Wiley
Journal of Optimization
Volume 2024, Article ID 5546940, 21 pages
https://doi.org/10.1155/2024/5546940

WILEY

Research Article

MOBBO: A Multiobjective Brown Bear Optimization Algorithm for Solving Constrained Structural Optimization Problems

Pranav Mehta,[1] **Sumit Kumar**,[2] **Ghanshyam G. Tejani**,[3,4] **and Mohammad khishe**[5,6,7]

[1]*Department of Mechanical Engineering, Dharmsinh Desai University, Nadiad 387001, Gujarat, India*
[2]*Centre for Maritime Engineering and Hydrodynamics, Australian Maritime College, College of Sciences and Engineering, University of Tasmania, Launceston 7248, Australia*
[3]*Research and Development, Ethics Infotech, Vadodara 390007, Gujarat, India*
[4]*Jadara University Research Center, Jadara University, Irbid, Jordan*
[5]*Department of Electrical Engineering, Imam Khomeini Naval Science University of Nowshahr, Nowshahr, Iran*
[6]*Innovation Center for Artificial Intelligence Applications, Yuan Ze University, Taoyuan, Taiwan*
[7]*Applied Science Research Center, Applied Science Private University, Amman 11937, Jordan*

Correspondence should be addressed to Mohammad khishe; m_khishe@alumni.iust.ac.ir

Received 25 June 2024; Accepted 25 September 2024

Academic Editor: Himadri Majumder

The multiobjective (MO) optimizers show great promise in solving constrained engineering structural problems. This paper introduces a MO version of the Brown Bear Optimization (BBO) algorithm, inspired by the foraging behavior of brown bears. The proposed Multiobjective Brown Bear Optimization (MOBBO) algorithm is applied to five structural optimization problems, including 10-bar, 25-bar, 60-bar, 72-bar, and 942-bar trusses, aiming to minimize both mass and maximum nodal deflection simultaneously. Comparative evaluations against six benchmark algorithms demonstrate MOBBO's superior convergence, solution diversity, and effectiveness in addressing highly constrained problems. The hypervolume (HV) and inverted generational distance (IGD) metrics place MOBBO in first rank according to the Friedman test, with an average standard deviation of 0.0002. Moreover, the spacing-to-extent (STE) and generational distance (GD) metrics rank MOBBO second. The final Friedman rank test highlights MOBBO's overall dominance, achieving a first rank. Best Pareto plots, diversity graphs, and box plot analyses further suggest MOBBO's superior performance and convergence compared to existing algorithms. Therefore, the MOBBO algorithm can be effectively applied to various MO optimization tasks in industry, offering refined global optimization solutions and contributing valuable insights to the field of MO algorithms.

Keywords: metaheuristics; multiobjective optimization; truss design

1. Introduction

Classical optimization techniques have been used by researchers and industries for many years to solve real-world challenges and optimize critical design issues. These techniques may offer global or local optimum solutions and provide diverse outcomes. However, solving multimodal, multiobjective (MO), and highly constrained problems is challenging with classical optimization techniques due to their low convergence rate, imbalance between the exploration and exploitation phases, and tendency to converge to local optima. To address these limitations, nature-inspired algorithms based on human activities, swarm intelligence, physical phenomena, and evolutionary concepts—known as metaheuristics (MHs) algorithms—have been developed [1–3]. These MHs are capable of finding global optimum solutions with less computational time, balanced exploration and exploitation, and accurate results. Consequently, MHs can solve a wide range of design optimization challenges, including but not limited to manufacturing problems, fuzzy systems, structural optimization, transportation problems, scheduling challenges, power system optimization, and automobile systems [4–7]. However, despite the continuous development of new MHs, the complexity of problems has also increased with the advancement of new technologies. As a result, many MHs tend to find local optima, exhibit an improper balance between exploration and exploitation, and struggle with highly constrained problems. To overcome these shortcomings, enhancements to existing optimizers—such as hybrid algorithms, oppositional-based techniques, chaotic maps, Lévy flight strategies, and oppositional learning methods—have been introduced [8–12]. These modified or hybrid algorithms have proven to be more efficient and effective compared to the original MHs for solving complex optimization tasks.

1.1. MO Optimization Algorithms and Literature Survey. MO optimization is an important task where, instead of a single fitness function, a problem or system involves multiple objective functions with constraints and design variables. Accordingly, the MO form of MH algorithms is crucial for solving constrained problems with multiple fitness functions. In MO problems, a set of Pareto-front solutions exists, and the goal of a proposed algorithm is to produce solution sets that closely approximate the Pareto-front with minimal deviations [13, 14]. This article develops a MO version of a newly established MH algorithm, the BBO algorithm, to solve five critical structural optimization problems. The BBO algorithm is inspired by the foraging, territory establishment, and sniffing behaviors of brown bears [15]. The developed algorithm is applied to five structural design challenges, ranging from 10-bar to 942-bar trusses, to minimize mass and nodal deflection as fitness functions for each problem.

Various MO versions of well-known MH algorithms have been developed and applied to structural problems, such as the MOSOS algorithm [16]. Several bio-inspired MO algorithms and MHs have been developed to address real-world engineering challenges [17–20]. This article compares the results obtained from the MOBBO with those of six benchmark MO algorithms. Additionally, hypervolume (HV) analysis, generation-IGD, and STE metrics analysis were used to assess the performance of the developed algorithm. Moreover, diversity curves, HV, and boxplot analysis further deepen the study by evaluating MOBBO's performance.

Researchers have studied decomposition-based MO symbiotic organism search algorithms for optimizing truss structures, such as the 37-bar, 60-bar, 72-bar, 120-bar, and 200-bar trusses. Augmenting decomposition techniques helps achieve converged solutions and provides superior performance compared to 10 benchmark algorithms. The proposed algorithm also outperforms MOSOS [21]. Effective grouping of truss structures was identified using MO structural optimization techniques, addressing two conflicting objective functions: the weight of the structure and the discrete cross-sectional areas of the members. Comparisons were made between sixteen well-established MO algorithms, which helped identify and optimize key structural members [22].

Additionally, hybrid grey wolf and cuckoo search optimizers have been applied in a MO approach. The proposed hybrid MO-grey wolf-cuckoo search algorithm was tested with CEC 2020 benchmarks and used to optimize critical truss structures, demonstrating accuracy and diversity in the results [23]. Furthermore, 60-bar and 200-bar intermediate truss structures were optimized for weight and nodal displacement using the MO-Lichtenberg optimizer [24]. A novel MO water strider algorithm (MOWSA) was proposed to optimize eight different trusses, from the 10-bar to the 942-bar truss. The results were compared with nine benchmark algorithms, revealing that MOWSA outperformed the others in terms of Pareto fronts, convergence, and solution trade-offs [25].

A novel MOLCA (MO lever cancer optimizer) was proposed, incorporating random oppositional–based learning (ROBL) to enhance local and global search capabilities, along with IFM (information feedback mechanism), NDS (nondominated sorting), and crowding distance selection to achieve effective Pareto optimal fronts [26]. In addition to MO versions, researchers have developed hybridized and improved MO algorithms to enhance truss optimization results. For instance, a unique hybrid MO-SHADE-MRFO (MO-success history-based parameter adaptive differential evolution with manta ray foraging optimizer) was proposed to optimize six benchmark truss designs. The results were analyzed using HV GD, IGD, and FSTE metrics, showing that enhancing SHADE improves exploration capabilities, while MRFO maintains balance between exploration and exploitation [27].

An interesting study explored the effectiveness of NSGA-II and its nine variants in terms of convergence, solution quality, and diversity. Each variant was applied to optimize six truss structures, with weight and nodal displacement as the objective functions. The results showed that NSGA-II, restricted NSGA-II, grid-based NSGA-II, and ARSBX (adaptive real-coded simulated binary crossover)-NSGA-II outperformed the other five NSGA-II variants [28]. Furthermore, a novel math-inspired MOEDO (MO exponential distribution algorithm) was developed for precise and global optimization of engineering problems, including structural optimization [29]. Additionally, DSC-MOAGDE, DSC-MOSOS, and IMOMRFO are among the unique modified MO algorithms proposed [30–32].

1.2. Novelty of the Study. Novelty of the Study: This article presents the development of the MOBBO algorithm for optimizing two critical parameters of trusses: weight and nodal displacement. The BBO algorithm is a promising optimizer that has demonstrated effective results in addressing various engineering challenges. The MO version of the BBO algorithm represents a significant enhancement within the BBO algorithm family. The MOBBO algorithm is applied to optimize five benchmark truss designs. The metrics analysis, including HV, GD, IGD, and STE, further supports the novel findings and underscores the merits of the proposed algorithm in the field of MO optimization.

The rest of the article is structured as follows: Section 2: Overview of the BBO algorithm; Section 3: MOBBO; Section 4: Practical Assessment; Section 5: Conclusion.

2. Understanding of BBO

The BBO algorithm is inspired by several intelligent behaviors of brown bears, such as following the group, identifying food locations, and establishing their territory. These behaviors are primarily based on their pedal scent marking and sniffing actions. The algorithm has been developed and applied to the economic dispatch problem in power transmission systems, demonstrating its capability to balance exploration and exploitation phases while handling complex problems. The mathematical structure of the BBO algorithm is discussed in the following subsection.

2.1. Initialization-Group Formation. In this phase, initial random solution sets, representing groups of brown bears, are generated, with their marked pedal scent marks treated as decision variables within the solution sets. The mathematical representation of a randomly selected group of brown bears within their specific territory is given by equation (1) [15].

$$P_{i,j} = P_{i,j}^{\min} + \lambda \left(P_{i,j}^{\max} - P_{i,j}^{\min}\right), \tag{1}$$

where random number is indicated with the notation λ ranging from 0 to 1. Also, i-th pedal mark of j-th group is denoted by $P_{i,j}$.

2.1.1. Technique of Pedal Scent Marking. This is one of the most impressive walking techniques observed in brown bears. It involves a distinctive manner of walking, where they twist their feet to avoid prior depressions on the ground and carefully step toward the targeted location. This specialized walking behavior is most commonly observed in male brown bears. Mathematically, this behavior can be modeled as shown in (2) [15].

$$P_{i,j,k}^{\text{new}} = P_{i,j,k}^{\text{old}} - \left(\theta_k \alpha_{i,j,k} P_{i,j,k}^{\text{old}}\right), \tag{2}$$

where updated pedal scent mark is indicated by $P_{i,j,k}^{\text{new}}$ at k-th iteration of i-th group created j-th pedal mark. Moreover, occurrence factor and random number ranging from 0 to one is indicated with θ_k and $\alpha_{i,j,k}$ respectively. The careful stepping characteristic involves repeating pedal mark impressions by verifying previously marked pedals. This behavior helps to alert other group members effectively. Equation (3) presents the mathematical formulation of the careful stepping technique observed in brown bears.

$$P_{i,j,k}^{\text{new}} = P_{i,j,k}^{\text{old}} + F_k\left(P_{j,k}^{\text{best}} - L_k P_{j,k}^{\text{worst}}\right). \tag{3}$$

In (3), step factor is denoted with F_k, along with j-th best and worst pedal mark at k-th iteration is denoted with $P_{j,k}^{\text{best}}$ and $P_{j,k}^{\text{worst}}$, respectively. L_k indicates length of the step at particular iteration.

A third unique walking behavior observed in brown bears is the twisting of their feet. Male brown bears typically twist their feet into previously formed pedal marks, making them deeper and more pronounced for easier identification. The selection of pedal marks is based on the worst and best pedal scent marks determined in the previous iteration. The mathematical representation of the twisting feet behavior is given by equation (4).

$$P_{i,j,k}^{\text{new}} = P_{i,j,k}^{\text{old}} + \omega_{i,k}\left(P_{j,k}^{\text{best}} - P_{i,j,k}^{\text{old}}\right) - \omega_{i,k}\left(P_{j,k}^{\text{worst}} - P_{i,j,k}^{\text{old}}\right), \tag{4}$$

where angular velocity of the feet is denoted with $\omega_{i,k}$ at i-th pedal mark and k-th iteration.

2.2. Sniffing Etiquette. This interactive behavior of brown bears involves sniffing to follow the pedal scent marks of their group members in the right direction. Additionally, they use sniffing to establish their own territory and avoid being misled by the pedal scent marks of other bears. The mathematical model for the sniffing behavior is given by (5) [15].

$$P_{m,j,k}^{\text{new}} = \begin{cases} P_{m,j,k}^{\text{old}} + \lambda_{j,k}\left(P_{m,j,k}^{\text{old}} - P_{n,j,k}^{old}\right) if\ f\left(P_{m,k}^{old}\right) < f\left(P_{n,k}^{old}\right), \\ P_{m,j,k}^{\text{old}} + \lambda_{j,k}\left(P_{n,j,k}^{\text{old}} - P_{m,j,k}^{old}\right) if\ f\left(P_{n,k}^{old}\right) < f\left(P_{m,k}^{old}\right), \end{cases} \tag{5}$$

where $\lambda_{j,k}$ is the random number that is evenly distributed and ranging in the range of 0–1. Also, $P_{m,j,k}^{new}$ is the updated pedal scent mark location with $m \neq n$. Moreover, $P_{m,k}^{old}$ and $P_{n,k}^{old}$ corresponds to fitness function value at k-th iteration of m and n groups, respectively. The updating process for all the stages described is applied to each group of brown bears until the necessary criterion is met. The pseudocode of the BBO algorithm is shown below.

