



Reliability-based multi-objective optimization of trusses with greylag goose algorithm

Nikunj Mashru¹ · Ghanshyam G. Tejani^{2,3} · Pinank Patel¹

Received: 14 May 2024 / Revised: 18 October 2024 / Accepted: 25 December 2024 / Published online: 9 January 2025
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2025

Abstract

This paper introduces a multi-objective variant of the Greylag Goose Optimizer (MOGGO) to tackle complex structural optimization problems. Inspired by the cooperative behavior of geese in flight, MOGGO employs dynamic grouping to enhance problem-solving efficiency. Six truss structures undergo simultaneous topology, shape, and size optimization using MOGGO, aiming to maximize reliability while minimizing structural mass. By incorporating non-dominance sorting and archiving techniques, MOGGO extends the single-objective Greylag Goose Optimizer to effectively address trade-offs between competing objectives. Evaluation metrics and statistical tests demonstrate MOGGO's superior performance in handling large structural optimization problems, preserving more Pareto-optimal sets, and achieving greater convergence and variance in objective and decision spaces. MOGGO's ability to manage conflicting objectives is further validated through diversity analysis, with swarm plots illustrating its superior convergence behavior across iterations. Overall, MOGGO proves to be an efficient and effective approach for addressing challenging reliability-based multi-objective structural optimization problems.

Keywords Non-dominated sorting · Truss · Reliability · Constraint-based methodologies · Diversity analysis · Convergence behavior

1 Introduction

Engineering optimization aims to discover the optimal solution for a given engineering challenge, which has historically required extensive trial and error. This involved repetitive prototype development to assess various designs, such as altering a car's shape to minimize wind resistance. However, this process was costly, labor-intensive, and prone to human error. To address these challenges, automated optimization

algorithms have been developed to efficiently find optimal designs, reducing costs, decreasing human involvement, and fewer errors. Nonetheless, building effective optimization algorithms remains crucial for solving complex engineering problems. The search for the most efficient solutions to challenging issues has long been a primary focus of engineering optimization. Conventional techniques frequently required arduous trial and error procedures that took time, money, and human interaction. But now that metaheuristic algorithms are gaining popularity, there's a new way to approach these problems. Metaheuristic [1], robust and versatile algorithms are an adaptable and optimistic approach to various engineering applications, particularly those with non-linear or high-dimensional problems [2].

As optimization techniques, metaheuristic algorithms are well known and have been used extensively in research to find optimal solutions by minimizing or maximizing fitness functions. These algorithms are categorized as single-objective optimization algorithms since they only search for a single global solution when dealing with a single fitness criterion. However, multi-objective optimization adds complexity to the process by maximizing many competing objectives simultaneously. Despite this complexity,

✉ Ghanshyam G. Tejani
p.shyam23@gmail.com

Nikunj Mashru
nikunj.mashru039@gmail.com

Pinank Patel
pinankpatel19@gmail.com

¹ Department of Mechanical Engineering, Faculty of Engineering and Technology, Marwadi University, Rajkot, Gujarat, India

² Applied Science Research Center, Applied Science Private University, Amman 11937, Jordan

³ Jadara Research Center, Jadara University, Irbid 21110, Jordan

Table 1 Details of exploration phase (Ex1)

Steps	Mathematical representation of Ex ₁
1	$P(t+1) = P^*(t) - A \cdot C \cdot P^*(t) - P(t) $ $P(t+1)$ is the updated position of the search agent. A and C can be updated with the below equation with a change linearly [2,0]; $p(t)$ is an agent at an iteration t $A = 2ar_1 - a$, $C = 2r_2$; where, r_1 and r_2 are random values [0,1]
2	$P(t+1) = w_1 * P_{paddle1} + z * w_2 * (P_{paddle2} - P_{paddle3}) + (1 - z) * w_3 * (P - P_{paddle1})$, for $ A \geq 1$ Where, $P_{paddle1}$, $P_{paddle2}$, $P_{paddle3}$ are three random search agents that will not affect the leader position to achieve greater exploration. w_1 , w_2 , $w_3 = [0,2]$, and z decreases exponentially by step3
3	$z = 1 - (\frac{t}{t_{max}})^2$, t is the iteration number, and t_{max} is the maximum number of iterations
4	$P(t+1) = w_4 * P^*(t) - P(t) \cdot e^{bl} \cdot \cos(2\pi l) + [2w_1(r_4 + r_5)] * P^*(t)$ $P(t+1)$ is a secondary updating process, during a and A decreases $r_3 \geq 0.5$. b is a constant, $l = [-1,1]$, $w_4 = [0,2]$, r_4 and $r_5 = [0,1]$

multi-objective optimization issues in structural optimization can be addressed using a variety of approaches. Recently proposed effective and efficient metaheuristics mainly classified into swarm intelligence, human behavior-based, physics low inspired and evolution-based such as Stochastic paint-based optimizer [3], Ebola optimization search algorithm [4], Botox optimization algorithm [5], Greylag Goose Optimization [6], Squid game based optimization [7], Grasshopper optimization algorithm [8], The cheetah optimizer [9], Fick's law algorithm [10], Geometric mean optimizer [11], Puma optimizer [12], Electric eel foraging optimization [13], Hippopotamus optimization algorithm [14], Synergistic swarm optimization algorithm [15], Zebra Optimization algorithm [16], Quadratic interpolation optimization [17], The mountain gazelle optimization [18], Mantis search algorithm [19], Crayfish optimization algorithm [20], Kepler Optimization algorithm [21], Altruistic population algorithm [22]. While metaheuristic algorithms

show promise for optimization, recent critiques emphasize the need for rigorous methodology and transparency in their design.

