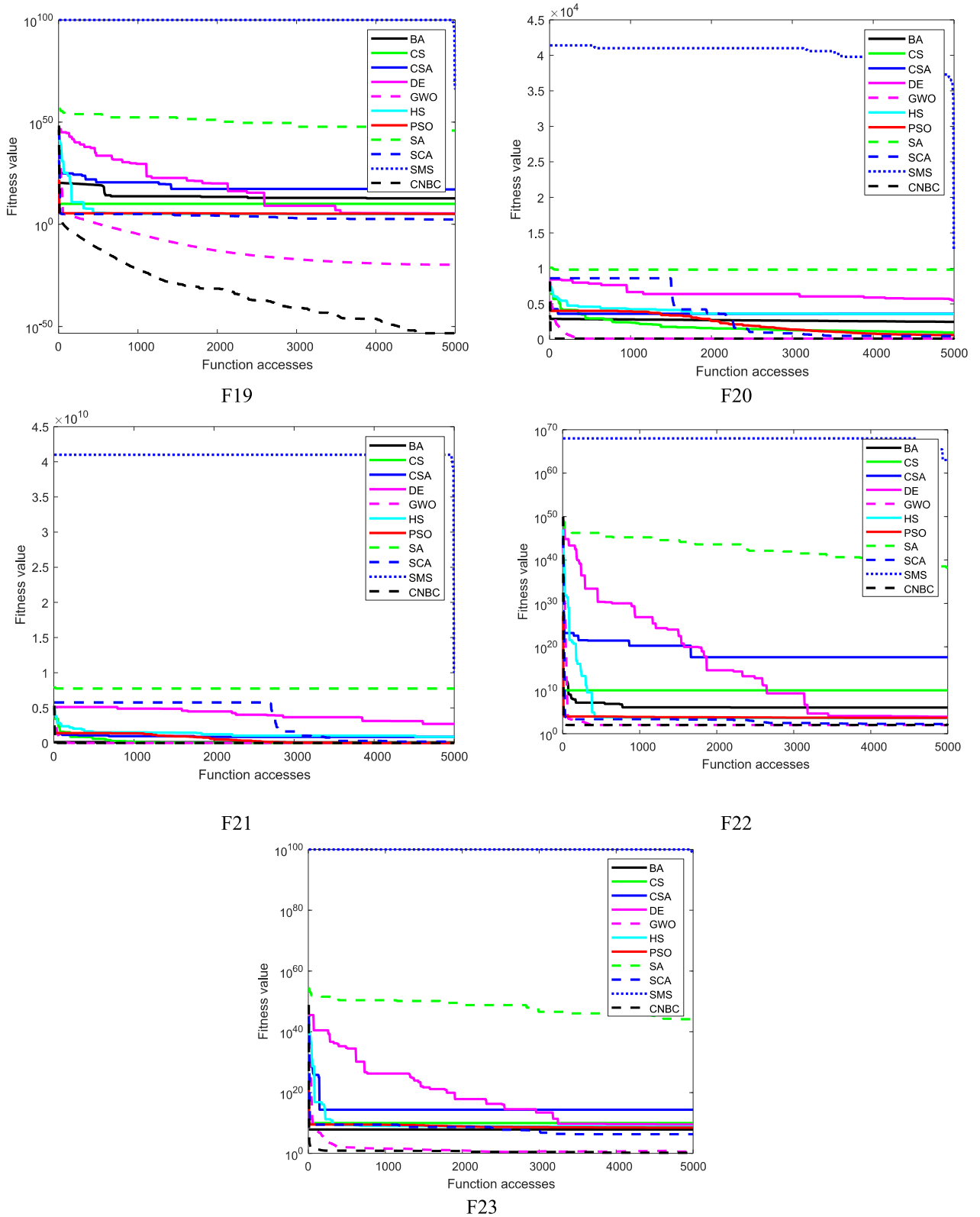


FIGURE 7. (Continued.) Convergence graphs from Table 4.

CNBC and PSO suggest that the rank sum cannot establish a significant difference between them. In addition, the

Wilcoxon test conducted for the 100-dimensional results supports the scalability characteristic of the proposed approach

TABLE 11. Numerical results of Three-Bar Truss Design Problem.

	x_1	x_2	$g_1(x)$	$g_2(x)$	$g_3(x)$	$f(x)$
BA	7.89E-01	4.07E-01	-8.31E-07	-1.47E+00	-5.35E-01	2.64E+02
CS	7.89E-01	4.08E-01	-1.12E-09	-1.46E+00	-5.36E-01	2.64E+02
CSA	8.37E-01	4.20E-01	-1.06E-01	-1.50E+00	-6.02E-01	2.64E+02
DE	7.89E-01	4.08E-01	8.06E-10	-1.46E+00	-5.36E-01	2.64E+02
GWO	7.90E-01	4.04E-01	-4.58E-06	-1.47E+00	-5.31E-01	2.64E+02
HS	7.92E-01	3.98E-01	-6.87E-06	-1.48E+00	-5.24E-01	2.64E+02
PSO	7.89E-01	4.08E-01	-2.79E-10	-1.46E+00	-5.35E-01	2.64E+02
SA	7.07E-01	7.07E-01	5.28E-10	-1.17E+00	-8.28E-01	2.64E+02
SCA	8.73E-01	2.10E-01	-5.74E-05	-1.71E+00	-2.91E-01	2.80E+02
SMS	7.56E-01	5.03E-01	4.15E-03	-1.36E+00	-6.37E-01	1.00E+300
CNBC	7.91E-01	4.02E-01	-2.20E-04	-1.47E+00	-5.29E-01	2.64E+02

TABLE 12. Statistical results of three-bar truss design problem.