START

Define objective function f(P), population size (m), set number of design variables (n), limits on design variables (LB, UB), and set termination criterion (‘FEmax’, or ‘gmax’); where f(P) is the objective function and ‘P’ is the design vector. The brown bears’ group are considered to be a part of the population ($i = 1,2,\ldots,$m) and the bears’ levels of the group are considered as the design variables ($j = 1,2,\ldots,$n)./∗ Initialization/∗

Initialize the randomly generated set of the population within its upper and lower bounds and evaluate it.

$P_{i,j} = P_{i,j}^{\min} + \lambda\ (P_{i,j}^{\max} - P_{i,j}^{\min})$/∗ ***Initialize population***/∗

λ is any random number evenly distributed in the range [0; 1].

Arrange the population in ascending order of $f(P_i)$ values and select the best solution (P_i^{best}) and the worst solution

FE = 0/∗ **Functional Evaluations (FE)** ∗/

for $k = 1$ to $g_{\max}$ **do**/∗ **Initialize the optimization loop** ∗/

(P_i^{wrost}).

$\theta_k = k/g_{\max}$/∗ the occurrence factor∗/

for $i = 1$ to m **do**/∗ **Start pedal scent marking behavior phase**/∗

% Characteristic gait while walking

if $\theta_k > 0$ && $\theta_k \le g_{\max}/3$/∗ Characteristic gait while walking ∗/

$P_{i,k}^{new} = P_{i,k}^{old} - (\theta_k \alpha_{i,k} P_{i,k}^{old})$/∗ ***Generate new population*** ∗/

elseif $\theta_k > g_{\max}/3$ && $\theta_k \le 2g_{\max}/3$/∗ **Careful stepping characteristic** ∗/

$P_{i,k}^{new} = P_{i,k}^{old} + F_k (P_{i,k}^{best} - L_k P_{i,k}^{worst})$;/∗ **Generate new population** ∗/

/∗ *where,* $F_k = \beta_{1,k}.\theta_k$, $L_k = 1 + \beta_{2,k}$, *and* $\beta_{1,k}$ *&* $\beta_{2,k}$ *is any random number in the range [0;1]* ∗/

elseif $\theta_k > 2g_{\max}/3$ && $\theta_k \le 1$/∗ **Twisting feet characteristic**∗/

$P_{i,k}^{new} = P_{i,k}^{old} + \omega_{i,k} (P_{j,k}^{best} - P_{i,k}^{old}) - \omega_{i,k} (P_{i,k}^{worst} - P_{i,k}^{old})$;

/∗ where, $\omega_{i,k} = 2\pi.\theta_k.\gamma_{i,k}$, $\gamma_{i,k}$ is an evenly distributed random number in the range [0, 1] ∗//∗ Generate new population ∗/

/∗ Select better group of bears. ∗/

if $F(X_i^{new}) < F(X_i^{old})$ **then**/∗ **Greedy selection**/∗

$P_{i,k} = P_{i,k}^{new}$

else

$P_{i,k} = P_{i,k}^{old}$

end if

/∗ Select better group of bears. ∗/

FE=FE+1;/∗ Count Function evaluation/∗

end if

end for/∗Pedal scent marking behavior ends ∗/

for $i = 1$ to m **do**/∗ **Sniffing behavior starts** ∗/

Select two random group of bears $P_{m,k}^{old}$ ***&*** $P_{n,k}^{old}$ ***where*** $n \ne m$.

$$P_{m,k}^{new} = \begin{cases} P_{m,k}^{old} + \lambda_{j,k} (P_{m,k}^{old} - P_{n,k}^{old})\, if\ f(P_{m,k}^{old}) < f(P_{n,k}^{old}) \\ P_{m,k}^{old} + \lambda_{j,k} (P_{n,k}^{old} - P_{m,k}^{old})\, if\ f(P_{n,k}^{old}) < f(P_{m,k}^{old}) \end{cases}$$

/∗ $\lambda_{j,k}$ is an evenly distributed random number in the range [0, 1] ∗/

/∗ Select better group of bears. ∗/

if $F(X_i^{new}) < F(X_i^{old})$ **then**/∗ **Greedy selection**/∗

$P_{i,k} = P_{i,k}^{new}$

else

$P_{i,k} = P_{i,k}^{old}$

end if

FE=FE=1;/∗ Count Function evaluation/∗

end for/∗ ***Sniffing behavior ends*** ∗/

Arrange the population in ascending order of $f(P_i)$ values and select the best solution (P_i^{best}) and the worst solution

if $FE \ge FE_{\max}$ Or $k = g_{\max}$ **then**/∗ **Termination criterion**/∗

break optimization loop

end if

$k = k+1$;

end for/∗ **Optimization loop ends**/∗

Display best solutions & store results

STOP

3. MOBBO Algorithm

MOBBO is an extension of the BBO algorithm originally designed for single-objective optimization problems. The MOBBO algorithm is based on the concept of dominance criteria for identifying sets of nondominated and dominated solutions. Nondominated solutions are determined using dominance strategies, which involve identifying the best solution while minimizing the detriment to other solutions for a particular objective function. Accordingly, the comparison was made amongst two design solution sets, X_2 and X_2 corresponds to function vectors f_1 and f_2. For instance, maximization of fitness function if X_1 dominates over X_2 then all elements in f_1 are greater than or equal to their corresponding elements in f_2. Solutions not dominated by any others in the set are deemed nondominated and stored in an external archive. This archive is utilized to construct the Pareto front, which represents the optimal set of trade-off solutions.

MOBBO uses an external archive for MO optimization to preserve nondominated solutions, unlike the single-objective BBO algorithm. The ε-dominance-based updating approach is employed to identify dominance relationships within this repository. This technique divides the solution space into boxes of different shapes based on the number of objectives. Solutions are grouped within these boxes, and the dominant solutions in each box are retained while others are removed. This process significantly enhances the diversity and quality of solutions by ensuring that only nondominated solutions are stored in the archive. The method employs a grid-based technique and a fixed-size archive to store the best solutions found during each update. The ε-dominance approach is used regularly to maintain and update the collection of nondominated solutions. This strategy enables MOBBO to systematically explore the MO search space and progressively approach well-distributed Pareto fronts.

The pseudocode of the BBO algorithm is shown below.

START

Define objective function f(P), population size (m), set number of design variables (n), limits on design variables (LB, UB), and set termination criterion ('FEmax', or 'gmax'); where f(P) is the objective function and 'P' is the design vector. The brown bears' group are considered to be a part of the population ($i = 1, 2, \ldots,$ m) and the *bears' levels of the group are considered as the design variables (j = 1, 2, . . ., n).* /∗ ***Initialization***/∗

Initialize the randomly generated set of the population within its upper and lower bounds and evaluate it.

$P_{i,j} = P_{i,j}^{\min} + \lambda (P_{i,j}^{\max} - P_{i,j}^{\min})$/∗ ***Initialize population***/∗

λ is any random number evenly distributed in the range [0; 1].

Perform Non-Dominated Sorting and identify the best solution, select the best solution (P_i^{best}), and the worst solution (P_i^{wrost}).

FE = 0/∗ **Functional Evaluations (FE)** ∗/

for $k = 1$ to $g_{\max}$ **do/∗ Initialize the optimization loop** ∗/

/∗ Pareto sorting/∗

$\theta_k = k/g_{\max}$/∗ the occurrence factor∗/

for $i = 1$ to m **do/∗ Start pedal scent marking behavior phase/∗**

% Characteristic gait while walking

if $\theta_k > 0$ && $\theta_k \le g_{\max}/3$/∗ Characteristic gait while walking ∗/

$P_{i,k}^{new} = P_{i,k}^{old} - (\theta_k \alpha_{i,k} P_{i,k}^{old})$/∗ ***Generate new population*** ∗/

elseif $\theta_k > g_{\max}/3$ && $\theta_k \le 2g_{\max}3$/∗ **Careful stepping characteristic** ∗/

$P_{i,k}^{new} = P_{i,k}^{old} + F_k (P_{i,k}^{best} - L_k P_{i,k}^{worst})$;

/ ∗ *where,* $F_k = \beta_{1,k}.\theta_k$, $L_k = 1 + \beta_{2,k}$, *and* $\beta_{1,k}$& $\beta_{2,k}$ *is any random number in the range [0;1]* ∗/

elseif $\theta_k > 2g_{\max}/3$ && $\theta_k \le 1$/∗ **Twisting feet characteristic**∗/

$P_{i,k}^{new} = P_{i,k}^{old} + \omega_{i,k} (P_{j,k}^{best} - P_{i,k}^{old}) - \omega_{i,k} (P_{i,k}^{worst} - P_{i,k}^{old})$;

/ ∗ *where,* $\omega_{i,k} = 2\pi.\theta_k.\gamma_{i,k}$, $\gamma_{i,k}$ *is an evenly distributed random number in the range [0, 1]* ∗/

/∗ Select better group of bears. ∗/

if $F(X_i^{new}) < F(X_i^{old})$ **then/∗ Greedy selection/∗**

$P_{i,k} = P_{i,k}^{new}$

else

$P_{i,k} = P_{i,k}^{old}$

end if

/∗ Select better group of bears. ∗/

FE=FE+1;/∗ Count Function evaluation/∗

end if

end for/∗Pedal scent marking behavior ends ∗/

for $i = 1$ to m **do/∗ Sniffing behavior starts ∗/**

Select two random group of bears $P_{m,k}^{old}$ ***&*** $P_{n,k}^{old}$ ***where*** $n \ne m$.

$$P_{m,k}^{new} = \begin{cases} P_{m,k}^{old} + \lambda_{j,k} (P_{m,k}^{old} - P_{n,k}^{old})\, if\ f(P_{m,k}^{old}) < f(P_{n,k}^{old}) \\ P_{m,k}^{old} + \lambda_{j,k} (P_{n,k}^{old} - P_{m,k}^{old})\, if\ f(P_{n,k}^{old}) < f(P_{m,k}^{old}) \end{cases}$$

/∗ $\lambda_{j,k}$ is an evenly distributed random number in the range [0, 1] ∗/

/∗ Select better group of bears. ∗/

if $F(X_i^{new}) < F(X_i^{old})$ **then/∗ Greedy selection/∗**

$P_{i,k} = P_{i,k}^{new}$

else

$P_{i,k} = P_{i,k}^{old}$

end if

FE=FE=1;/∗ Count Function evaluation/∗

end for/∗ Sniffing behavior ends ∗/

Perform Non-Dominated Sorting, identify the best solution, select the best solution (P_i^{best}), the worst solution (P_i^{wrost}) & sore in External Archive

if $FE \ge FE_{\max}$ Or $k = g_{\max}$ **then/∗ Termination criterion/∗**

break optimization loop

end if

$k = k+1$;

end for/∗ Optimization loop ends/∗

Display Pareto Optimal Set & store results

STOP

3.1. Design Aspect for Structural Optimization of Trusses. MO structural optimization involves minimizing the structural weight of the overall truss while simultaneously controlling the maximum nodal deflection. The mathematical formulations for the population sets, weight optimization objective, and nodal deflection objective are given in equations (6)–(8).

Population sets:

$$A = \{A_1, A_2, \ldots, A_m\}. \tag{6}$$

To minimize the truss weight:

$$f_1(A) = \sum_{i=1}^{m} A_i \rho_i L_i. \tag{7}$$

To maximize maximum nodal deflection:

$$f_2(A) = \max\left(\left|\delta_j\right|\right). \tag{8}$$

Subject to:
Behavior constraints:
Stress constraints,

$$g(A) = |\sigma_i| - \sigma_i^{\max} \le 0. \tag{9}$$

Side constraints:
Cross – sectional area constraints,

$$A_i^{\min} \le A_i \le A_i^{\max}, \tag{10}$$

where σ_i is the permissible stress with weight density and bar length of truss element is denoted by ρ_i and L_i, respectively. Design vector is denoted with A. The permissible upper and lower limits are signified by superscripts 'max' and 'min,' respectively.

Five different truss structures, as shown in Figure 1 (a, b, c, d, and e), are optimized using MOBBO and six benchmark algorithms. The design configurations for each truss structure (10-bar, 25-bar, 60-bar, 72-bar, and 942-bar trusses) are detailed in Table 1. Table 1 provides information on design variables, including truss elements, stress conditions as constraints, density for weight reduction, and loading conditions on each node. Additionally, the table records the elemental member groupings and size numberings for truss structures ranging from the 25-bar to the 942-bar truss.