The "No Free Lunch" [23] Theorem underscores the understanding that no single metaheuristic can universally solve all real-world problems. This recognition has driven the continuous development and refinement of various metaheuristic methods. In recent times, the literature has documented numerous Multi-Objective optimization (MO) approaches, such as MO water cycle optimization [24], MO grasshopper [25], Differential evolution for MO optimization [26], MO dragonfly algorithm [27], MO bat algorithm [28], MO artificial vultures optimization [29], MO Lichtenberg algorithm [30], MO passing vehicle search [31], MO heat transfer search [32], MO atomic orbital search [33], MO material generation algorithm [34], MO crystal structure algorithm [35], Fast and elitist MO genetic algorithm (NSGA-II) [36] and MO Hippopotamus algorithm[37], are

Table 2 Details of the exploitation phase (Ex2)

Steps	Mathematical representation of Ex ₂
1	$P_1 = P_{Sentry1} - A_1 \cdot C_1 P_{Sentry1} - P $ $P_2 = P_{Sentry2} - A_2 \cdot C_2 P_{Sentry2} - P $ $P_3 = P_{Sentry3} - A_3 \cdot C_3 P_{Sentry3} - P $ P_1 , P_2 , and P_3 are position updates by three solutions ($P_{Sentry1}$, $P_{Sentry2}$, $P_{Sentry3}$), which guide other individuals ($P_{nonSentry}$). A_1 , A_2 , A_3 are calculated as $A = 2ar_1 - a$ and C_1 , C_2 and C_3 are calculated as $C = 2r_2$
2	$P(t+1) = \bar{P}_i^3$ The updated positions for the population $P(t+1)$ are an average of P_1 , P_2 and P_3
3	$P(t+1) = P(t) + D(1+z) * w * (P - P_{Flock1})$ While flying, $P(t+1)$ updates its position by searching the area around the best solution (leader). P_{Flock1} is an enhancement that explores the region closest to the ideal response

proposed by many researchers to solve numerous MO and many-objective optimization problems [38]. Also, improving the MO algorithms by various strategies further enhances the effectiveness, such as opposition-based sine cosine algorithm for global optimization [39], decomposition-based MO heat transfer search for structural optimization [40], MO multi-verso optimization algorithm with two-archive strategy [41] [42] Adaptive symbiotic organisms search for MO truss structures [43], improved MODE based on a decomposition strategy [44].

Truss optimization, usually divided into topology, shape, and size optimization (TSS), is an essential component of structural optimization. Recent research supports simultaneous optimization of these variables—also referred to as simultaneous TSS [45]. Design—instead of the more conventional sequential approach. Optimization is done on objective functions like weight, compliance, and displacement; however, permissible stress, displacement, buckling load, and occasionally frequency limits are considered constraints. Truss reliability optimization (TRO) [46] is a more thorough technique since it considers uncertainties in applied loads and material qualities. The optimization problem involves two objectives: minimizing structural mass and maximizing structural reliability, with constraints based on the probability of failure.

This study presents a MO adaptation of the Greylag Goose Optimization (GGO) [6]. The algorithm MOGGO is inspired by the social and dynamic behaviours observed in geese. Geese's lifelong bonds and collective behaviours, including protective grouping and V formations during migration, are key influences in the algorithm's design. As the iteration process moves forward in search of the optimal solution, the algorithm dynamically controls two potent stages: exploration and exploitation. In multi-objective optimization, the main technique for determining a Pareto front combines population-based metaheuristics with non-dominated sorting. This iterative method continuously enhances the Pareto archive by combining data from the current population and the archive from the previous iteration. The algorithm's two phases optimize parameter settings dynamically, thereby improving search intensity and diversity. Additionally, effective clustering techniques are employed to preserve search diversity and enable compelling exploration of the large design space, which is particularly difficult in MO metaheuristics.

1. The single-objective swarm-based Greylag Goose algorithm combines an elitist non-dominant sorting method and archive mechanism to develop the novel MOGGO algorithm. This integration maintains Pareto's optimum