Algorithm	Worst	Mean	Std	Best
BA	2.64E+02	2.64E+02	1.09E-04	2.64E+02
CS	2.64E+02	2.64E+02	9.38E-09	2.64E+02
CSA	2.64E+02	2.64E+02	3.01E-02	2.64E+02
DE	2.63E+02	2.64E+02	3.58E-13	2.64E+02
GWO	2.63E+02	2.64E+02	1.70E-03	2.64E+02
HS	2.64E+02	2.64E+02	2.42E-02	2.64E+02
PSO	2.64E+02	2.64E+02	1.14E-04	2.64E+02
SA	2.75E+02	2.70E+02	2.26E+00	2.64E+02
SCA	2.82E+02	2.81E+02	1.56E+00	2.80E+02
SMS	1.00E+300	1.00E+300	6.55E+04	1.00E+300
CNBC	2.64E+02	2.64E+02	1.08E-01	2.64E+02

because it is observed that only two algorithms (GWO and PSO) generate similar results to CNCB in five functions over twenty-three benchmark functions. It is evident that CNCB overcomes the other techniques that have been compared, especially in higher-dimensional search spaces.

Finally, based on Table 8, it can be pointed out that the Wilcoxon test cannot distinguish the results generated in f_{13} and f_{22} between the pairwise comparisons between CNBC and GWO. In addition, the p-values obtained by the rank-sum method indicate that the proposed method and PSO generate similar results in function f_8 . Results of Table 8 confirm the scalability property of CNBC because it should be mentioned that only two algorithms (GWO and PSO) generate similar results to CNCB in three functions over twenty-three benchmark functions. In addition, it verifies CNBC's effectiveness, robustness, and scalability because even when the dimensions increase, the proposed method produces feasible solutions, whereas other algorithms decrease their effectiveness.

To make a deeper analysis of the capabilities of the proposed method for dealing with high-dimensionals, a comparison is made against a metaheuristic algorithm for high dimensional optimization problems called Dynamic Stochastic

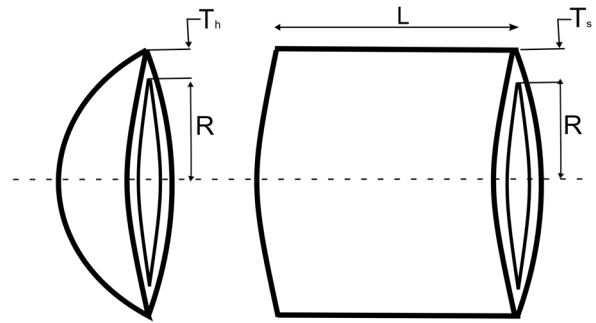


FIGURE 8. Graphical description of the Pressure Vessel Design Problem.

TABLE 13. Statistical results of pressure vessel design problem.

Algorithm	Worst	Mean	Std	Best
BA	8.60E+05	2.03E+05	2.33E+05	2.75E+04
CS	8.81E+03	8.81E+03	3.70E-12	8.81E+03
CSA	2.45E+05	1.00E+05	5.40E+04	2.55E+04
DE	8.98E+03	8.83E+03	3.27E+01	8.80E+03
GWO	1.37E+04	1.03E+04	2.28E+03	7.81E+03
HS	1.37E+04	9.21E+03	1.23E+03	8.81E+03
PSO	1.39E+04	1.09E+04	2.40E+03	8.81E+03
SA	1.00E+300	3.33E+299	6.55E+04	1.18E+05
SCA	8.83E+03	8.83E+03	1.23E-12	8.83E+03
SMS	1.00E+300	1.00E+300	6.55E+04	1.00E+300
CNBC	1.22E+04	1.39E+03	9.46E+03	7.24E+03

Search (DSS) [62]. This algorithm has presented its effectiveness in developing experiments over 300 and 3000 dimensions over unimodal and multimodal benchmark functions. In our comparison, it is considering a representative sample of these functions. The set of functions used for our comparison is constituted by unimodal and multimodal functions: Sphere, Step, Ackley, Generalized Penalized, and Griewank. These functions are described in Appendix A Table 17 as f_1, f_6, f_{11}, f_{12} , and f_{17} . Also, the parameters used are equal to the original article: the population size has been set in 50, the maximum number of iterations is set to 500, and both algorithms are performed for 30 experiments. The best results are highlighted in bold to emphasize the better performances.

Table 9 presents the comparative results between the Dynamic Stochastic Search and the proposed method on 300-dimensions. The proposed method shows its effectiveness over most of the analyzed functions. In Table 10, the comparative analysis is made over 3000-dimensions. Examining both tables, it can be observed that even when the dimensionality had increased ten times (from 300 to 3000 iterations), the method's effectiveness was maintained.

Based on this analysis and the statistical analysis presented in Tables 6–8, it can be corroborated that the metaheuristic approach based on complex networks and Bezier curves (CNBC) reduces the complexity of high-dimensional

TABLE 14. Numerical results of pressure vessel design problem.