3.2. Constraint Handling Technique. MOBBO uses a static penalty approach to handle the critical constraints of each truss design [33, 34]. The mathematical formulation for this approach is given by (11).

$$f\_j\,(X) = \begin{cases} f_j\,(X) \text{ no constraint violation,} \\ f_j\,(X) * \left(1 + \varepsilon_1 * C\right)^{\varepsilon_2}, C = \sum_{i=1}^{q} C_i,\ C_i = \left|1 - \dfrac{p_i}{p_i^*}\right|, \text{ otherwise,} \end{cases} \tag{11}$$

where p_i is a value of constraint infringement with reference to bound p_i^*. The constants values, ε_1 and ε_2 are taken 3 based on experiments.

The computational complexity of the MOBBO algorithm is largely influenced by the size of the population, the number of objectives, and the number of iterations. The most computationally expensive operations are usually those related to Pareto front identification, which results in the complexity. This complexity indicates that as the population size and the number of objectives increase, the computational demand of the algorithm grows significantly, particularly in MO optimization problems with large populations and many objectives.

4. Practical Assessment

The proposed MOBBO optimizer is tested for structural optimization of five different complex truss structures ranging from 10-bar truss to 942 bars including 25-bar, 60-bar, and 72-bar trusses. Additionally, opted results in terms of statistics were compared with well-known and benchmark six MO versions of algorithms namely: MOALO [35], NSGA-II [36], MOWCA [37], MOBA [38], DEMO [39], and MODA [40]. The compared algorithms are competitive with MOBBO and assist in identifying effectiveness and dominance of the proposed algorithm. The parameter settings in the experimental section were tested five times using different sets. The best performance for each algorithm was considered for this study. Each algorithm was independently executed 30 times for each design example, using a population size of 100 and 50,000 function evaluations (FEs). Accordingly, Table 1 designates design variables, constant parameters and loading conditions for each individual truss structure. Moreover, Figure 1 shows 3-D structural diagram of analyzed truss structures including loading conditions, dimensions and parametric constraints.

4.1. Performance Assessments

- The HV indicator measures the volume of the HV dominated by the set of nondominated solutions (i.e., the Pareto front) found by the algorithm. It quantifies the portion of the objective space covered by the obtained solutions that are not dominated by any other solution. Higher values of HV indicate better performance, as they represent a larger portion of the Pareto front covered by the algorithm's solutions.
- GD measures the average distance from a set of solutions produced by the algorithm to a reference set of true Pareto-optimal solutions in the objective space. It provides an indication of how well the algorithm has converged to the Pareto front. Smaller values of GD indicate better convergence. IGD, as the name suggests, is the inverse of GD. It measures how close the solutions produced by the algorithm are to the true Pareto front. Therefore, smaller values of IGD imply better performance, indicating that the algorithm's solutions are closer to the Pareto front.
- The STE test evaluates the diversity and spread of solutions along the Pareto front. It calculates the ratio of the average distance between consecutive solutions along the Pareto front to the extent of the Pareto front. A higher STE ratio suggests better diversity and spread among the solutions. This test helps assess whether the algorithm is producing solutions that are evenly distributed along the Pareto front or if they are clustered in certain regions.
- Box-plot analysis, diversity curves, and HV vs. functional graphs are plotted to further demonstrate the proposed algorithm's capabilities to converge on critical problems and effectively optimize the solution.

4.2. Results and Discussion. In this section, statistical results attained by MOBBO for the each tests were discussed along with comparison of the data with other considered algorithms. For each statistical tests, maximum, minimum, average, standard deviation, and Friedman rank was attained and comparison was made to confirm the effectiveness of MOBBO for global optimization.

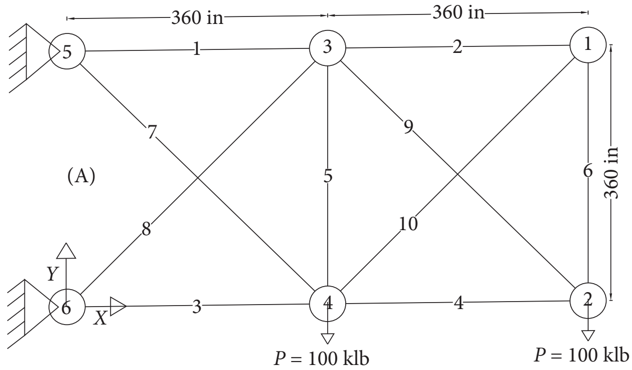
4.2.1. HV Analysis for Truss Structures. Table 2 shows the results of the HV test for all compared algorithms, including MOBBO, across various truss configurations. Bold-faced values indicate results achieved by MOBBO. For the 10-bar structure, MOBBO achieves a maximum objective function value of 59,973, which is better than all other algorithms. Additionally, MOBBO obtains a superior Friedman rank of 1 with the least standard deviation of 239,

Table 1: Design concerns for the truss problems.

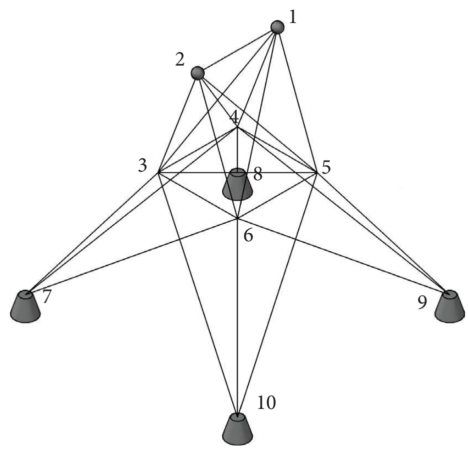
Structures	Design variables	Constraints (σ^{max} in ksi)	Density (ρ in lb/in^3)	Young modules (E in ksi)	Loading conditions (kips)
The 10-bar truss	$X_{i,} i = 1, 2, \ldots, 10$	25	0.1	10,000	$P_{y2} = P_{y4} = -100$
The 25-bar truss	$X_{i,} i = 1, 2, \ldots, 8$	40		10,000	$P_{x1} = 1$, $P_{y1} = P_{z1} = P_{y2} = P_{z2} = -10$, $P_{x3} = 0.5$, $P_{x6} = 0.6$
The 60-bar truss	$X_{i,} i = 1, 2, \ldots, 25$	40	0.1	10,000	Case 1: $P_{x1} = -10$, $P_{x7} = 9$ Case 2: $P_{x15} = P_{x18} = -8$, $P_{y15} = P_{y18} = 3$ Case 3: $P_{x22} = -20$ and $P_{y22} = 10$
The 72-bar truss	$X_{i,} i = 1, 2, \ldots, 16$	25	0.1	10,000	Case 1:$F_{1x} = F_{1y} = 5$, $F_{1z} = -5$ Case 2: $F_{1z} = F_{2z} = F_{3z} = F_{4z} = -5$
The 942-bar truss	$X_{i,} i = 1, 2, \ldots, 59$	25	0.1	10,000	At each node: Vertical loading: Segment 1; $P_z = -3$ Segment 2; $P_z = -6$ Segment 3; $P_z = -9$ Lateral loading: Right side; $P_x = 1.5$ Left side; $P_x = 1.0$ Lateral loading: $P_y = 1.0$

Structures	Member grouping
The 10-bar truss	—
The 25-bar truss	X_1 (1,2); X_2 (1–4, 2–3, 1–5, 2–6); X_3 (2–5, 2–4, 1–3, 1–6); X_4 (3–6, 4–5); X_5 (3–4, 5–6); X_6 (3–10, 6–7, 4–9, 5–8); X_7 (3–8, 4–7, 6–9, 5–10); X_8 (3–7, 4–8, 5–9, 6–10)
The 60-bar truss	X_1 (49–60); X_2 (1,13); X_3 (2,14); X_4 (3,15); X_5 (4,16); X_6 (5,17); X_7 (6,18); X_8 (7,19); X_9 (8,20); X_{10} (9,21); X_{11} (10,22); X (11,23); X_{13} (12,24); X_{14} (25,37); X_{15} (26,38); X_{16} (27,39); X_{17} (28,40); X_{18} (29,41); X_{19} (30,42); X_{20} (31,43); X_{21} (32,44); X_{22} (33,45); X_{23} (34,46); X_{24} (35,47); X_{25} (36,48)
The 72-bar truss	X_1(1–4); X_2 (5–12); X_3 (13–16); X_4 (17,18); X_5 (19–22); X_6 (23–30); X_7 (31–34); X_8 (35,36); X_9 (37–40); X_{10} (41–48); X_{11} (49–52); X_{12} (53,54); X_{13} (55–58); X_{14} (59–66); X_{15} (67–70); X_{16} (71,72)
The 942-bar truss	X_1 (1,2); X_2 (3–10); X_3 (11–18); X_4 (19–34); X_5 (35–46); X_6 (47–58); X_7 (59–82); X_8 (83–86); X_9 (87–90); X_{10} (97–98); X_{11} (99–106); X_{12} (107–122); X_{13} (123–130); X_{14} (131–162); X_{15} (163–170); X_{16} (171–186); X_{17} (187–194); X_{18} (195–226); X_{19} (227–234); X_{20} (235–258); X_{21} (259–270); X_{22} (271–318); X_{23} (319–330); X_{24} (331–338); X_{25} (339–342); X_{26} (343–350); X_{27} (351–358); X_{28} (359–366); X_{29} (367–382); X_{30} (383–390); X_{31} (391–398); X_{32} (399–430); X_{33} (431–446); X_{34} (447–462); X_{35} (463–486); X_{36} (487–498); X_{37} (499–510); X_{38} (511–558); X_{39} (559–582); X_{40} (583–606); X_{41} (607–630); X_{42} (631–642); X_{43} (643–654); X_{44} (655–702); X_{45} (703–726); X_{46} (727–750); X_{47} (751–774); X_{48} (775–786); X_{49} (787–798); X_{50} (799–846); X_{51} (847–870); X_{52} (871–894); X_{53} (895–902); X_{54} (903–906); X_{55} (907–910); X_{56} (911–918); X_{57} (919–926); X_{58} (927–934); X_{59} (935–942)

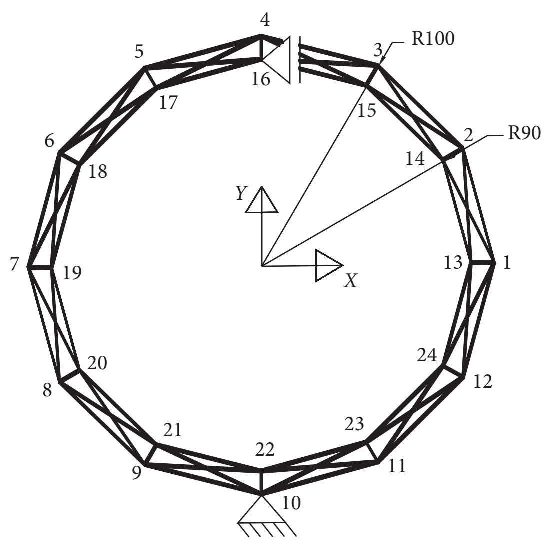
Structures	Size variables (in^2)
The 10-bar truss	[1.62, 1.8, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.8, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16, 16.9, 18.8, 19.9, 22, 22.9, 26.5, 30, 33.5]
The 25-bar truss	[1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3, 3.2, 3.4]
The 60-bar truss	[0.5, 0.6, 0.7, ..., 4.9]
The 72-bar truss	[0.1, 0.2, 0.3,..., 2.5]
The 942-bar truss	[1, 2, 3,..., 200]



(a)

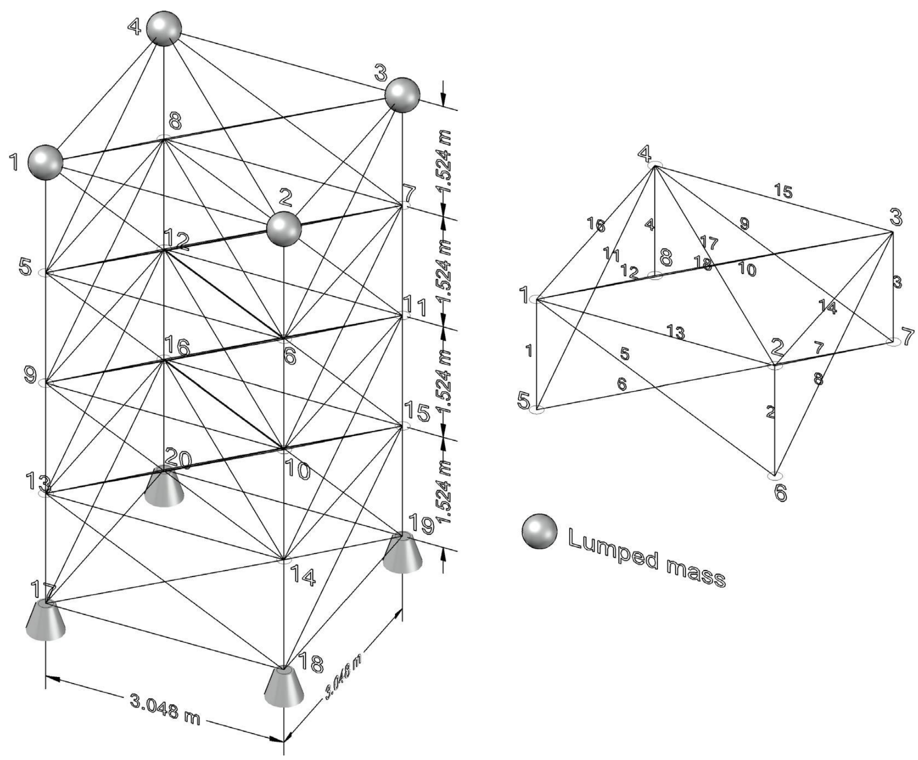


(b)