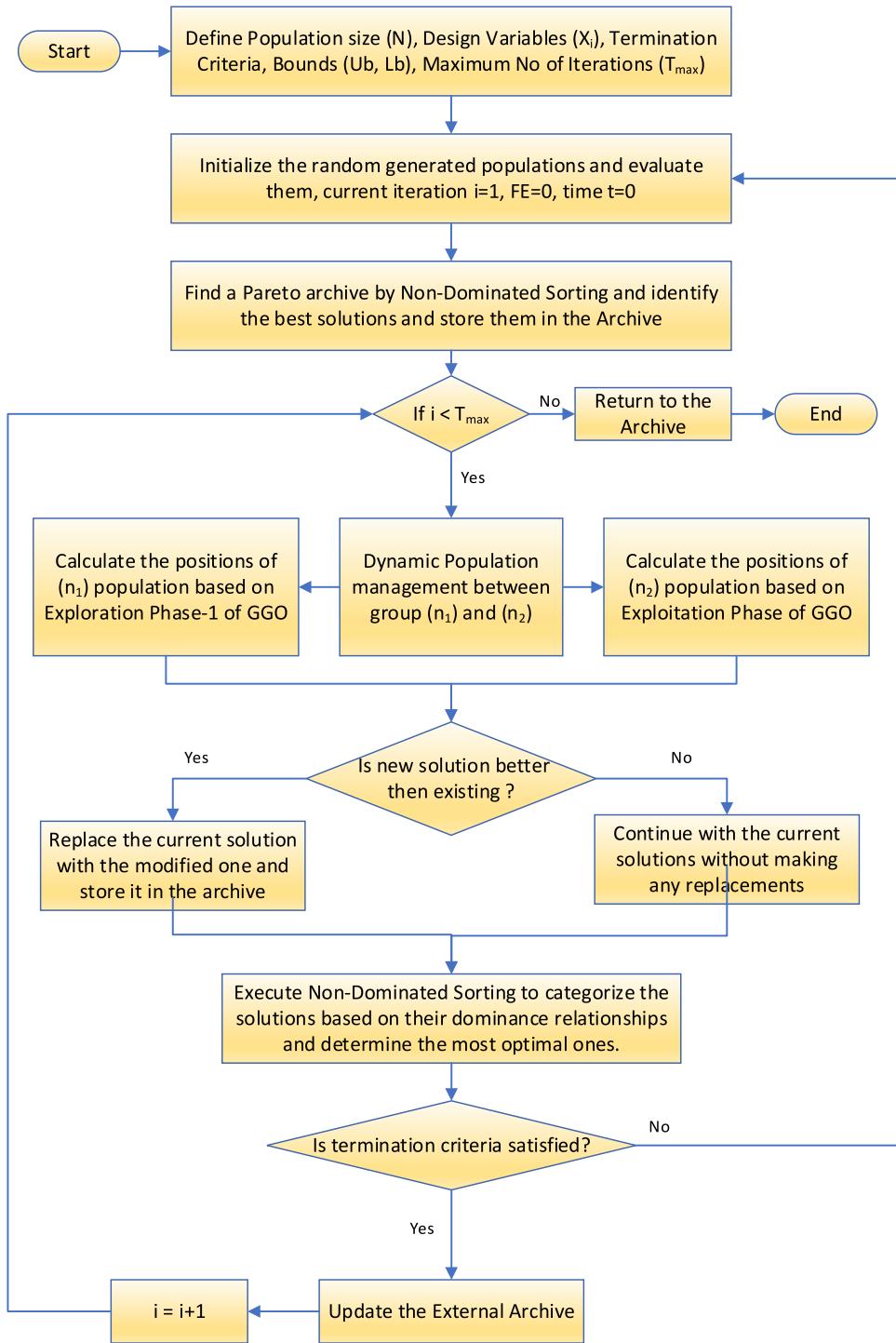
dominance while improving the convergence and diversity of solutions.

2. The performance of MOGGO for six truss structures is compared with seven state-of-the-art MO optimization algorithms, viz. MO bat algorithm (MOBA) [28], MO water cycle algorithm (MOWCA) [24], MO dragonfly algorithm (MODA) [27], MO differential evolution (DEMO) [26], non-dominated sorting genetic algorithm-II (NSGA-II) [36], MO grasshopper optimization algorithm (MOGOA) [25], MO ant lion optimizer (MOALO) [47].
3. Performance comparisons between the proposed MOGGO and other state-of-the-art algorithms are conducted using various performance metrics across all selected structural optimization problems.
4. The qualitative characteristics of each algorithm's best Pareto-front plots are analyzed. Additionally, a comprehensive study ranks the algorithms using a statistical test at a prescribed significance level.
5. Swarm plots and diversity curves illustrate the convergence and divergence of the proposed MO algorithms, providing visualizations of efficient optimization processes.

A newly developed MOGGO algorithm is utilized for simultaneous TSS optimization of six truss structures: 45-bar, 15-bar, 25-bar, 39-bar, 68-bar, and 224-bar. The objective is to minimize structural mass while maximizing reliability, with the constraint of ensuring that the probability of failure does not exceed 5%. The critical advancements of this research and its evolution, which exceed current contemporary standards, are outlined as follows:

The structure of the manuscript is organized as follows.

- Section 2 represents the mathematical model of the innovative GGO algorithm with a dual-phase approach.
- Section 3 elaborates on the proposed MOGGO and outlines the formulations of the MO structure reliability-based optimization problems.
- Section 4 formulates the truss design problems, non-dominating strategy, archiving techniques, and an evaluation matrix overview.
- Section 5 of the manuscript provides the experimental assessment of the MOGGO optimizer and compares its performance with other well-known algorithms with well-known performance matrices and statistical tests.
- The study concludes by offering final observations and insights in Sect. 6.