	x_1	x_2	x_3	x_4	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$	$f(x)$
BA	3.35E+00	4.64E+01	5.71E+01	1.97E+02	-2.25E+00	-4.58E+01	-1.50E+06	-4.33E+01	2.75E+04
CS	9.56E-01	6.49E-01	5.08E+01	9.14E+01	2.43E-02	-1.64E-01	7.89E+03	-1.49E+02	8.81E+03
CSA	5.28E+01	9.81E+01	1.36E+02	5.90E+01	-5.02E+01	-9.68E+01	-1.27E+07	-1.81E+02	2.55E+04
DE	1.13E+00	1.29E+00	5.07E+01	9.08E+01	-1.47E-01	-8.08E-01	1.48E+04	-1.49E+02	8.80E+03
GWO	6.58E-01	1.07E+00	5.10E+01	9.08E+01	3.26E-01	-5.87E-01	-2.10E+03	-1.49E+02	7.81E+03
HS	1.36E+00	1.23E+00	5.11E+01	9.15E+01	-3.73E-01	-7.46E-01	-1.53E+04	-1.49E+02	8.81E+03
PSO	2.30E+00	7.02E-01	6.55E+01	1.12E+01	-1.04E+00	-7.71E-02	-3.06E+04	-2.29E+02	8.81E+03
SA	6.90E+01	1.47E+01	1.42E+01	1.08E+02	-6.87E+01	-1.46E+01	1.22E+06	-1.32E+02	1.18E+05
SCA	6.28E-01	1.23E+00	4.11E+01	2.00E+02	1.65E-01	-8.40E-01	-5.57E+04	-4.00E+01	8.83E+03
SMS	4.35E+01	8.39E+01	4.44E+01	7.93E+01	-4.26E+01	-8.35E+01	4.37E+05	-1.61E+02	1.00E+300
CNBC	1.16E+00	6.03E-01	3.85E+01	4.48E+01	-4.19E-01	-2.36E-01	8.48E+05	-1.95E+02	7.24E+03

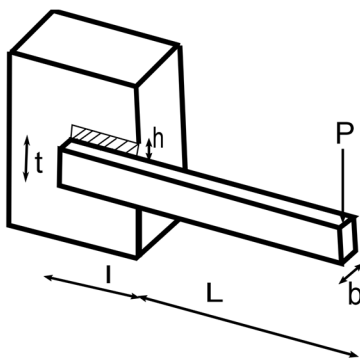


FIGURE 9. Graphical description of the Welded-Beam Design Problem.

functions by modeling the fitness relations between search agents using complex networks. In addition, the effectiveness of the method in terms of accuracy and scalability is validated.

The remarkable capabilities of the proposed method can be attributed to its unique mechanisms that are employed during the optimization process. Two fundamental mechanisms contribute to its effectiveness: the generation of agents used for generating possible trajectories and the selection of points p_1 and p_2 based on different cases depending on the algorithm's performance.

Firstly, the method leverages the use of agents to generate trajectories. These agents represent different candidate

TABLE 15. Statistical results of welded-beam design problem.

Algorithm	Worst	Mean	Std	Best
BA	1.00E+300	3.67E+299	6.55E+04	2.94E+00
CS	3.34E+00	2.60E+00	1.93E-01	2.45E+00
CSA	1.00E+300	7.33E+299	6.55E+04	8.74E+00
DE	2.71E+00	2.55E+00	7.38E-02	2.45E+00
GWO	1.00E+300	3.33E+299	6.55E+04	2.44E+00
HS	3.33E+00	2.85E+00	2.49E-01	2.46E+00
PSO	2.47E+00	2.45E+00	1.90E-02	2.44E+00
SA	1.00E+300	8.67E+299	6.55E+04	3.69E+00
SCA	1.00E+300	1.00E+300	6.55E+04	1.00E+300
SMS	1.00E+300	1.00E+300	6.55E+04	1.00E+300
CNBC	3.08E+00	3.29E+00	4.12E-01	2.53E+00

solutions within the search space. By considering the relationships and distances between these agents, the method constructs complex networks. It applies graph-based algorithms such as Dijkstra's algorithm to identify the shortest paths. This enables the generation of feasible trajectories that guide the exploration and exploitation of the search space. Secondly, the method utilizes different cases for selecting the points p_1 and p_2 . These cases are determined based on the algorithm's performance, which is reflected in a counter

TABLE 16. Statistical results of welded-beam design problem.

	x_1	x_2	x_3	x_4	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$	$g_5(x)$	$g_6(x)$	$g_7(x)$	$f(x)$
BA	5.28E-01	5.15E+00	4.35E+00	8.89E-01	-8.93E+00	-3.43E+00	-3.61E-01	-1.41E+00	-4.03E-01	-2.20E-01	-2.69E+05	2.94E+00
CS	2.13E-01	8.59E+00	8.76E+00	2.21E-01	-9.24E+01	-3.15E+02	-8.57E-03	-2.89E+00	-8.78E-02	-2.35E-01	-1.32E+03	2.45E+00
CSA	1.83E+00	2.95E-01	2.01E-01	1.82E+00	2.03E+05	6.82E+06	1.33E-02	-4.40E+00	-1.71E+00	1.48E+02	-1.17E+05	8.74E+00
DE	1.99E-01	9.12E+00	8.97E+00	2.13E-01	-3.46E+02	-5.76E+02	-1.39E-02	-2.87E+00	-7.41E-02	-2.36E-01	-6.26E+02	2.45E+00
GWO	2.04E-01	8.75E+00	9.04E+00	2.06E-01	-4.32E+00	-1.77E+00	-1.64E-03	-2.96E+00	-7.91E-02	-2.36E-01	-1.89E+00	2.44E+00
HS	2.37E-01	7.94E+00	8.43E+00	2.38E-01	-6.92E+00	-1.91E+02	-7.51E-04	-2.88E+00	-1.12E-01	-2.35E-01	-2.86E+03	2.46E+00
PSO	1.72E-01	1.00E+01	9.04E+00	2.06E-01	-7.28E-12	-5.09E-11	-3.37E-02	-2.85E+00	-4.70E-02	-2.36E-01	-4.55E-11	2.44E+00
SA	8.00E-01	6.17E+00	9.96E+00	1.70E+00	-8.74E+03	-2.70E+04	-9.05E-01	1.15E+01	-6.75E-01	-2.49E-01	-3.62E+06	3.69E+00
SCA	1.30E+00	8.40E+00	8.93E+00	1.94E+00	-1.26E+04	-2.67E+04	-6.35E-01	1.38E+01	-1.18E+00	-2.48E-01	-4.95E+06	1.00E+300
SMS	1.09E+00	8.04E+00	7.93E+00	9.35E-01	-1.07E+04	-2.14E+04	1.51E-01	2.98E+00	-9.61E-01	-2.45E-01	-5.08E+05	1.00E+300
CNBC	3.23E-01	2.39E+00	7.49E+00	3.83E-01	-6.82E+01	-6.56E+03	-6.02E-02	-2.73E+00	-1.98E-01	-2.36E-01	-2.80E+04	2.53E+00