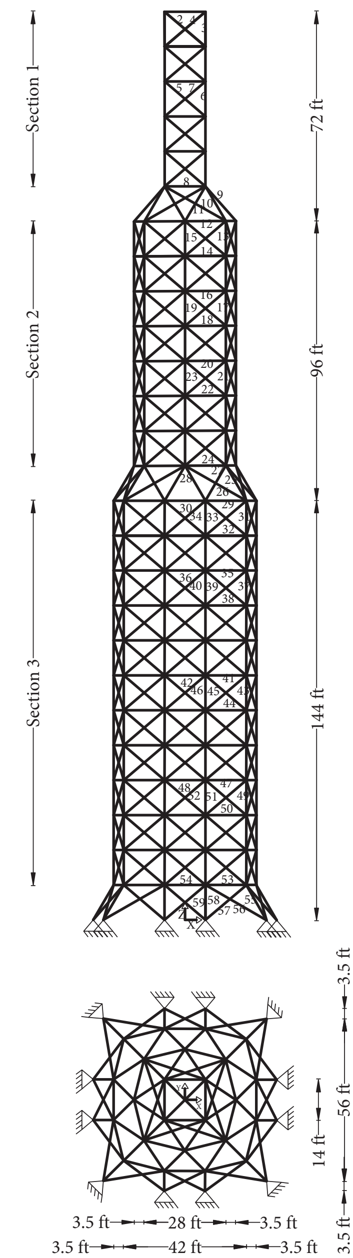
(c)

Figure 1: Continued.



(d)

FIGURE 1: Continued.



(e)

Figure 1: (a) 10-bar truss, (b) 25-bar truss, (c) 60-bar truss, (d) 72-bar truss, and (e) 942-bar truss.

compared to all other algorithms, indicating better convergence and minimal deviations. A similar scenario is observed for the 25-bar truss, where MOBBO exhibits the least deviations, with a value of 9, compared to other algorithms. For the 25-bar truss, MOBBO achieves a maximum objective function value of 1,956, which is 1.2%, 1.01%, and 3.9% better than MODA, MOBA, and DEMO, respectively. In this case, MOBBO has a Friedman rank of 1.03, followed by MODA and MOBA with ranks of 2.73 and 2.40, respectively.

For the 60-bar truss, MOBBO excels in terms of standard deviation with a value of 3, which is significantly lower than that of other algorithms. MOWCA and NSGA-II have minimum values of 85 and 2,438, respectively. MOBBO achieves a maximum objective function value of 4,111, which is 11.1%, 8.2%, 0.3%, 39.3%, 6.56%, and 3.8% better than MOALO, MODA, MOWCA, NSGA-II, DEMO, and MOBA, respectively. For the 72-bar truss, MOBBO realizes the most favorable standard deviation of 55, competitive with NSGA-II but significantly lower than MOWCA, which has a standard deviation of 288.

The 942-bar truss is particularly challenging to optimize due to its constraints. MOBBO achieves one of the most favorable standard deviation values for the HV test, with a value of 541,016. Additionally, MOBBO realizes a maximum fitness function value of 41,280,653, which is 4.3%,

Table 2: The HV of the considered truss structures.

HV		MOALO	MODA	MOWCA	MOBBO	NSGA_II	DEMO	MOBA
10-bar	Average	41,661	56,658	44,689	59,541	38,774	57,234	57,923
	Max	54,036	58,153	56,501	59,973	45,306	58,484	59,680
	min	26,026	53,284	13,537	58,981	0	54,034	56,393
	Std	6422	955	9620	239	8128	923	829
	Friedman	6.10	3.70	5.50	1.00	6.33	2.90	2.47
25-bar	Average	1402	1870	1068	1942	1641	1681	1893
	Max	1805	1931	1695	1956	1716	1882	1936
	min	919	1704	163	1918	1509	737	1823
	Std	261	50	497	9	44	268	25
	Friedman	5.83	2.73	6.60	1.03	5.00	4.40	2.40
60-bar	Average	3230	3468	3202	4117	2705	3639	3920
	Max	3695	3794	4111	4122	2916	3855	3961
	min	2475	3081	85	4110	2438	3404	3827
	Std	329	187	1191	3	101	114	32
	Friedman	5.27	4.60	4.10	1.00	6.70	4.03	2.30
72-bar	Average	2008	2045	1925	2010	1640	2098	2150
	Max	2190	2210	2212	2171	1759	2263	2236
	min	1738	1755	1207	1914	1553	1909	1897
	Std	139	114	288	55	55	81	69
	Friedman	4.00	3.50	4.40	4.43	6.83	2.90	1.93
942-bar	Average	35,043,988	31,077,869	32,023,740	40,376,435	25,364,659	33,692,954	31,608,886
	Max	37,891,766	32,936,248	34,830,945	41,280,653	26,940,679	36,135,237	34,550,582
	min	28,913,236	28,802,015	28,419,845	38,772,387	23,983,018	30,230,056	27,473,372
	Std	1,674,722	902,204	1,627,971	541,016	798,545	1,634,927	1,648,535
	Friedman	2.43	5.20	4.47	1.00	7.00	3.13	4.77
Average Friedman		4.73	3.95	5.01	1.69	6.37	3.47	2.77
Overall Friedman rank		5	4	6	1	7	3	2

Table 3: The GD metric of the considered truss structures.

GD		MOALO	MODA	MOWCA	MOBBO	NSGA_II	DEMO	MOBA
10-bar	Average	1.7545	4.8806	5.5589	4.2326	3.33E+08	4.6625	5.0069
	Max	3.5112	11.5492	7.9363	4.5119	1.00E+10	6.0712	5.7857
	min	0.7382	3.6190	2.5413	3.9115	4.3175	3.2262	3.9409
	Std	0.6821	1.7381	1.2889	0.1259	1.83E+09	0.7183	0.4895
	Friedman	1.03	3.67	5.07	2.93	6.90	3.80	4.60
25-bar	Average	0.1033	0.4172	0.5095	0.3786	1.2125	0.6415	0.5140
	Max	0.1641	0.6795	0.6976	0.6497	2.0225	2.0462	0.6516
	min	0.0583	0.2078	0.2869	0.2852	0.8281	0.2577	0.4098
	Std	0.0315	0.1151	0.1039	0.0671	0.2697	0.4822	0.0685
	Friedman	1.00	3.53	4.67	2.80	6.90	4.37	4.73
60-bar	Average	0.1263	0.3757	0.3127	0.4645	0.2919	0.6766	0.6315
	Max	0.2215	0.7467	0.7420	0.8227	0.9202	2.1146	2.0914
	min	0.0649	0.2972	0.0000	0.2728	0.0988	0.3055	0.2735
	Std	0.0407	0.0814	0.1909	0.1712	0.1655	0.4399	0.5299
	Friedman	1.37	4.17	3.43	4.90	2.93	6.23	4.97
72-bar	Average	0.3187	0.9545	0.9785	2.0781	2.8682	3.4568	0.9943
	Max	0.8288	1.5601	1.1481	3.8788	4.5004	8.2418	2.0693
	min	0.1030	0.6330	0.6465	0.9917	2.1003	0.8155	0.7531
	Std	0.1828	0.1771	0.1179	0.8032	0.5159	1.8999	0.2531
	Friedman	1.00	3.07	3.27	5.37	6.20	6.23	2.87
942-bar	Average	805.1905	1919.9638	2104.3739	1502.1881	7633.1544	1840.4447	3183.2252
	Max	1580.3798	2907.6639	2546.4594	1673.9293	11,473.2453	2102.8043	6118.9575
	min	359.4539	1583.9472	1774.8939	1356.6083	4946.4910	1124.9166	2127.5528
	Std	277.9754	264.6549	181.6167	74.0905	1601.2645	196.2857	842.6339
	Friedman	1.03	3.73	4.73	2.00	7.00	3.57	5.93
Average Friedman		1.09	3.63	4.23	3.60	5.99	4.84	4.62
Overall Friedman rank		1	3	4	2	7	6	5

Table 4: The IGD metric of the considered truss structures.

IGD		MOALO	MODA	MOWCA	MOBBO	NSGA_II	DEMO	MOBA
10-bar	Average	337.7851	65.1999	242.6063	10.7533	3.33E+08	24.0586	26.3277
	Max	428.8015	135.5445	686.8785	18.0493	1.00E+10	113.7475	51.5326
	min	105.0098	17.8933	89.2989	5.9684	183.2933	9.1740	4.6722
	Std	71.3183	28.0497	131.6239	2.6639	1.83E+09	23.5373	12.5089
	Friedman	6.67	3.93	5.50	1.27	5.80	2.20	2.63
25-bar	Average	31.0825	5.5220	34.2540	0.7006	12.0472	4.6953	1.7156
	Max	39.1653	16.7248	72.4983	1.5462	16.1233	15.9779	4.6814
	min	6.9256	0.9529	7.5047	0.3051	8.4898	0.9246	0.4990
	Std	7.4648	4.1291	21.0961	0.2530	1.9265	3.7638	0.8528
	Friedman	6.47	3.70	6.23	1.07	5.07	3.20	2.27
60-bar	Average	21.8304	11.5426	15.0571	0.4063	9.1679	3.5243	0.8292
	Max	30.5137	17.8789	61.6500	0.4886	10.8505	6.8207	1.5599
	min	6.1189	5.3564	0.6448	0.3163	7.3776	1.5637	0.4933
	Std	5.8297	3.3372	17.5291	0.0430	0.9009	1.3710	0.2542
	Friedman	6.67	5.43	4.77	1.00	4.77	3.33	2.03
72-bar	Average	42.7454	17.2242	26.6017	18.0953	37.9962	8.6316	5.6075
	Max	61.1874	33.8744	69.0456	23.6792	44.4788	18.9953	17.2069
	min	12.5160	7.1265	7.9792	9.8172	29.3506	2.1094	1.9515
	Std	14.3817	6.9514	17.2978	3.1619	4.2663	3.8530	2.8618
	Friedman	6.33	3.57	4.67	4.03	6.10	1.97	1.33
942-bar	Average	70,895.2758	63,269.4682	65,138.7225	5764.7019	86,768.1714	32,294.1375	63,795.3654
	Max	120,872.6063	76,067.8545	79,545.9719	14,708.9321	96,494.8847	52,695.5855	85,851.7687
	min	21,126.8742	41,499.8782	46,504.2621	1348.0140	77,461.3977	18,879.1596	40,106.5176
	Std	26,071.6203	7422.0750	10,400.0847	3120.8487	4933.3513	7347.5856	10,259.4185
	Friedman	5.17	4.37	4.37	1.00	6.60	2.13	4.37
Average Friedman		6.26	4.20	5.11	1.67	5.67	2.57	2.53
Overall Friedman rank		7	4	5	1	6	3	2

Table 5: The STE metric values obtained for the truss problems.

STE		MOALO	MODA	MOWCA	MOBBO	NSGA-II	DEMO	MOBA
10-bar	Average	0.0337	0.0116	0.0297	0.0058	3.33E+18	0.0063	0.0049
	Max	0.0655	0.0321	0.0849	0.0060	1.00E+20	0.0137	0.0059
	min	0.0000	0.0046	0.0026	0.0055	0.0064	0.0039	0.0038
	Std	0.0193	0.0073	0.0199	0.0001	1.83E+19	0.0019	0.0006
	Friedman	5.20	4.03	5.23	2.60	6.77	2.77	1.40
25-bar	Average	0.0304	0.0229	0.0286	0.0067	0.0565	0.0065	0.0050
	Max	0.0531	0.0575	0.0846	0.0071	0.1156	0.0162	0.0074
	min	0.0058	0.0056	0.0000	0.0064	0.0230	0.0020	0.0035
	Std	0.0187	0.0121	0.0230	0.0002	0.0259	0.0028	0.0012
	Friedman	4.97	4.73	4.67	2.80	6.57	2.63	1.63
60-bar	Average	0.0345	0.0173	0.0252	0.0060	0.0194	0.0069	0.0077
	Max	0.0515	0.0546	0.0995	0.0069	0.0368	0.0193	0.0222
	min	0.0015	0.0059	0.0000	0.0056	0.0065	0.0036	0.0046
	Std	0.0161	0.0106	0.0290	0.0003	0.0081	0.0032	0.0031
	Friedman	6.03	4.90	4.77	1.80	5.40	2.10	3.00
72-bar	Average	0.0328	0.0206	0.0162	0.0070	0.0557	0.0058	0.0079
	Max	0.0520	0.0457	0.0382	0.0074	0.1051	0.0086	0.0112
	min	0.0000	0.0062	0.0069	0.0065	0.0187	0.0042	0.0044
	Std	0.0192	0.0100	0.0081	0.0002	0.0185	0.0014	0.0018
	Friedman	5.03	4.97	4.60	2.37	6.77	1.57	2.70
942-bar	Average	0.0364	0.0145	0.0125	0.0066	0.1135	0.0073	0.0232
	Max	0.0710	0.0447	0.0238	0.0072	0.4678	0.0183	0.0537
	min	0.0007	0.0062	0.0060	0.0060	0.0240	0.0015	0.0077
	Std	0.0162	0.0084	0.0044	0.0003	0.0812	0.0035	0.0117
	Friedman	5.33	3.77	3.60	1.63	6.90	1.93	4.83
Average Friedman		5.31	4.48	4.57	2.24	6.48	2.20	2.71
Overall Friedman rank		6	4	5	2	7	1	3

Table 6: The Overall Friedman rank obtained for the truss problems.