Fig. 1 Flowchart of MOGGO

2 Mathematical representations of GGO

The social and dynamic behaviours of geese serve as inspiration for the proposed GGO algorithm, which highlights the creatures' collective dynamics, loyalty, and protective instincts. GGO starts by creating populations at random, each of whom stands for a possible solution. These individuals comprise a population, and their fitness is assessed using

an objective function. The algorithm dynamically divides individuals into exploration and exploitation groups, which modifies the number of groups according to the optimal solution. GGO strives for effective optimization by avoiding local optima and combining aspects of both exploration and exploitation. The GGO's approach starts by randomly creating individuals to symbolize possible solutions. The individuals who comprise the GGO population represented

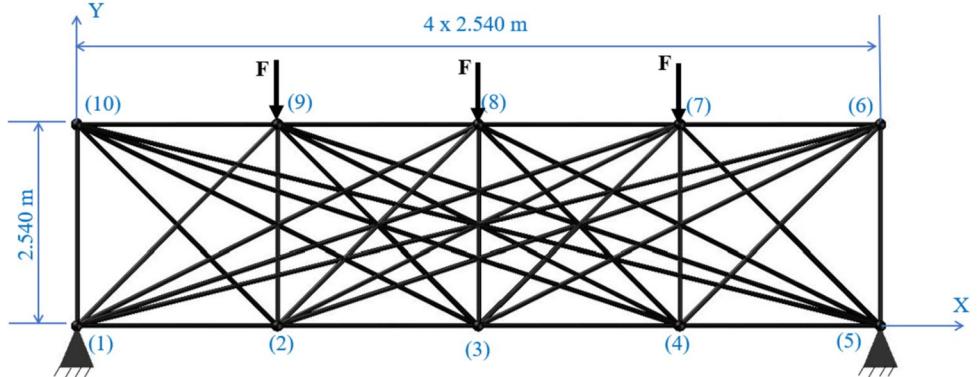
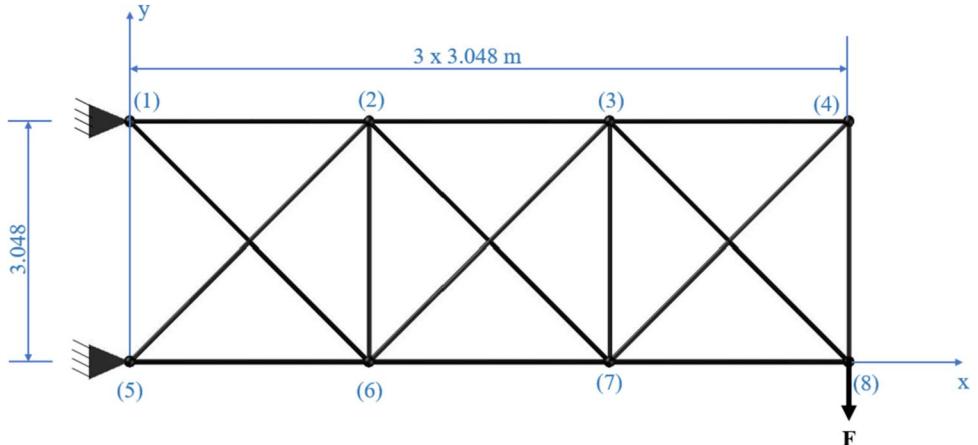
Table 3 Design considerations for the truss Problems