variable. By considering the success or delay in finding better solutions, the method adapts its selection strategy. This flexibility allows for the incorporation of both random exploration and targeted exploitation, enabling a more robust and effective search process. By combining these mechanisms, the proposed method enhances its capacity to navigate the search space, balance exploration and exploitation, and identify promising trajectories to improve the optimization process. These distinctive features contribute to the remarkable performance and capabilities of the method in tackling complex optimization problems.

C. CONVERGENCE ANALYSIS

Convergence analysis in metaheuristic methods refers to the study of the algorithm’s behavior and its convergence towards an optimal or near-optimal solution over iterations. The main objective of convergence analysis is to assess the performance and effectiveness of a metaheuristic algorithm in terms of its ability to converge to a high-quality solution within a reasonable number of iterations. This subsection studied the converge analysis between the proposed method (CNBC) and ten well-known metaheuristic algorithms (BA, CS, CSA, DE, GWO, HS, PSO, SA, SCA, and SMS).

Convergence data from the 100-dimensional test (Table 4) were used to generate the convergence graphs. This information was selected because it can represent the general performance of the proposed method, as it is the middle experiment of the three conducted in Section (4.2) over 30-, 100-, and 200-dimensional spaces. This information

facilitates understanding the performance in both low and high dimensionality. Convergence graphs are presented in Figure 7. Based on its convergence, the CNBC approach has a faster convergence rate than its competitors. This suggests that complex networks and Bezier curve operators allow the manipulation of a large amount of information and represent it by simple connections between nodes, generating feasible agent trajectories to explore and exploit the search space. Some of the functions on which it is easier to observe CNBC profit are $f_1, f_7, f_{11}, f_{13}, f_{19}$, and f_{23} .

To produce the trajectories, the Dijkstra’s algorithm selects the path with the minimal cost in terms of the weights (fitness values) of the edges in a graph. Prioritizing small differences can accelerate the algorithm’s convergence towards a near-optimal solution. By placing more weight on subtle improvements, the algorithm can quickly converge towards regions with incremental gains in fitness.

D. ENGINEERING DESIGN PROBLEMS

The optimization method’s main purpose is to generate an optimal solution to an optimization problem. Many disciplines, such as medicine, engineering, economics, among others had defined many of their fundamental problems as optimization problems. This section exhibits the proposed method applied over several engineering design problems to evaluate the capabilities of CNBC in real-world applications.

The engineering design problems used in this section are the three-bar Truss design problem, pressure vessel design problem, and welded-beam design problem [63], [64]. They

TABLE 17. Mathematical description of the test functions.

Function	S	Dimension	Minimum
$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100,100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-100,100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j^2 \right)^2$	$[-100,100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100,100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_5(x) = \sum_{i=1}^n \sum_{j=1}^i x_j^2$	$[-65.536, 65.536]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	$[-100,100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}(0,1)$	$[-1.28, 1.28]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_8(x) = \sum_{i=1}^n x_i^6 (\sin(x_i) + 2)$	$[-1, 1]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_9(x) = \sum_{i=1}^n x_i^2 + \sum_{i=1}^n \frac{x_i^2}{(-x_i + \sum_{i=1}^n x_{i1})^4}$	$[10e^{-6}, 2]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = \left(\frac{1}{n-1}, \dots, \frac{1}{n-1} \right);$ $f(x^*) = \frac{2n}{(n-1)^2}$
$f_{10}(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_{11}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_{12}(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_{13}(x) = -0.1 \sum_{i=1}^n \cos(5\pi x_i) - \sum_{i=1}^n x_i^2$	$[-1, 1]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = -0.1 \times n$
$f_{14}(x) = \sum_{i=1}^n [x_i^2(2x_i^2 + x_{i+1} + 2) - x_i x_{i-1}(3x_i + 3x_{i-1} - x_{i+1})]$ $+ \frac{1}{n^2} \sum_{i=1}^n (x_i - 1)^2 + e; x_{n+1} = x_1, x_0 = x_n$	$[-1, 2]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (1, \dots, 1);$ $f(x^*) = 1$
$f_{15}(x) = -(n+1)e^{-10\sqrt{n}[\sum_{i=1}^n (x_i-1)^2]^{1/2}} + \sum_{i=1}^n x_i^2$	$[-2, 2]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (1, \dots, 1);$ $f(x^*) = -1$
	$[-4, 5]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$

TABLE 17. (Continued.) Mathematical description of the test functions.