	MOALO	MODA	MOWCA	MOBBO	NSGA_II	DEMO	MOBA	MOFA
10-bar	4.75	3.83	5.33	1.95	6.45	2.92	2.78	4.75
25-bar	4.57	3.68	5.54	1.93	5.88	3.65	2.76	4.57
60-bar	4.83	4.78	4.27	2.18	4.95	3.93	3.08	4.83
72-bar	4.09	3.78	4.23	4.05	6.48	3.17	2.21	4.09
942-bar	3.49	4.27	4.29	1.41	6.88	2.69	4.98	3.49
Average Friedman	4.35	4.07	4.73	2.30	6.13	3.27	3.16	4.35
Overall Friedman rank	5	4	6	1	7	3	2	5

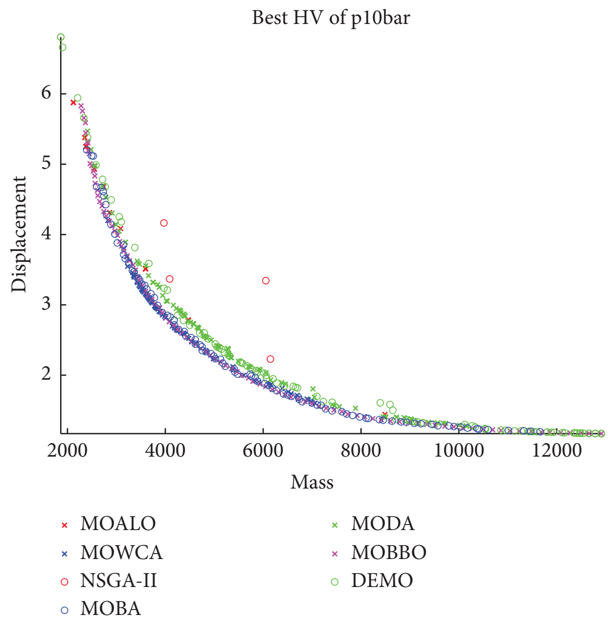


Figure 2: Best Pareto fronts of the 10-bar truss.

23.15%, 19.7%, 45.67%, 12.10%, and 17% superior to MOALO, MODA, MOWCA, NSGA-II, DEMO, and MOBA, respectively. MOBBO's dominance in the HV test for the 942-bar truss is confirmed by its first-place F-test rank, indicating a superior convergence rate. Moreover, MOBBO achieves an average Friedman rank of 1.68, with a first-place rank overall in the HV test, followed by MOBA and DEMO in second and third places, respectively. MOBA, DEMO, and MODA achieve second, third, and fourth Friedman ranks in the HV metrics.

4.2.2. GD and IGD Metrics Analysis for Truss Structures. GD metrics analysis (Table 3) indicates the difference between the true Pareto front optimal set and the solutions produced by the tested algorithms within the objective space. IGD metrics (Table 4) provide information regarding how close the solution sets produced by the algorithm are to the Pareto front solutions. For both analyses, smaller values are preferred.

For the 10-bar truss, MOBBO achieved the best values for standard deviations with 0.1259 for GD metrics and 2.6639 for IGD metrics, outperforming all other competitive algorithms. MOALO also performed effectively in GD metrics for the 10-bar truss structure, with a Friedman rank of 1.07, followed by MOBBO with a rank of 2.93. For IGD analysis, MOBBO achieved a Friedman rank of 1.27, indicating minimal difference between the solution sets produced by MOBBO and the Pareto optimal set. GD metric analysis for the 25-bar and 60-bar trusses, as shown in Table 3, suggests that MOALO performed better in terms of maximum values for objective functions, with values of 0.1641 and 0.2215, respectively, and Friedman ranks of 1 and 1.37 for the 25-bar and 60-bar trusses, respectively. However, MOBBO also demonstrated effective results for GD metrics analysis. For the 25-bar and 60-bar truss designs, MOBBO recorded effective maximum values for the fitness functions as 1.5462 and 0.4886, respectively, for IGD metrics analysis, showing closeness of the results to global optimum solutions. For IGD metrics analysis, MOBBO achieved Friedman ranks of 1.07 and 1 for the 25-bar and 60-bar truss structures, respectively. In contrast, MOBA, DEMO, and MODA achieved Friedman ranks of 2.27, 3.20, and 3.70, respectively, for the 25-bar truss and 2.03, 3.33, and 4.77, respectively, for the 60-bar truss.

For the 72-bar truss, GD metrics analysis showed that MOBBO achieved a Friedman rank of 5.37 with a maximum fitness function value of 3.8788, which is competitive compared to other algorithms. MOALO, MODA, and MOBA achieved effective standard deviations of 0.1828,

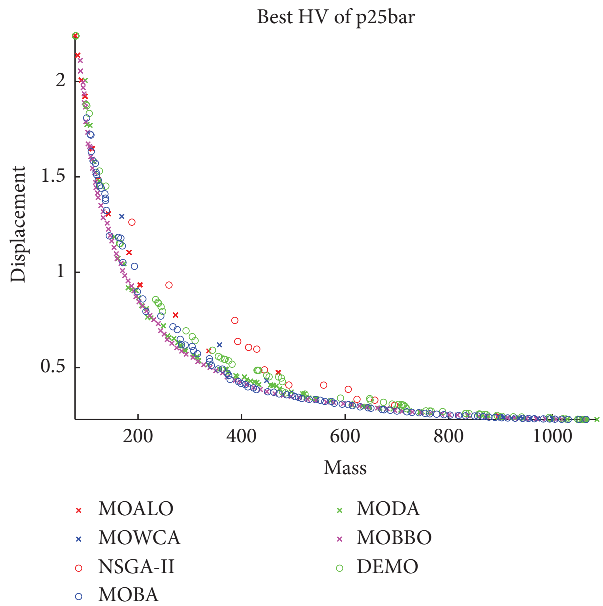


Figure 3: Best Pareto fronts of the 25-bar truss.

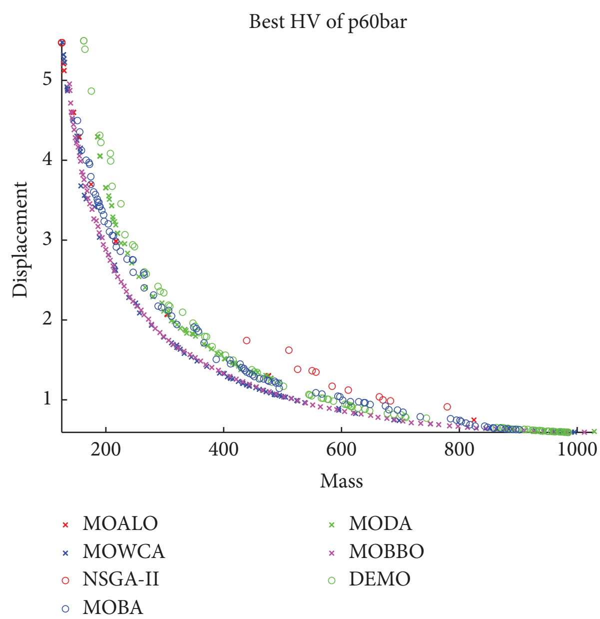


Figure 4: Best Pareto fronts of the 60-bar truss.

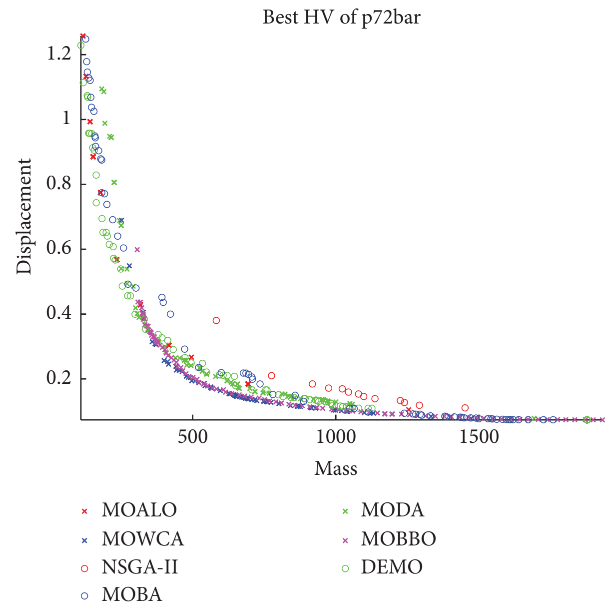


Figure 5: Best Pareto fronts of the 72-bar truss.

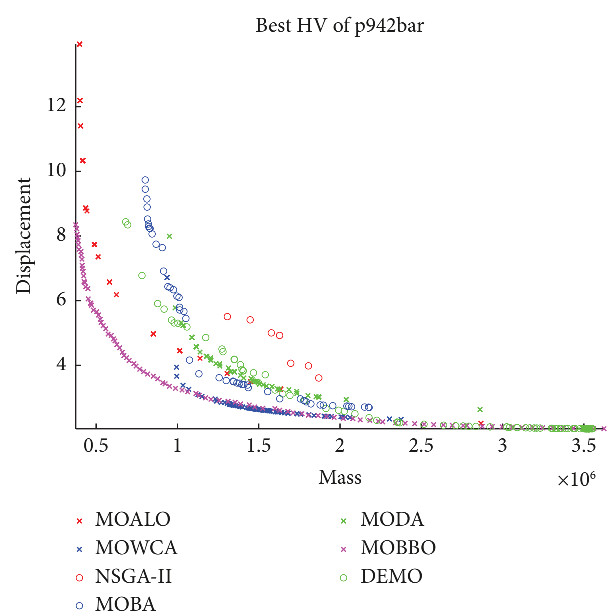


Figure 6: Best Pareto fronts of the 942-bar truss.

0.1771, and 0.2531, respectively, demonstrating effective results for the 72-bar truss. For IGD metrics analysis, MOBBO realized standard deviations that were 76.85%, 54.78%, and 81.43% less than those of MOALO, MODA, and MOWCA, respectively, indicating that the achieved solutions are closer to the Pareto front solutions. MOBBO showed effective results for the most challenging 942-bar truss structure compared to other studied truss structures. For GD and IGD metrics of the 942-bar truss, MOBBO achieved Friedman ranks of 2.0 and 1, respectively, demonstrating the algorithm's dominance over the compared MO versions. Additionally, MOBBO achieved an average of 50% and 70% less standard deviation in the results compared to other algorithms for GD and IGD metrics analysis, respectively. Overall, MOBBO achieved Friedman ranks of 2 and 1 for GD and IGD metrics analysis, respectively, highlighting the effectiveness of the proposed algorithm compared to existing MO versions.

4.2.3. STE Metric Analysis. STE metrics analysis provides information on the diversity of solutions relative to Pareto fronts, the convergence of results, and the clustering of solution sets within the search space. Higher values of STE metrics indicate better diversity among the solutions. Table 5 presents the statistics recorded in STE tests for MOBBO and

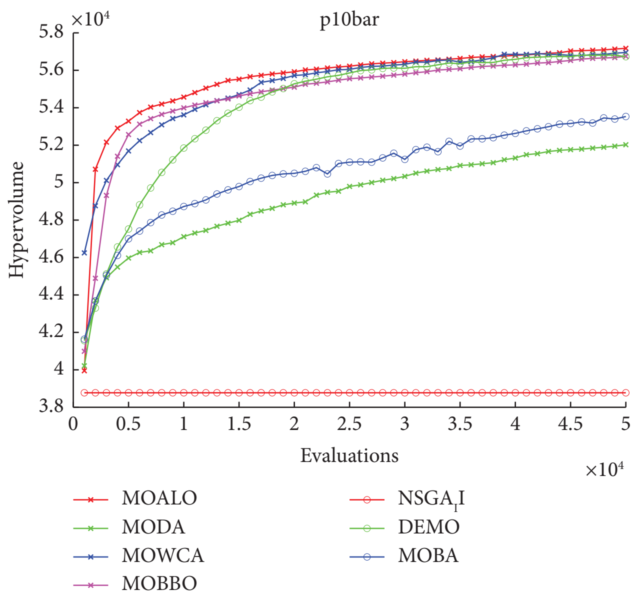


FIGURE 7: The hypervolume vs. function evaluations of the 10-bar truss.