Truss structures	Variable details	Design parameters details
45-bar	Loading conditions Material considerations	Nodes 7,8 and 9 ($F_y = -44.482$) $E = 6.895 \times 10^{10}$ $\rho = 2767.990$ $\sigma_y = 1.724 \times 10^8$
15-bar	Topology, size (45) Loading conditions Material considerations	$A_i \in [5.806, 64.52] \times 10^{-5} \text{ m}^2, i = 1, 2, 3, \dots, 45$ Nodes 8 ($F_y = -44.482$) $E = 6.895 \times 10^{10}$ $\rho = 2767.990$ $\sigma_y = 1.724 \times 10^8$
25-bar	Shape (8) Topology, size (15) Loading conditions	$x_2 = x_6; x_3 = x_7; y_2; y_3; y_4; y_6; y_7; y_8$ $2.540 \text{ m} \leq x_2, y_2, y_3 \leq 3.556 \text{ m}; 5.588 \text{ m} \leq x_3 \leq 6.604 \text{ m}; 1.270 \text{ m} \leq y_4 \leq 2.268 \text{ m};$ $0.508 \text{ m} \leq y_8 \leq 1.524 \text{ m}; -0.508 \text{ m} \leq y_6, y_7 \leq 0.508 \text{ m}$ $S = \{7.161, 9.097, 11.23, 14.19, 17.42, 18.52, 22.39, 28.39, 34.77, 61.55, 67.74, 75.74, 86.00, 96.00,$ $113.8, 138.2, 174.0, 180.6, 202.0, 230.0, 246.0, 310.0, 384.0, 424.0, 464.0, 550.0, 600.0, 700.0,$ $860.0, 921.9, 1108, 1237\} \times 10^{-5} \text{ m}^2$ Nodes 1 ($F_x = 4.448, F_y = -44.482, F_z = -44.482$) Nodes 2 ($F_y = -44.482, F_z = -44.482$) Nodes 3 ($F_x = 2.224$) Nodes 6 ($F_x = 2.669$)
39-bar	Material considerations	$E = 6.895 \times 10^{10}$ $\rho = 2767.990$ $\sigma_y = 2.758 \times 10^8$
68-bar	Shape (5) Topology, size (8) Loading conditions	$x_4 = x_5 = -x_3 = -x_6; x_8 = x_9 = -x_7 = -x_{10}; y_3 = y_4 = -y_5 = -y_6; y_7 = y_8 = -y_9 = -y_{10}; z_3 = z_4 = z_5 = z_6$ $0.508 \text{ m} \leq x_4 \leq 1.524 \text{ m}; 1.016 \text{ m} \leq x_8, y_4 \leq 2.032 \text{ m}; 2.540 \text{ m} \leq y_8 \leq 3.556 \text{ m}; 2.286 \text{ m} \leq z_4 \leq 3.302 \text{ m}$ $A_i \in S, i = 1, 2, 3, \dots, 8$ $S = \{6.452, 12.90, 19.35, 25.81, 32.26, 38.71, 45.16, 51.61, 58.06, 64.52, 70.97, 77.42, 83.87, 90.32,$ $96.77, 103.2, 109.7, 116.1, 122.6, 129.0, 135.5, 141.9, 148.4, 154.8, 161.3, 167.7, 180.6, 193.5,$ $206.5, 219.4\} \times 10^{-5} \text{ m}^2$ Nodes 2, 3 and 4 ($F_y = -88.964$) $E = 6.895 \times 10^{10}$ $\rho = 2767.990$ $\sigma_y = 1.379 \times 10^8$
68-bar	Material considerations	$\Delta y_{11}, \Delta x_6 = -\Delta x_9; \Delta y_6 = \Delta y_9; \Delta x_7 = -\Delta x_8; \Delta y_7 = \Delta y_8; \Delta x_{10} = -\Delta x_{12}; \Delta y_{10} = \Delta y_{12}$. Horizontal and vertical coordinates may vary within $\pm 3.048 \text{ m}$ of their initial values $A_i \in [3.226, 145.2] \times 10^{-5} \text{ m}^2, i = 1, 2, 3, \dots, 21$
68-bar	Shape (7) Topology, size (21) Loading conditions	$A_i \in [3.226, 145.2] \times 10^{-5} \text{ m}^2, i = 1, 2, 3, \dots, 21$ Case I: node 17 ($F_x = -222.411$) Case II: node 17 ($F_x = -222.411, F_y = -66.723$)
68-bar	Material considerations	$E = 2.068 \times 10^{11}$ $\rho = 8303.971$ $\sigma_y = 1.379 \times 10^8$
68-bar	Shape (31)	$y_{17}, x_i, y_i, i = 2, 4, 5, 6, \dots, 14, 15, 16, 18$
68-bar	Topology, size (68)	$A_i \in S, i = 1, 2, 3, \dots, 68$ $S = \{7.161, 9.097, 11.23, 14.19, 17.42, 18.52, 22.39, 28.39, 34.77, 61.55, 69.74, 75.74, 86.00,$ $96.00, 113.8, 138.2, 174.0, 180.6, 202.0, 230.0, 246.0, 310.0, 384.0, 424.0, 464.0, 550.0, 600.0,$ $700.0\} \times 10^{-5} \text{ m}^2$

Table 3 (continued)

Truss structures	Variable details	Design parameters details
224-bar	Loading conditions	Node 1 ($F_x = 500$, $F_y = 500$, $F_z = -1000$)
	Material considerations	$E = 1.999 \times 10^{11}$ $\rho = 7850$ $\sigma_y = 2.482 \times 10^8$
Shape (18)		$x_2, x_3, y_3, y_4, x_{18}, y_{19}, y_{20}, x_{34}, x_{35}, y_{35}, y_{36}, x_{50}, x_{51}, y_{51}, z_2, z_{18}, z_{34}$ x_2, x_3, y_3 and y_4 may vary within ± 1.25 m; x_{18}, y_{19} , and y_{20} may vary within ± 2.5 m; x_{34}, x_{35}, y_{35} and y_{36} may vary within ± 3.75 m; x_{50}, x_{51} , and y_{51} may vary within ± 5.0 m; z_2, z_{18} and z_{34} may vary within ± 2.5 m of their initial value
Topology, size (32)		$\{A_{1-2}, A_{1-3}, A_{1-4}, A_{2-3}, A_{3-4}, A_{2-18}, A_{2-19}, A_{3-18}, A_{3-19}, A_{3-20}, A_{4-19}, A_{4-20}, A_{18-19}, A_{19-20}, A_{19-34}, A_{18-35}, A_{19-34}, A_{19-35}, A_{19-36}, A_{20-35}, A_{20-36}, A_{34-35}, A_{35-36}, A_{34-50}, A_{34-15}, A_{35-50}, A_{35-51}, A_{35-52}, A_{36-51}, A_{36-52}, A_{50-51}, A_{50-52}\} \in S$ $S = \{6.452, 12.90, 19.35, 25.81, 32.26, 38.71, 45.16, 51.61, 58.06, 64.52, 70.97, 77.42, 83.87, 90.32, 96.77, 103.2, 109.7, 116.1, 122.6, 129.0, 135.5, 141.9, 148.4, 154.8, 161.3, 167.7, 174.2, 180.6, 187.1, 193.5, 200.0, 206.5, 212.9, 219.4, 225.8, 232.3, 238.7, 245.2, 251.6, 258.1, 264.5, 271.0, 277.4, 283.9, 290.3, 296.8, 303.2, 309.7, 316.1, 322.6\} \times 10^{-4} \text{ m}^2$