$f_{16}(x) = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$			
$f_{17}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_i - 1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1}) + (y_n - 1)^2] \right\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4);$ $y_i = 1 + \frac{x_i + 1}{4};$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_{18}(x) = 0.1 \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_1 + 1)] + ((x_n - 1)^2 [1 + \sin^2(2\pi x_n)]) \right\} + u(x_i, 5, 100, 4)$	$[-50, 50]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (1, \dots, 1);$ $f(x^*) = 0$
$f_{19}(x) = f_1(x) + f_2(x) + f_{10}(x)$	$[-100, 100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = 0$
$f_{20}(x) = f_5(x) + f_{10}(x) + f_{12}(x)$	$[-100, 100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = n - 1$
$f_{21}(x) = f_3(x) + f_5(x) + f_{11}(x) + f_{18}(x)$	$[-100, 100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = (1.1 \times n) - 1$
$f_{22}(x) = f_2(x) + f_5(x) + f_{10}(x) + f_{11}(x) + f_{12}(x)$	$[-100, 100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (0, \dots, 0);$ $f(x^*) = n - 1$
$f_{23}(x) = f_1(x) + f_2(x) + f_{11}(x) + f_{16}(x) + f_{17}(x)$	$[-100, 100]^n$	$n = 30$ $n = 100$ $n = 200$	$x^* = (1, \dots, 1);$ $f(x^*) = -3 \times n$

are described in Appendix B. Below, the obtained results for each optimization problem are presented.

1) THREE-BAR TRUSS DESIGN PROBLEM

The Three-Bar Truss Design Problem is an optimization design problem that considering a 2-dimensional search space subject to three constraints (see Appendix B Table 18). This optimization problem focuses on reducing the volume of a loaded three-bar truss. Figure 6 illustrates the structure of this problem. Additionally, Table 11 shows the decision variables, the constraints, and the fitness value obtained for applying CNBC in the Three-Bar Truss design problem.

Table 12 reports the statistical results obtained for the Three-Bar Truss executed independently 30 times. For each experiment, the maximum number of generations is set to 1000 for each algorithm. In this table the worst, mean, standard deviation, and the best fitness values are presented. Based on both tables, it is exhibited that the proposed method can obtain competitive results like other well-known meta-

heuristic algorithms such as BA, CS, CSA, DE, GWO, HS, and PSO. Also, the obtained results validate the adaptability of the CNBC algorithm to function adequately for high and low-dimensional problems.

2) PRESSURE VESSEL DESIGN PROBLEM

The Pressure Vessel Design is one of the most common engineering problems used for validating optimization algorithms. It consists of finding the optimal design of a compressed air storage tank considering the thickness of shell ($T_s = x_1$), thickness of head ($T_h = x_2$), inner radius ($R = x_3$), and length of shell ($L = x_4$) such that the total cost of material, forming, and welding is minimized based on four constraints. The complete definition of this problem can be found in Appendix B Table 19. In addition, Figure 8 illustrates a graphic representation of the Pressure Vessel Design Problem.

In Table 13 the statistical results are presented, and Table 14 exposes the numerical results obtained for

TABLE 18. Definition of three-bar truss design problem.

Equation	$\min_{\vec{x}} f(\vec{x}) = (2\sqrt{2}x_1 + x_2) \times H$
Constraints	$g_1 = \frac{\sqrt{2x_1} + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$ $g_2 = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$ $g_3 = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0$
Parameters	$0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, H = 100cm,$ $P = 2KN/cm^2$, and $\sigma = 2KN/cm^2$

TABLE 19. Definition of pressure vessel design problem.

Equation	$\min_{\vec{x}} f(\vec{x}) = 0.6224x_1x_3x_4$ $+1.7781x_2x_3^2 + 3.1661x_1^2x_4$ $+19.84x_1^2x_3$
Constraints	$g_1 = x_1 + 0.0193x_3 \leq 0,$ $g_2 = -x_2 + 0.00954x_3 \leq 0,$ $g_3 = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,$ $g_4 = x_4 - 240 \leq 0,$
Parameters	$0.0625 \leq x_1 \leq 6.1875,$ $0.0625 \leq x_2 \leq 6.1875,$ $10 \leq x_3 \leq 200,$ $10 \leq x_4 \leq 200$

employing the CNBC algorithm to solve the Pressure Vessel Optimization Problem. It is notable that the flexibility of the proposed approach allows the incorporation of both random

TABLE 20. Definition of welded design problem.

Equation	$\min_{\vec{x}} f(\vec{x}) = 1.10471x_1^2x_2$ $+0.04811x_3x_4(14 + x_2)$
Constraints	$g_1 = \tau(\vec{x}) - \tau_{max} \leq 0,$ $g_2 = \sigma(\vec{x}) - \sigma_{max} \leq 0,$ $g_3 = x_1 - x_4 \leq 0,$ $g_4 = 0.10471x_1^2 + 0.04811x_3x_4$ $* (14 + x_2) - 5 \leq 0,$ $g_5 = 0.125 - x_1 \leq 0,$ $g_6 = \delta(\vec{x}) - \delta_{max} \leq 0,$ $g_7 = P - P_c(\vec{x}) \leq 0$
Parameters	$0.1 \leq x_1 \leq 2, 0.1 \leq x_2 \leq 10,$ $0.1 \leq x_3 \leq 10, 0.1 \leq x_4 \leq 2,$ $\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$ $\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right),$ $R = \sqrt{\frac{x_2^2}{2} + \left(\frac{x_1 + x_3}{2}\right)^2},$ $\delta(x) = \frac{4PL^3}{Ex_3^3x_4},$ $J = 2\left[\sqrt{2}x_1x_2\left\{\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right\}\right],$ $\sigma(x) = \frac{6PL}{x_4x_3^2},$ $P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$

exploration and targeted exploitation, enabling a more robust and effective search process.