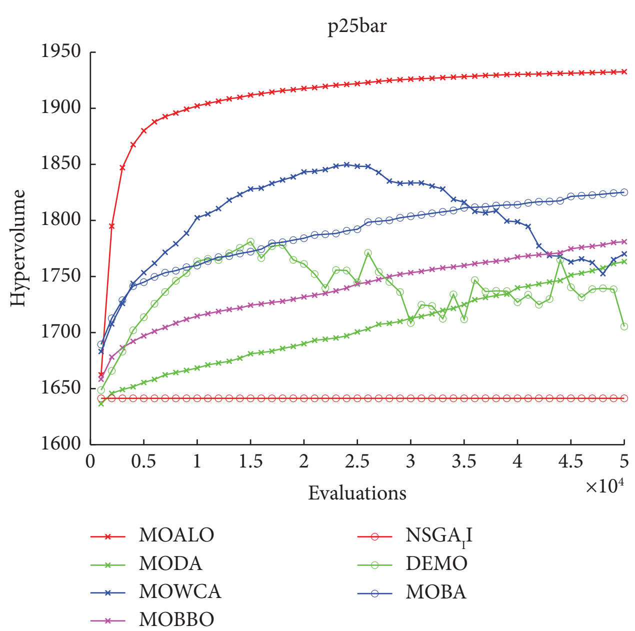


FIGURE 8: The hypervolume vs. function evaluations of the 25-bar truss.

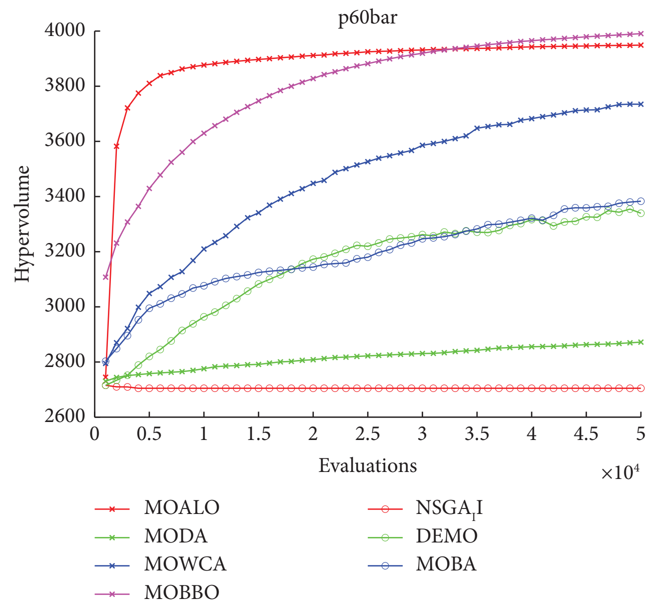


FIGURE 9: The hypervolume vs. function evaluations of the 60-bar truss.

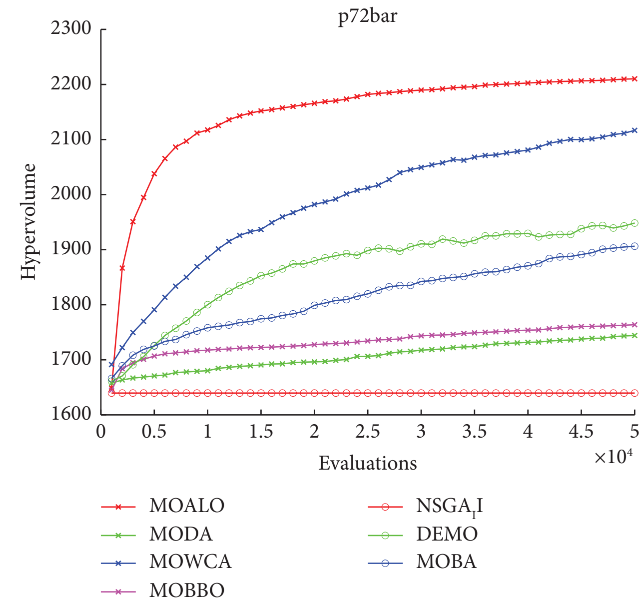


FIGURE 10: The hypervolume vs. function evaluations of the 72-bar truss.

other compared algorithms. For the 10-bar, 25-bar, and 60-bar trusses, MOBBO achieved Friedman ranks of 2.60, 2.80, and 1.80, respectively. Additionally, MOBBO showed minimal deviations in the results compared to all other algorithms, with values of 0.0001, 0.0002, and 0.0003 for the 10-bar, 25-bar, and 60-bar trusses, respectively. For the 72-bar truss, MOBBO demonstrated better average and maximum values for fitness functions at 0.0070 and 0.0074, respectively, with a standard deviation of 0.0002, which was lower than that of the other algorithms. However, DEMO, MOBA, and MOBBA had Friedman ranks of 1.57, 2.70, and 2.37, respectively, for the 72-bar truss. For the 942-bar truss, MOBBO showed dominance with the least standard deviation of 0.0003 and a Friedman rank of 1.63, outperforming MOALO and NSGA-II. In contrast, DEMO had a standard deviation of 0.0035 with a Friedman rank of 1.93.

Table 6 provides information on the overall Friedman ranks obtained for each truss structure by the compared algorithms, including MOBBO. MOBBO achieved the first position in the overall Friedman rank analysis, followed by MOBA and DEMO in second and third positions, respectively. For each individual truss case, MOBBO realized

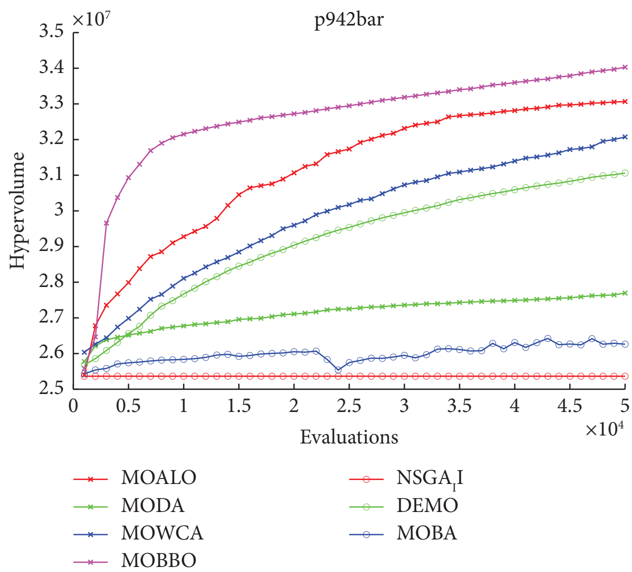


Figure 11: The hypervolume vs. function evaluations of the 942-bar truss.

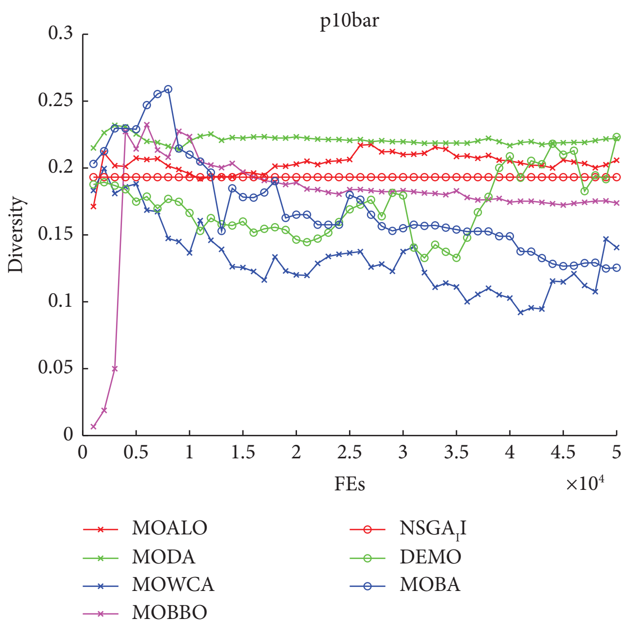


Figure 12: The diversity curve of 10-bar truss problem.

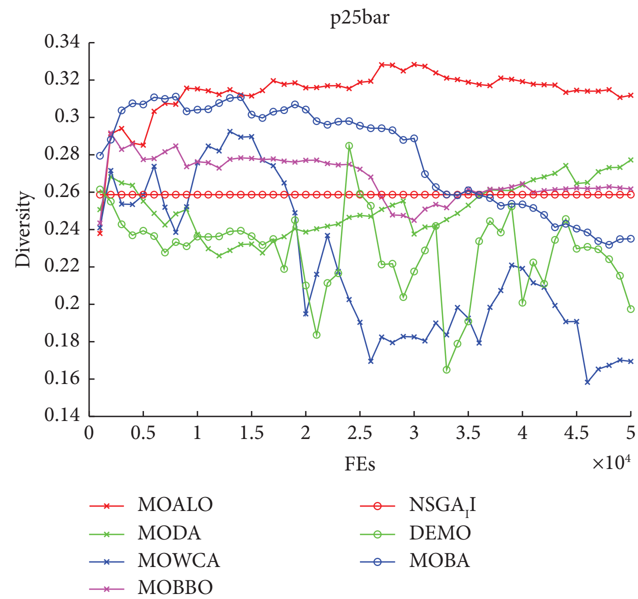


Figure 13: The diversity curve of 25-bar truss problem.

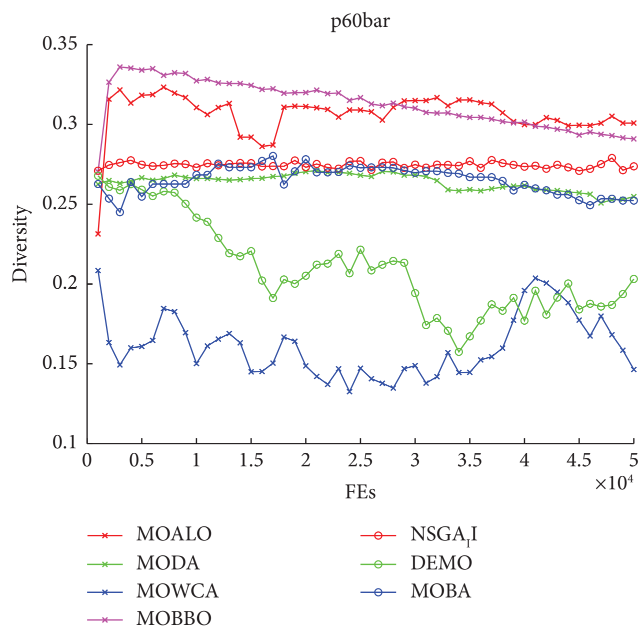


Figure 14: The diversity curve of 60-bar truss problem.

an average Friedman rank of 2.60, demonstrating the algorithm's superior capability in attaining global optima compared to the other algorithms.

4.2.4. Best Pareto Fronts, HV, Diversity Curves, and Boxplots Graphs Analysis for MOBBO. Figures 2, 3, 4, 5, 6 show the mass of truss structures versus nodal displacements curves for the best Pareto front solutions achieved by the algorithms. Figures 2 and 3 display the trend lines of the best Pareto fronts for the 10-bar and 25-bar structures, respectively, as achieved by all compared algorithms. The graphs reveal that MOBBO for the 10-bar and 25-bar structures is well-converged and smooth compared to the erratic nature of MOBA and NSGA-II. Specifically, MOBBO, MOBA, and MODA converge around 9000 mass values and exhibit a constant nature with minimal nodal displacement for the 10-bar truss. For the 25-bar truss, MOBBO maintains a constant trend after 800 mass values with 0.25 displacements.

Figures 4 and 5 present the best Pareto front graphs for the 60-bar and 72-bar trusses, respectively. MOBBO demonstrates evenly distributed and converged patterns in both cases, whereas other algorithms, especially NSGA-II and DEMO, show intermittent and irregular patterns. Figure 6 illustrates the 942-bar truss case, where MOBBO displays

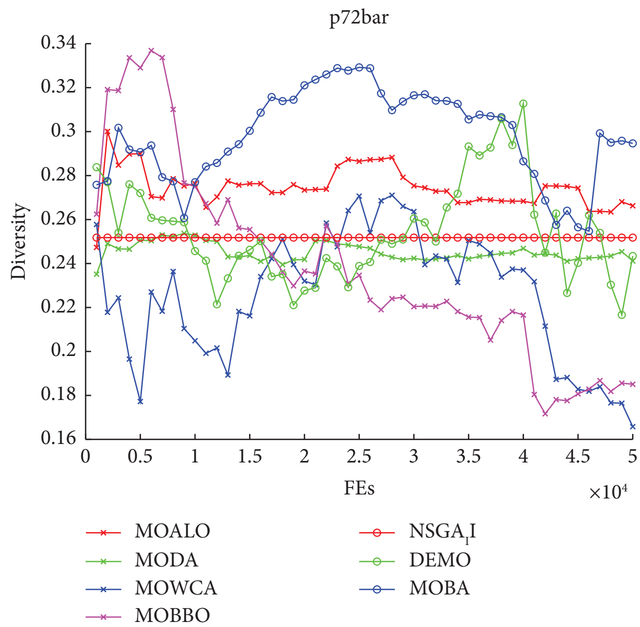


Figure 15: The diversity curve of 72-bar truss problem.