Where Load F is in kN, Young's Modulus E is in N/m², Density ρ is in kg/m³, Yield strength σ_y is in N/m²

Fig.2 The truss structure with 45-bar**Fig.3** The truss structure with 15-bar

by the symbol P_i (where $i = 1, 2, \dots, n$) are assessed using an objective function F_n to determine the optimal solution, P . Individuals are divided into exploration (Ex1) and exploitation (Ex2) groups by the GGO algorithm, which employs

dynamic groups. These groups are dynamically altered based on the optimal solution found. Both groups initially participated in 50% exploration and exploitation to improve solution quality and avoid local optima, but the ratio changes

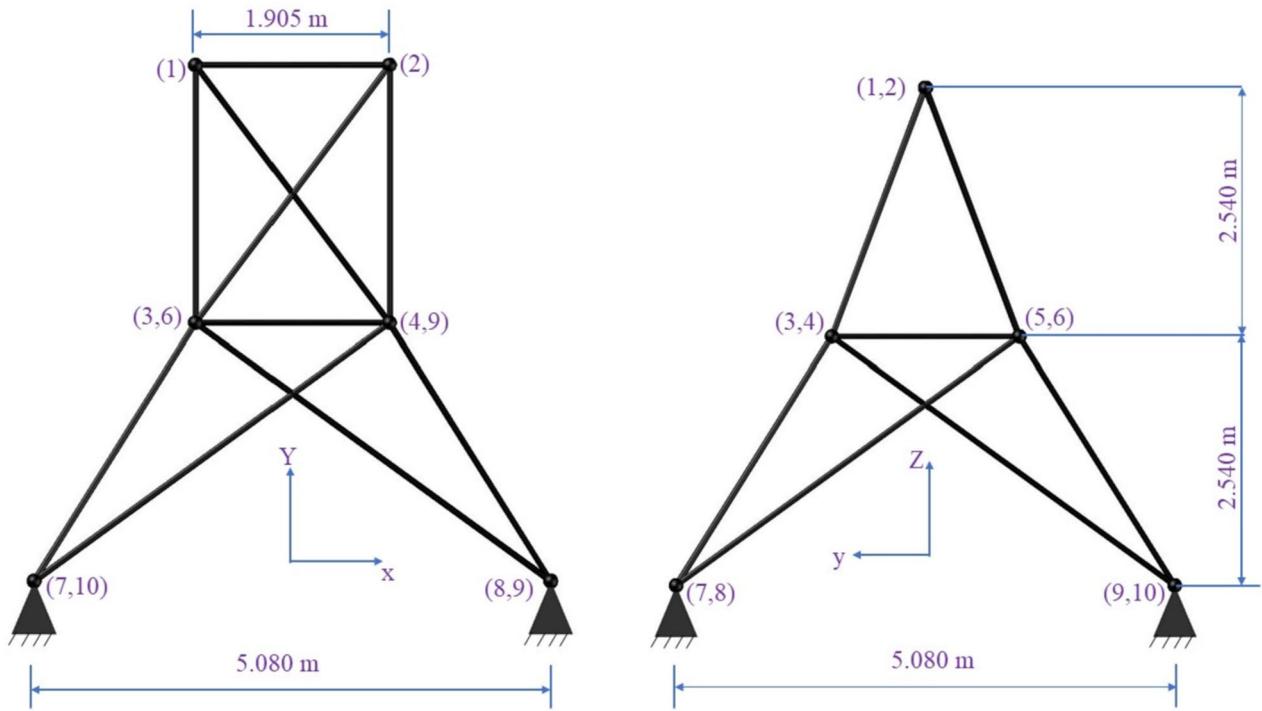
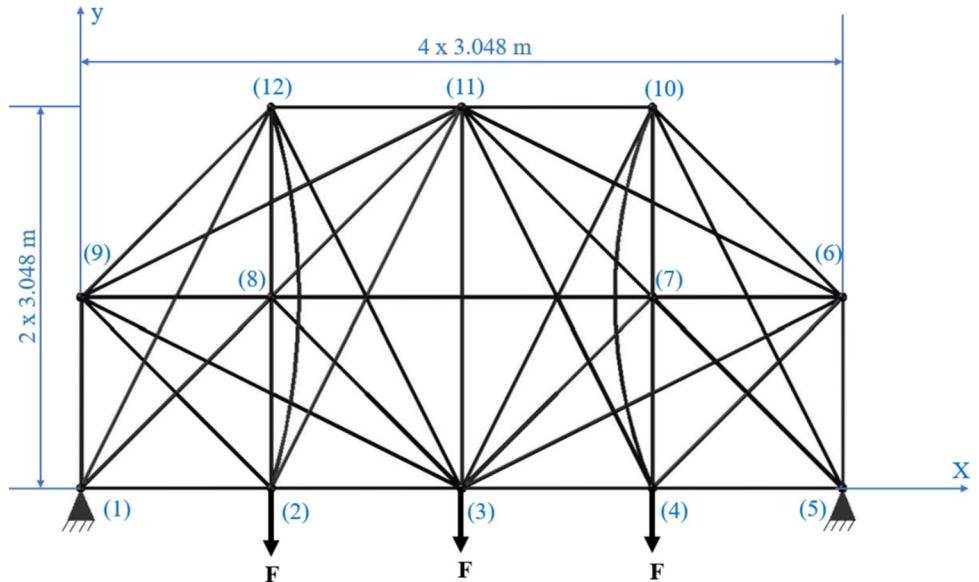