3) WELDED-BEAM DESIGN PROBLEM

This design problem optimizes the fabrication cost of a welded beam based on the width $h = x_1$, length $l = x_2$, depth $t = x_3$ and thickness $b = x_4$. Also, this challenge considers seven constraints. The specifications of the Welded-Beam design problem are exposed in Appendix B Table 20. In addition, figure 9 illustrates the graphical representation of this optimization problem.

Table 15 displays the statistical results, and Table 16 shows the numerical results of the Welded-Beam optimization problem. However, the proposed method generates a less favorable fitness value than to CS, DE, GWO, HS, and PSO. It has the capability to produce better results than BA, CSA, SA, SCA, and SMS. Therefore, the proposed approach produces competitive results. Its performance is attributed to the higher-level representation applied using complex networks that capture the essential relationships and dependencies among the solutions, which are employed to create potential search trajectories using Bezier curves, producing feasible solutions that can correctly explore and exploit the search space.

V. CONCLUSION

Metaheuristic techniques are powerful optimization methods capable of handling complex and large-scale problems without relying on explicit mathematical models. However, due to the inherent complexity and diversity of optimization problems, not all metaheuristics can solve all problems competitively. Each problem requires specific strategies for optimal solutions, necessitating the introduction of new methods to enhance the capabilities of metaheuristics across various domains. Traditional metaheuristic algorithms often draw inspiration from natural systems, using metaphors and analogies. While this approach has its merits, it can limit innovation by encouraging the replication of existing algorithms with slight modifications. In contrast, metaphor-free metaheuristic algorithms focus on developing novel algorithmic mechanisms based on a combination of mathematical and computational principles. Considering these mechanisms, these methods can explore unconventional search operators and offer improved performance across diverse problem domains.

This paper introduces a novel metaphor-free metaheuristic algorithm that combines complex networks and Bezier curves. In this approach, solutions are represented as nodes in a graph, where the edges capture the differences in their objective function values. This graph-based representation enables a higher-level understanding of solution relationships and dependencies. The algorithm calculates the shortest path between each solution and the best solution, using the resulting nodes as control points in the Bezier equation to determine new agent positions. Throughout the optimization process, the graph is continuously updated based on the evaluation of new solutions and their objective function values, leading to trajectories that facilitate the exploration and exploitation of the search space.

A set of unimodal, multimodal, and hybrid benchmark functions was used to numerically and statistically compare the performance of the proposed approach with various state-of-the-art metaheuristic algorithms. The comparative results demonstrated the effectiveness of the method. In addition, the robustness and scalability of the proposed approach have been verified because even when the dimensionality increases, the proposed method produces feasible solutions, whereas other algorithms decrease their effectiveness.

APPENDIX A

See Table 17.

APPENDIX B

See Tables 18–20.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding this work.

REFERENCES

- [1] X.-S. Yang. (2011). *Metaheuristic Optimization—Scholarpedia*. Scholarpedia. Accessed: Jun. 16, 2022. [Online]. Available: http://var.scholarpedia.org/article/Metaheuristic_Optimization
- [2] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: A comprehensive survey," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2191–2233, Dec. 2019.
- [3] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic Applications in Structures and Infrastructures*. Boston, MA, USA: Newnes, 2013.
- [4] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [5] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, *Metaheuristic Algorithms: A Comprehensive Review*. Amsterdam, The Netherlands: Elsevier, 2018.
- [6] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021.
- [7] J. R. Sampson, "Adaptation in natural and artificial systems (John H. Holland)," *SIAM Rev.*, vol. 18, no. 3, pp. 529–530, Jul. 1976.
- [8] J. R. Koza, *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, vol. 34. Stanford, CA, USA: Stanford Univ., 1990.
- [9] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986.
- [10] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *Australas. Plant Pathol.*, vol. 38, no. 3, pp. 284–287, 1995.
- [11] D. B. Fogel, *Artificial Intelligence Through Simulated Evolution*. Hoboken, NJ, USA: Wiley, 1998.
- [12] I. Rechenberg, "Evolution strategy: Optimization of technical systems by means of biological evolution," *Fromman-Holzboog, Stuttgart*, vol. 104, pp. 15–16, Jan. 1973.
- [13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Perth, WA, Australia, 1995, pp. 1942–1948, doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [14] X.-S. S. Yang, "A new metaheuristic bat-inspired algorithm BT—Nature inspired cooperative strategies for optimization (NCSO 2010)," *Stud. Comput. Intell.*, vol. 284, pp. 65–74, Jan. 2010.
- [15] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.