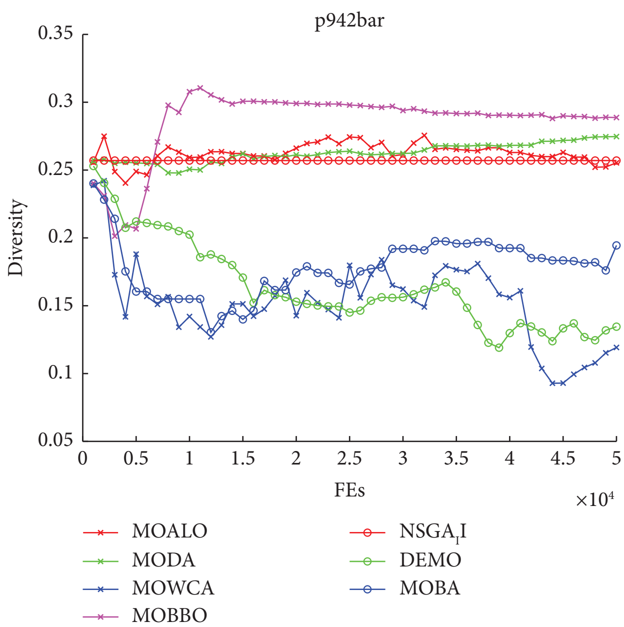


Figure 16: The diversity curve of the 942-bar truss problem.

a regular and smooth graph compared to other algorithms, with a notable break in the pattern after 1.0 mass value. In contrast, DEMO, NSGA-II, MODA, MOALO, and MOBA exhibit erratic and uneven patterns. Overall, the best Pareto front graphs for each truss structure case highlight the effectiveness and efficiency of the MOBBO algorithm.

Figures 7, 8, 9, 10, 11 show the graphs of each tested algorithm for HV versus FEs for different truss structures. According to Figures 7 and 8, MOBBO exhibits a smooth and constant curve after 50,000 and 1800 HVs, respectively. All algorithms show converged patterns for both the 10-bar and 25-bar cases. For the 60-bar and 72-bar truss structures,

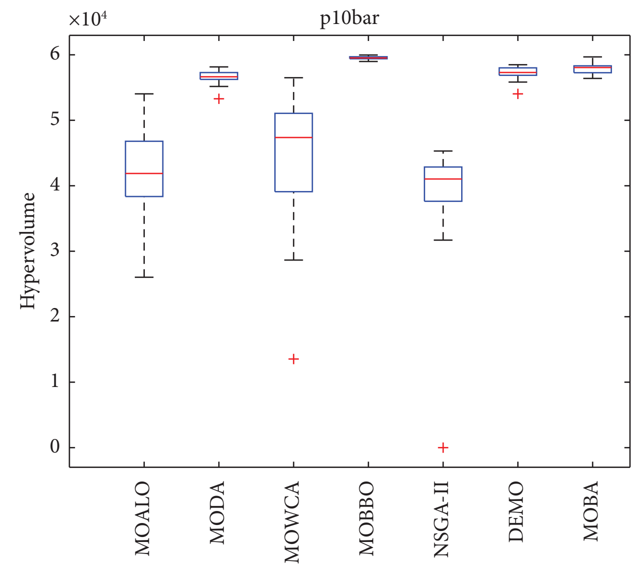


Figure 17: Boxplots of 10-bar truss.

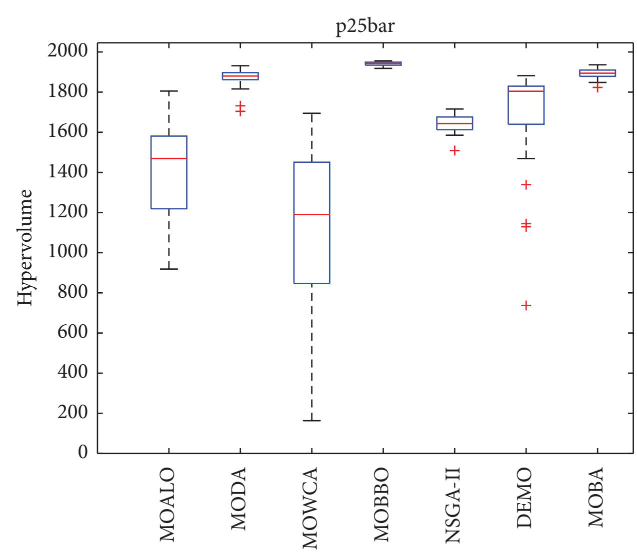


Figure 18: Boxplots of 25-bar truss.

as depicted in Figures 9 and 10, NSGA-II maintains a constant pattern with no noticeable variations. Meanwhile, the MOBBO graph rises up to 3500 HVs and then becomes constant around 30,000 evaluations for the 60-bar truss, and shows a nearly constant trend around 1000 HVs for the 72-bar truss case.

For the 942-bar truss case (Figure 11), MOBBO's graph increases up to 7×107 HVs and then shows a linear trend, while MOALO and MOBA exhibit irregular patterns. Additionally, NSGA-II displays straight constant lines, whereas MOWCA, DEMO, and MODA show increasing trend lines after certain HVs and evaluations for the 942-bar truss.

Figures 12, 13, 14, 15, 16 show the diversity graphs for all truss structures generated by the tested algorithms, including MOBBO. For the 10-bar and 25-bar structures, MOBBO initially shows slight diversity around 10,000 FEs, but it performs better compared to other algorithms

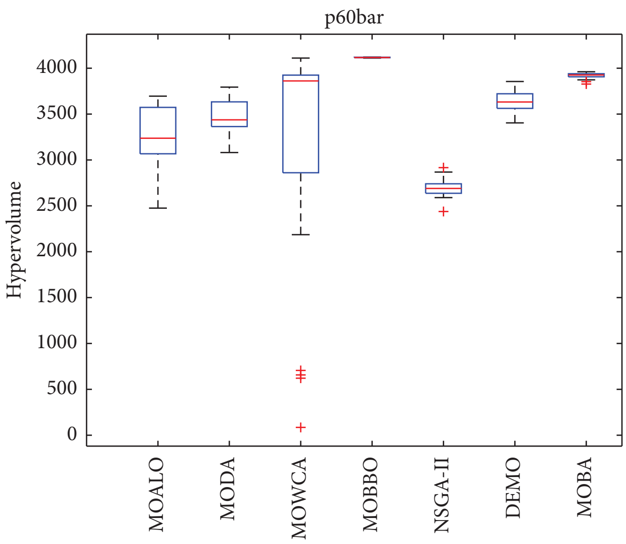


Figure 19: Boxplots of 60-bar truss.

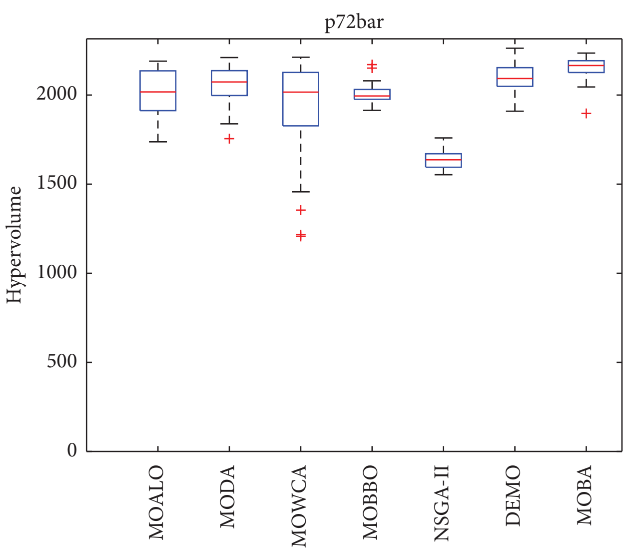


Figure 20: Boxplots of 72-bar truss.

thereafter. In contrast, MOWCA, DEMO, MODA, and MOBA exhibit significant diversity changes up to 50,000 iterations, with no clear convergence for the 10-bar and 25-bar cases.

According to Figure 14, for the 60-bar truss, the MOBBO graph is a smooth line after 500 iterations with diversity values around 0.3, while MOWCA and DEMO present erratic graphs with widespread variations across iterations. Figures 15 and 16 display the diversity curves for the 72-bar and 942-bar truss cases, respectively. In both cases, MOBBO performs well, with its trend lines showing promising results compared to other algorithms.

Figures 17, 18, 19, 20, 21 show boxplot analyses for all the tested algorithms. Smaller boxplots indicate better performance of the algorithm. The upper and lower lines within the boxplots represent the boundary values for the search space.

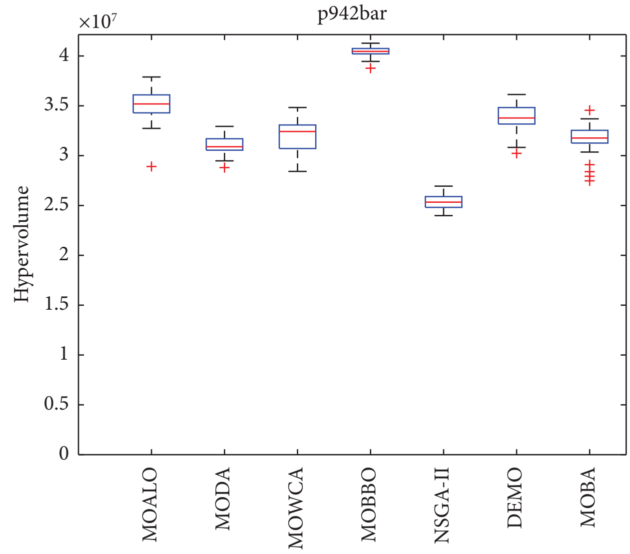


Figure 21: Boxplots of 942-bar truss.

Figures 17 and 18 depict the boxplots for the 10-bar and 25-bar trusses, respectively, with MOBBO showing thin boxplots. MODA, MOBA, and NSGA-II present competitive boxplots similar to MOBBO, while MOWCA shows the worst case.

For the 60-bar truss, Figure 19 reveals that MOBBO has a nearly flat boxplot with minimal deviations, demonstrating its dominance over the compared optimizers. In the 72-bar truss case, MOBBO, NSGA-II, DEMO, and MOBA show superior results compared to the wider spread of MOWCA and MOALO. For the 942-bar truss study, all optimizers show wide spread boxplots except MOBBO and MOALO.

5. Conclusion and Future Works

In conclusion, the comprehensive analysis demonstrates the superior performance of the MOBBO algorithm in tackling complex structural optimization problems. MOBBO's efficient convergence, solution diversity, and effectiveness in handling highly constrained problems make it a promising approach for MO optimization tasks across various engineering domains. The key points are as follows:

- Across all truss structures, MOBBO consistently achieved competitive or superior HV values compared to benchmark algorithms.
- For instance, in the case of the 10-bar truss structure, MOBBO attained a HV of 59,541.
- MOBBO also demonstrated better convergence with a standard deviation of 239 and a Friedman rank of 1.0, indicating minimal deviations and efficient convergence.
- MOBBO exhibited impressive performance in GD metrics, indicating its convergence to the Pareto front. MOBBO's competitive performance was further validated by its Friedman rank of 2.0 for GD metrics and 1.0 for IGD metrics, demonstrating its effectiveness in achieving solutions close to the Pareto front.

- Notably, MOBBO consistently exhibited lower standard deviations compared to other algorithms, indicating better solution spread and diversity.
- Visual representations, including Pareto front graphs, HV vs. FE curves, diversity curves, and boxplots, further support MOBBO's effectiveness. MOBBO consistently showed smooth convergence and stable performance across different truss structures, contrasting with erratic patterns observed in some benchmark algorithm. Boxplot analyses revealed MOBBO's narrow spread of results, indicating its robust performance and effectiveness in addressing optimization challenges.