Fig. 4 The truss structure with 25-bar

Fig. 5 The truss structure with 39-bar



with each iteration. If the best solution does not improve for three iterations, the exploration group size increases to avoid local optima. Subsequent rounds involve changing group compositions.

2.1 Exploration phase (Ex_1)

To find novel regions, the algorithm broadens its search throughout the solution space during the exploration phase

(Ex_1). This phase entails investigating various potential solutions to guarantee thorough search space coverage. This approach seeks to find promising regions and prevents early convergence to inferior solutions by prioritizing exploration. The algorithm broadens its search throughout the solution space during the exploration phase to find novel regions. This phase entails investigating various potential solutions to guarantee thorough search space coverage. This approach seeks to find promising regions and prioritizes exploration

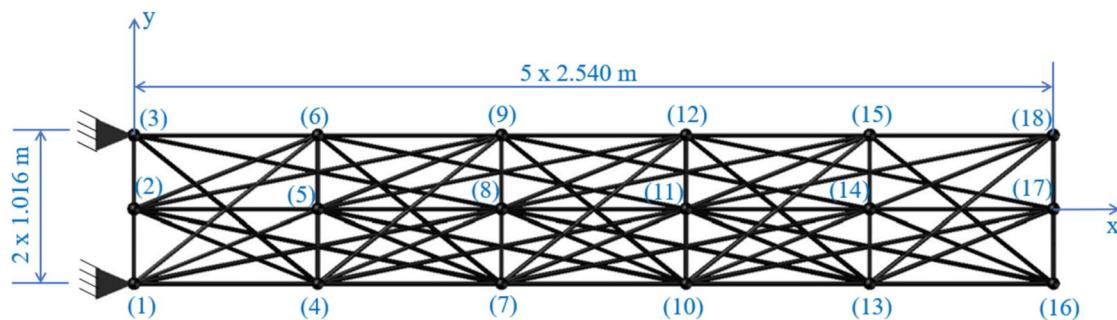


Fig.6 The truss structure with 68-bar

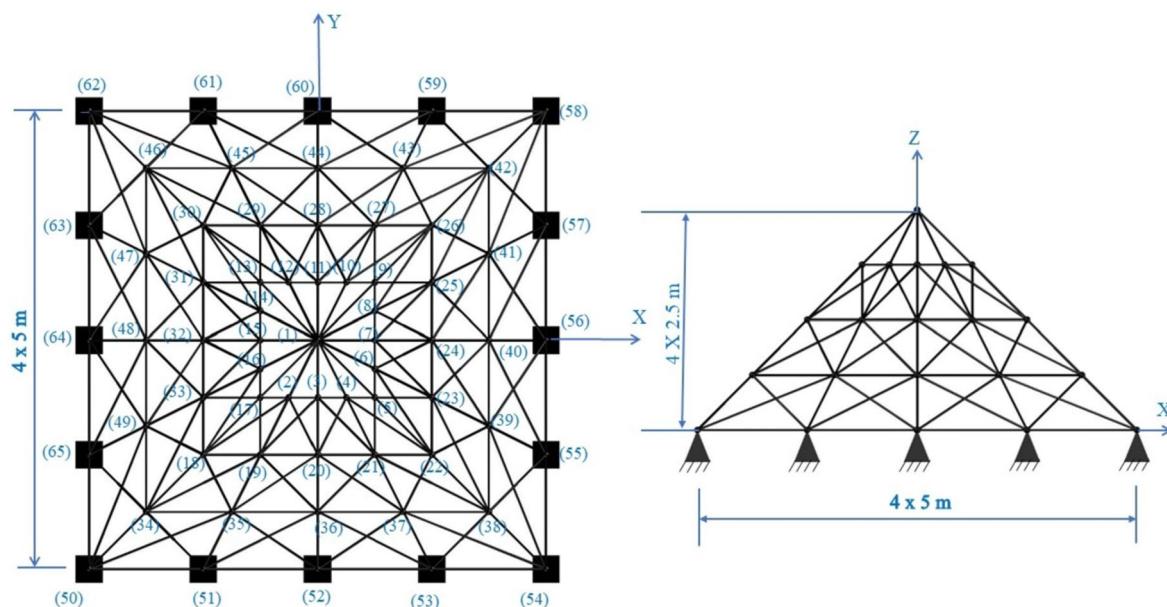


Fig.7 The truss structure with 224-bar

to prevent early convergence to inferior solutions. The geese explorer methodically looks for potential areas close to its current position to find an optimal solution. This means iteratively assessing local possibilities to determine the most favourable fitness.