- [16] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.
- [17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [18] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [19] Z. Woo Geem, J. Hoon Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001.
- [20] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
- [21] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, "An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation," *Appl. Intell.*, vol. 40, no. 2, pp. 256–272, Mar. 2014.
- [22] S. Kumar, P. Jangir, G. G. Tejani, M. Premkumar, and H. H. Alhelou, "MOPGO: A new physics-based multi-objective plasma generation optimizer for solving structural optimization problems," *IEEE Access*, vol. 9, pp. 84982–85016, 2021.
- [23] S. Kumar, G. G. Tejani, N. Pholdee, S. Bureerat, and P. Jangir, "Multi-objective teaching-learning-based optimization for structure optimization," *Smart Sci.*, vol. 10, no. 1, pp. 56–67, Jan. 2022.
- [24] P. Singh, R. Kottath, and G. G. Tejani, "Ameliorated follow the leader: Algorithm and application to truss design problem," *Structures*, vol. 42, pp. 181–204, Aug. 2022.
- [25] S. Kumar, P. Jangir, G. G. Tejani, and M. Premkumar, "A decomposition based multi-objective heat transfer search algorithm for structure optimization," *Knowl.-Based Syst.*, vol. 253, Oct. 2022, Art. no. 109591.
- [26] S. Kumar, G. G. Tejani, N. Pholdee, and S. Bureerat, "Performance enhancement of meta-heuristics through random mutation and simulated annealing-based selection for concurrent topology and sizing optimization of truss structures," *Soft Comput.*, vol. 26, no. 12, pp. 5661–5683, Jun. 2022.
- [27] N. Panagant, S. Kumar, G. G. Tejani, N. Pholdee, and S. Bureerat, "Many-objective meta-heuristic methods for solving constrained truss optimisation problems: A comparative analysis," *MethodsX*, vol. 10, 2023, Art. no. 102181.
- [28] A. Nonut, Y. Kanokmedhakul, S. Bureerat, S. Kumar, G. G. Tejani, P. Artrit, A. R. Yildiz, and N. Pholdee, "A small fixed-wing UAV system identification using metaheuristics," *Cogent Eng.*, vol. 9, no. 1, pp. 1–20, Dec. 2022. [Online]. Available: <http://www.editorialmanager.com/cogenteng>
- [29] C. Aranha, C. L. Camacho Villalón, F. Campelo, M. Dorigo, R. Ruiz, M. Sevaux, K. Sörensen, and T. Stützle, "Metaphor-based metaheuristics, a call for action: The elephant in the room," *Swarm Intell.*, vol. 16, no. 1, pp. 1–6, Nov. 2021.
- [30] K. Sörensen, "Metaheuristics—The metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, Jan. 2015.
- [31] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method," *Expert Syst. Appl.*, vol. 181, Nov. 2021, Art. no. 115079.
- [32] C. Blum and A. Roli, "Hybrid metaheuristics: An introduction," *Stud. Comput. Intell.*, vol. 114, pp. 1–30, Jan. 2008.
- [33] E. Estrada, *The Structure of Complex Networks*. New York, NY, USA: Oxford Univ. Press, 2012.
- [34] M. A. Niazi, "Introduction to the modeling and analysis of complex systems: A review," *Complex Adapt. Syst. Model.*, vol. 4, no. 1, p. 3 and 1–14, doi: 10.1186/s40294-016-0015-x.
- [35] P. Passos, K. Davids, D. Araújo, N. Paz, J. Minguéns, and J. Mendes, "Networks as a novel tool for studying team ball sports as complex social systems," *J. Sci. Med. Sport*, vol. 14, no. 2, pp. 170–176, Mar. 2011.
- [36] F. Vega-Redondo, *Complex Social Networks*, no. 44. Cambridge, U.K.: Cambridge Univ. Press, 2007.
- [37] N. N. Daud, S. H. A. Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, "Applications of link prediction in social networks: A review," *J. Netw. Comput. Appl.*, vol. 166, Sep. 2000, Art. no. 102716.
- [38] B. J. Kim, T. Jun, J.-Y. Kim, and M. Y. Choi, "Network marketing on a small-world network," *Phys. A, Stat. Mech. Appl.*, vol. 360, no. 2, pp. 493–504, Feb. 2006.
- [39] L. S. de Pontes and L. C. Rêgo, "Impact of macroeconomic variables on the topological structure of the Brazilian stock market: A complex network approach," *Phys. A, Stat. Mech. Appl.*, vol. 604, Oct. 2022, Art. no. 127660.
- [40] Z. Chen, J. Wu, Y. Xia, and X. Zhang, "Robustness of interdependent power grids and communication networks: A complex network perspective," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 1, pp. 115–119, Jan. 2018.
- [41] X. Li and S. Xu, "Multi-sensor complex network data fusion under the condition of uncertainty of coupling occurrence probability," *IEEE Sensors J.*, vol. 21, no. 22, pp. 24933–24940, Nov. 2021.
- [42] D. Ficzer, "Complex network theory to model 5G network slicing," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, Jun. 2022, pp. 1–4.
- [43] G. A. Pagani and M. Aiello, "The power grid as a complex network: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 11, pp. 2688–2700, Jun. 2013.
- [44] Y. Zhou, S. Zhang, L. Wu, and Y. Tian, "Predicting sectoral electricity consumption based on complex network analysis," *Appl. Energy*, vol. 255, Dec. 2019, Art. no. 113790.
- [45] G. Farin, "Algorithms for rational Bézier curves," *Comput. Des.*, vol. 15, no. 2, pp. 73–77, Mar. 1983.
- [46] H.-J. Hochfeld and M. Ahlers, "Role of Bézier curves and surfaces in the Volkswagen CAD approach from 1967 to today," *Comput.-Aided Des.*, vol. 22, no. 9, pp. 598–607, Nov. 1990.
- [47] M. H. Sohn and K. J. Lee, "Bezier curve application in the shape optimization of transonic airfoils," in *Proc. 18th Appl. Aerodyn. Conf.*, 2000, pp. 883–893.
- [48] R. Ponalagusamy, R. Narayanasamy, and P. Srinivasan, "Design and development of streamlined extrusion dies a Bézier curve approach," *J. Mater. Process. Technol.*, vol. 161, no. 3, pp. 375–380, Apr. 2005.
- [49] J.-W. Choi, R. Curry, and G. Elkaim, "Path planning based on Bézier curve for autonomous ground vehicles," in *Proc. Adv. Electr. Electron. Eng.-World Congr. Eng. Comput. Sci.*, Oct. 2008, pp. 158–166.
- [50] Z. Duraklı and V. Nabiyev, "A new approach based on Bézier curves to solve path planning problems for mobile robots," *J. Comput. Sci.*, vol. 58, Feb. 2022, Art. no. 101540.
- [51] J. Cao and D. Sui, "Well path design and optimization using composite cubic Bézier curves," *SPE J.*, vol. 2022, pp. 1–17, May 2022.
- [52] F. Zou, S. Tang, X. Li, and T. Yu, "Book image distortion correction based on Bézier curve model," in *Proc. IEEE 10th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, vol. 10, Jun. 2022, pp. 1959–1963.
- [53] A. S. D. Mata, "Complex networks: A mini-review," *Brazilian J. Phys.*, vol. 50, no. 5, pp. 658–672, Oct. 2020.
- [54] E. B. Koffman and P. A. T. Wolfgang, *Objects, Abstraction, Data Structures, and Design: Using C++*. Hoboken, NJ, USA: Wiley, 2006.
- [55] D. R. Lanning, G. K. Harrell, and J. Wang, "Dijkstra's algorithm and Google maps," in *Proc. ACM Southeast Reg. Conf. ACM SE*. Kennesaw, GA, USA: Association for Computing Machinery, Mar. 2014.
- [56] Y. Xu, Z. Wang, Q. Zheng, and Z. Han, "The application of Dijkstra's algorithm in the intelligent fire evacuation system," in *Proc. 4th Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, vol. 1, Aug. 2012, pp. 3–6.
- [57] M. Behún, D. Knežo, M. Cehlár, L. Knapčíková, and A. Behúnová, "Recent application of Dijkstra's algorithm in the process of production planning," *Appl. Sci.*, vol. 12, no. 14, p. 7088, Jul. 2022.
- [58] J. Luis, A. Luzania, J. Armando, B. Camacho, and R. N. González, "Curvas de Bézier," XXXI Semana Nacional de Investigación y Docencia en Matemáticas, Universidad de Sonora, Mexico, 2021.
- [59] P. E. Black, Ed., "Big-O notation," in *Dictionary of Algorithms and Data Structures*, Sep. 2019. Accessed: Sep. 23, 2023. [Online]. Available: <https://www.nist.gov/dads/HTML/bigOnotation.html>
- [60] J. Zhang and J. S. Wang, "Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator," *IEEE Access*, vol. 8, pp. 77013–77048, 2020.
- [61] F. Wilcoxon, "Probability tables for individual comparisons by ranking methods," *Biometrics*, vol. 3, no. 3, pp. 119–122, 1947.
- [62] Q. Liu, M. Liu, F. Wang, and W. Xiao, "A dynamic stochastic search algorithm for high-dimensional optimization problems and its application to feature selection," *Knowl.-Based Syst.*, vol. 244, May 2022, Art. no. 108517.
- [63] J. B. Cardoso and A. Barreiros, "Design optimization average-based algorithm," *Informatica*, vol. 44, no. 1, pp. 23–33, 2017.
- [64] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA J.*, vol. 29, no. 11, pp. 2013–2015, Nov. 1991.