While the MOBBO algorithm has demonstrated significant advantages in structural optimization, there are several avenues for future research and development:

- Scalability to Larger and More Complex Problems: Future studies could explore MOBBO's performance on even larger and more intricate structural optimization problems, including those with higher dimensionality or more complex constraints. Investigating how MOBBO scales with problem complexity would be valuable in understanding its broader applicability.
- Hybridization with Other Algorithms: Integrating MOBBO with other optimization techniques could potentially enhance its performance. Future research could explore hybrid models that combine the strengths of MOBBO with other algorithms like genetic algorithms or particle swarm optimization to tackle specific challenges or improve convergence speed.
- Parameter Sensitivity Analysis: A detailed analysis of the sensitivity of MOBBO to its control parameters (e.g., population size and mutation rate) could be conducted to optimize its performance further. Understanding how these parameters influence the algorithm's behavior could lead to more efficient tuning and application across various problem types.
- Application to Other Engineering Domains: While MOBBO has been tested on truss structures, its application in other engineering domains such as aerospace, automotive, and civil engineering could be explored. Investigating its effectiveness in different fields would help generalize the algorithm's utility.
- Real-World Case Studies: Applying MOBBO to real-world optimization problems with practical constraints and objectives would provide insights into its effectiveness in real engineering scenarios. This could also involve collaboration with industry to validate the algorithm's performance in practical applications.
- Algorithmic Improvements: Further refinement of the algorithm's structure, such as improving its convergence rate or enhancing its ability to escape local optima, could be explored. Additionally, introducing adaptive mechanisms that dynamically adjust parameters based on the optimization progress might improve overall performance.

Despite its promising results, the MOBBO algorithm has some limitations that should be acknowledged as computational cost, dependence on initial conditions and limited benchmarking.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

Author Contributions

Conceptualization, methodology, validation, and data curation, G.T., S.K., and P.M.; software, investigation, visualization, and supervision, G.T.; formal analysis, writing–original draft preparation, and writing–review and editing, G.T., S.K., P.M., and M.K. All authors have read and agreed to the published version of the manuscript.

Funding

The authors received no specific funding for this work.

References

[1] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, *Metaheuristic Algorithms: A Comprehensive Review. Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications* (2018).

[2] X. S. Yang, *Nature-inspired Metaheuristic Algorithms* (Luniver press, 2010).

[3] X. S. Yang, "Optimization and Metaheuristic Algorithms in Engineering," *Metaheuristics in water, geotechnical and transport engineering* 1 (2013): 23.

[4] M. F. Hamza, H. J. Yap, and I. A. Choudhury, "Recent Advances on the Use of Meta-Heuristic Optimization Algorithms to Optimize the Type-2 Fuzzy Logic Systems in Intelligent Control," *Neural Computing & Applications* 28, no. 5 (2017): 979–999, https://doi.org/10.1007/s00521-015-2111-9.

[5] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, and A. Baroutaji, "Review of Metaheuristic Optimization Algorithms for Power Systems Problems," *Sustainability* 15, no. 12 (2023): 9434, https://doi.org/10.3390/su15129434.

[6] A. R. Yıldız, H. Özkaya, M. Yıldız, S. Bureerat, B. S. Yıldız, and S. M. Sait, "The Equilibrium Optimization Algorithm and the Response Surface-Based Metamodel for Optimal Structural Design of Vehicle Components," *Materials Testing* 62, no. 5 (2020): 492–496, https://doi.org/10.3139/120.111509.

[7] V. Kesavan, R. Kamalakannan, R. Sudhakarapandian, and P. Sivakumar, "Heuristic and Meta-Heuristic Algorithms for Solving Medium and Large Scale Sized Cellular Manufacturing System NP-Hard Problems: A Comprehensive

Review," *Materials Today Proceedings* 21 (2020): 66–72, https://doi.org/10.1016/j.matpr.2019.05.363.

[8] S. Ghosh, A. Singh, and S. Kumar, "HPB3C-3PG Algorithm: A New Hybrid Global Optimization Algorithm and its Application to Plant Classification," *Ecological Informatics* 81 (2024): 102581, https://doi.org/10.1016/j.ecoinf.2024.102581.

[9] S. Imchen and D. K. Das, "Scheduling of Distributed Generators in an Isolated Microgrid Using Opposition Based Kho-Kho Optimization Technique," *Expert Systems with Applications* 229 (2023): 120452, https://doi.org/10.1016/j.eswa.2023.120452.

[10] Z. Wu, Y. Zhang, H. Bao, R. Lan, and Z. Hua, "nD-CS: A Circularly Shifting Chaotic Map Generation Method," *Chaos, Solitons & Fractals* 181 (2024): 114650, https://doi.org/10.1016/J.CHAOS.2024.114650.

[11] S. K. Joshi, "Levy Flight Incorporated Hybrid Learning Model for Gravitational Search Algorithm," *Knowledge-Based Systems* 265 (2023): 110374, https://doi.org/10.1016/j.knosys.2023.110374.

[12] S. Kumar, G. G. Tejani, N. Pholdee, S. Bureerat, and P. Mehta, "Hybrid Heat Transfer Search and Passing Vehicle Search Optimizer for Multi-Objective Structural Optimization," *Knowledge-Based Systems* 212 (2021): 106556, https://doi.org/10.1016/j.knosys.2020.106556.

[13] N. Gunantara, "A Review of Multi-Objective Optimization: Methods and its Applications," *Cogent Engineering* 5, no. 1 (2018): 1502242, https://doi.org/10.1080/23311916.2018.1502242.

[14] K. Deb, K. Sindhya, and J. Hakanen, "Multi-objective Optimization," in *Decision Sciences* (CRC Press, 2016).

[15] T. Prakash, P. P. Singh, V. P. Singh, and S. N. Singh, "A Novel Brown-Bear Optimization Algorithm for Solving Economic Dispatch Problem," in *Advanced Control & Optimization Paradigms for Energy System Operation and Management* (River Publishers, 2023).

[16] G. G. Tejani, N. Pholdee, S. Bureerat, D. Prayogo, and A. H. Gandomi, "Structural Optimization Using Multi-Objective Modified Adaptive Symbiotic Organisms Search," *Expert Systems with Applications* 125 (2019): 425–441, https://doi.org/10.1016/j.eswa.2019.01.068.

[17] F. Chen, S. Wu, F. Liu, J. Ji, and Q. Lin, "A Novel Angular-Guided Particle Swarm Optimizer for Many-Objective Optimization Problems," *Complexity* 2020 (2020): 1–18, https://doi.org/10.1155/2020/6238206.

[18] H. Ji and C. Dai, "A Simplified Hypervolume-Based Evolutionary Algorithm for Many-Objective Optimization," *Complexity* 2020 (2020): 1–7, https://doi.org/10.1155/2020/8353154.

[19] W. Zhong, J. Xiong, A. Lin, L. Xing, F. Chen, and Y. Chen, "Big Archive-Assisted Ensemble of Many-Objective Evolutionary Algorithms," *Complexity* 2021 (2021): 1–17, https://doi.org/10.1155/2021/6614283.

[20] S. Zapotecas-Martínez, A. García-Nájera, and A. Menchaca-Méndez, "Engineering Applications of Multi-Objective Evolutionary Algorithms: A Test Suite of Box-Constrained Real-World Problems," *Engineering Applications of Artificial Intelligence* 123 (2023): 106192, https://doi.org/10.1016/j.engappai.2023.106192.

[21] K. Kalita, J. S. Chohan, P. Jangir, and S. Chakraborty, "A New Decomposition-Based Multi-Objective Symbiotic Organism Search Algorithm for Solving Truss Optimization Problems," *Decision Analytics Journal* 10 (2024): 100371, https://doi.org/10.1016/j.dajour.2023.100371.

[22] J. P. G. Carvalho, D. E. Vargas, B. P. Jacob, B. S. Lima, P. H. Hallak, and A. C. Lemonge, "Multi-objective Structural Optimization for the Automatic Member Grouping of Truss Structures Using Evolutionary Algorithms," *Computers & Structures* 292 (2024): 107230, https://doi.org/10.1016/j.compstruc.2023.107230.

[23] N. Vo, H. Tang, and J. Lee, "A Multi-Objective Grey Wolf-Cuckoo Search Algorithm Applied to Spatial Truss Design Optimization," *Applied Soft Computing* 155 (2024): 111435, https://doi.org/10.1016/j.asoc.2024.111435.

[24] S. Farahmand-Tabar, "Multi-objective Lichtenberg Algorithm for the Optimum Design of Truss Structures," in *Applied Multi-Objective Optimization* (Singapore: Springer Nature Singapore, 2024).

[25] K. Kalita, N. Ganesh, R. C. Narayanan, P. Jangir, D. Oliva, and M. Pérez-Cisneros, "Multi-Objective Water Strider Algorithm for Complex Structural Optimization: A Comprehensive Performance Analysis," *IEEE Access* 12 (2024): 55157–55183, https://doi.org/10.1109/access.2024.3386560.

[26] C. Zhong, G. Li, Z. Meng, H. Li, and W. He, "Multi-objective SHADE with Manta Ray Foraging Optimizer for Structural Design Problems," *Applied Soft Computing* 134 (2023): 110016, https://doi.org/10.1016/j.asoc.2023.110016.

[27] K. Kalita, J. V. Naga Ramesh, R. Čep, S. B. Pandya, P. Jangir, and L. Abualigah, "Multi-objective Liver Cancer Algorithm: A Novel Algorithm for Solving Engineering Design Problems," *Heliyon* 10, no. 5 (2024): e26665, https://doi.org/10.1016/j.heliyon.2024.e26665.

[28] K. Kalita, G. Shanmugasundar, P. Jangir, J. S. Chohan, and L. Abualigah, "Prescriptive Analysis of NSGA-2 Variants for Performance Optimization in Constrained Truss Systems," *International Journal on Interactive Design and Manufacturing* 18, no. 7 (2024): 4595–4615, https://doi.org/10.1007/s12008-024-01737-x.

[29] K. Kalita, J. V. N. Ramesh, L. Cepova, S. B. Pandya, P. Jangir, and L. Abualigah, "Multi-objective Exponential Distribution Optimizer (MOEDO): a Novel Math-Inspired Multi-Objective Algorithm for Global Optimization and Real-World Engineering Design Problems," *Scientific Reports* 14, no. 1 (2024): 1816, https://doi.org/10.1038/s41598-024-52083-7.

[30] S. Duman, M. Akbel, and H. T. Kahraman, "Development of the Multi-Objective Adaptive Guided Differential Evolution and Optimization of the MO-ACOPF for wind/PV/tidal Energy Sources," *Applied Soft Computing* 112 (2021): 107814, https://doi.org/10.1016/j.asoc.2021.107814.

[31] B. Ozkaya, H. T. Kahraman, S. Duman, U. Guvenc, and M. Akbel, "Combined Heat and Power Economic Emission Dispatch Using Dynamic Switched Crowding Based Multi-Objective Symbiotic Organism Search Algorithm," *Applied Soft Computing* 151 (2024): 111106, https://doi.org/10.1016/j.asoc.2023.111106.

[32] H. T. Kahraman, M. Akbel, and S. Duman, "Optimization of Optimal Power Flow Problem Using Multi-Objective Manta Ray Foraging Optimizer," *Applied Soft Computing* 116 (2022): 108334, https://doi.org/10.1016/j.asoc.2021.108334.

[33] N. Mashru, G. G. Tejani, P. Patel, and M. Khishe, "Optimal Truss Design with MOHO: A Multi-Objective Optimization Perspective," *PLoS One* 19, no. 8 (2024): 03084744–e308537, https://doi.org/10.1371/journal.pone.0308474.

[34] S. Kumar, G. G. Tejani, P. Mehta, S. M. Sait, A. R. Yildiz, and S. Mirjalili, "Optimization of Truss Structures Using Multi-Objective Cheetah Optimizer," *Mechanics Based Design of Structures and Machines* (2024): 1–22, https://doi.org/10.1080/15397734.2024.2389109.

[35] I. Soesanti and R. Syahputra, "Multiobjective Ant Lion Optimization for Performance Improvement of Modern Distribution Network," *IEEE Access* 10 (2022): 12753–12773, https://doi.org/10.1109/ACCESS.2022.3147366.
[36] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation* 6, no. 2 (2002): 182–197, https://doi.org/10.1109/4235.996017.
[37] A. Sadollah, H. Eskandar, and J. H. Kim, "Water Cycle Algorithm for Solving Constrained Multi-Objective Optimization Problems," *Applied Soft Computing* 27 (2015): 279–298, https://doi.org/10.1016/j.asoc.2014.10.042.
[38] L. L. Laudis, S. Shyam, C. Jemila, and V. Suresh, "MOBA: Multi Objective Bat Algorithm for Combinatorial Optimization in VLSI," *Procedia Computer Science* 125 (2018): 840–846, https://doi.org/10.1016/j.procs.2017.12.107.
[39] M. Mlakar, D. Petelin, T. Tušar, and B. Filipič, "GP-DEMO: Differential Evolution for Multiobjective Optimization Based on Gaussian Process Models," *European Journal of Operational Research* 243, no. 2 (2015): 347–361, https://doi.org/10.1016/j.ejor.2014.04.011.
[40] S. Mirjalili, "Dragonfly Algorithm: a New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems," *Neural Computing & Applications* 27, no. 4 (2016): 1053–1073, https://doi.org/10.1007/s00521-015-1920-1.