KARLA AVILA is currently pursuing the master’s degree in computer engineering and in electronic and computer engineering. She is also pursuing the Ph.D. degree in electronic and computer engineering, with a specialization in automatic control and intelligent systems. She has published numerous articles in JCR-indexed journals and several book chapters. Additionally, she has been a speaker and an organizer of scientific dissemination events. She has actively participated in various programs that foster gender equality and early scientific vocations. Her current research interests include the design, development, application of artificial intelligence techniques, complex systems modeling, and metaheuristic algorithms for solving problems.



ERIK CUEVAS received the B.S. degree in electronics and communication engineering from Universidad de Guadalajara, in 1996, the M.E. degree in applied electronics from ITESO, in 1998, and the Ph.D. degree in artificial intelligence from Freie Universität Berlin, Germany, in 2007. He is currently a Professor with the University of Guadalajara, Mexico. His research interests include metaheuristics and evolutionary algorithms in a wide range of applications, such as image processing and machine learning. Also, he serves as an Editorial Board Member or an Associate Editor for *Expert System with Applications*, *Applied Soft Computing*, *Applied Mathematical Modeling*, *Artificial Intelligence Review*, *International Journal of Machine Learning and Cybernetics*, *ISA Transactions*, *Neural Processing Letters*, and *Mathematics and Computers in Simulation*. He is a member of the Mexican National Research System (SNI III).



MARCO PEREZ (Senior Member, IEEE) received the B.S. degree (Hons.) in electronics and communications engineering from the University of Guadalajara, Mexico, in 1995, the M.Sc. degree in industrial electronics from ITESO University, Mexico, in 2000, and the Ph.D. degree from the University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., in 2004. He was a Research Consultant in robotic manipulators with TQ-U.K. Ltd., U.K., in 2003. Since 2005, he has been with the University of Guadalajara, where he is currently a Professor and the Head of the University Center of Exact Sciences and Engineering CUCEI. Additionally, he has been a member of the Mexican National Research System (SNI), since 2007. His current research interests include robotics and computer vision, especially visual servoing and humanoid walking control.



RAM SARKAR (Senior Member, IEEE) received the B.Tech. degree in computer science and engineering from the University of Calcutta, India, in 2003, and the M.E. degree in computer science and engineering and the Ph.D. (Engineering) degree from Jadavpur University, Kolkata, India, in 2005 and 2012, respectively. He joined the Department of Computer Science and Engineering, Jadavpur University, as an Assistant Professor, in 2008, where he is currently a Professor. He received the Fulbright-Nehru Fellowship (USIEF) for postdoctoral research with the University of Maryland, College Park, USA, in 2014 and 2015. He has published more than 300 research papers in various journals and conference proceedings. His research interests include image and video processing, optimization algorithms, and deep learning. He is a member of ACM.

...