The 4-step process can find the best solution by Ex_1 Phase as per the below Table 1.

2.2 Exploitation phase (Ex_2)

The algorithm concentrates on stepping up its search inside promising areas of the solution space during the exploitation phase. It accomplishes this by utilizing the insights from earlier searches to refine and optimize solutions. The

algorithm aims to refine solutions and converge towards the best answer by focusing on taking advantage of recognized attractive areas. As per Table 2, dual strategies viz, moving towards the best solution and searching around the best solution with a three-step process will guide the GGO algorithms toward the best optimal solution.

3 The proposed reliability-based MO optimization

Truss optimization confronts material properties and applied loads uncertainties, necessitating reliability-based multi-objective optimization (MO)[48]. Anti-optimization and

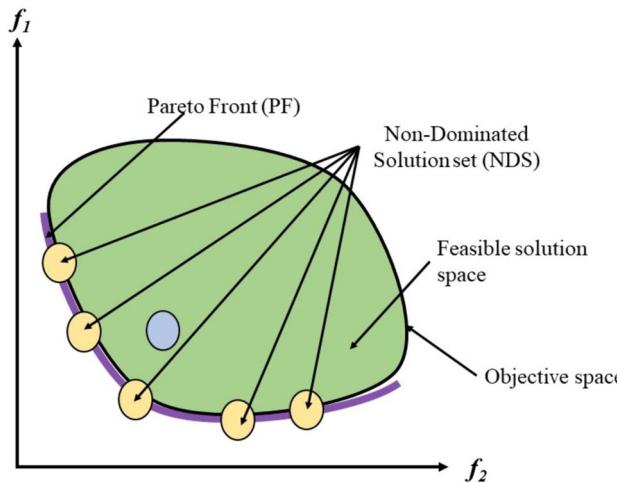


Fig. 8 Pareto dominance and NDS

possibility-based design optimization deal with worst-case scenarios and uncertain probability distributions. The approach of Greiner and Hajela[49] Park et al. use Taylor's series expansion to approximate the most probable point while combining structural mass and reliability optimization. Their method provides valuable information for effectively optimizing multi-objective truss systems. [50] also contributed to optimizing truss dependability using MOEAs by introducing efficient techniques for failure probability calculation[51]. Comparing the outcomes of different optimizers using a hypervolume indicator offers insightful information for subsequent optimization endeavours. Deterministic challenges with truss optimization seek to minimize objectives with two sorts of variables: constant settings such as material parameters and design variables[52]. Uncertain variables are introduced by reliability optimization, such as defects in material yield strength, necessitating adopting a reliability index as

an objective function[53]. Thus, the MO optimization problem is formulated as follows in this study:

3.1 Formulation of the truss design problem and FEA perspective

The comprehensive approach to MO truss design problems uses two objective functions subjected to the probability of failure rather than as per Eq. 1.

$$\begin{aligned} \min_x f_1(x, y) &= \sum_{(i=1)}^{N \text{ bar}} \rho_i L_i A_i \\ \max_x f_2(x, y) &= \beta \end{aligned} \quad (1)$$

subject to

probability of failure ≤ 0.05 ,

where, $f_1(x)$ is mass of truss structures and $f_2(x)$ is reliability index[49] of the truss structures, which can be calculated per Eq. 3. x is a vector of design variables representing shape, size, and topology simultaneously. At the same time, y is a vector containing random variables or uncertainties, including yield strength and applied loads. Mass is influenced by the density of the material used in a structure (ρ), the length of elements of the truss structure (L), and the cross-sectional area of the element (A). The vector y represents random variables associated with the material yield strength and the magnitudes of applied loads. Both objectives are subjected to the probability of failure, which should be less than 5%, as per Eq. 2.

$$\left[1 - 0.5 * \left\{ 1 + \operatorname{erf} \left(\beta / \sqrt{2} \right) \right\} \right] \leq 0.05. \quad (2)$$

Table 4 Details of performance quality indicators

Sr. no	Performance matrix	Descriptions
1	$HV = \operatorname{volume} \left(\bigcup_{i=1}^A V_i \right)$	It offers insights into the quality of the solution set S . Each solution i in S is associated with a hypercube V_i formed by a set of reference points
2	$GD = \frac{\sqrt{\sum_{i=1}^M d_i^2}}{ P }$	$ P $ represents the number of outcomes in the Pareto front, where d_i denotes the Euclidean distance to the nearest solution from the reference front and the objective function vector of the i th solution in the obtained front. It measures the volume of the objective space that is dominated by the solutions in the Pareto front
3	$IGD = \frac{\sqrt{\sum_{i=1}^M (d_i^2)}}{ P }$	$ P $ indicates the number of solutions on the reference plane. This metric is used to evaluate both front expansions and advancements
4	$SP = \frac{1}{ P - 1} \sum_{i=1}^{ P } (d_i - \bar{d})^2$ $ET = \sum_{i=1}^M f_i^{\max} - f_i^{\min} $	The Euclidean distance d_i measures the separation between the objective function vector of the i^{th} solution and its nearest neighbor. \bar{d} represents the mean value of all d_i , where M is the number of objective functions. f_i^{\max} and f_i^{\min} represent the maximum and minimum values of the i^{th} objective function of the front, respectively

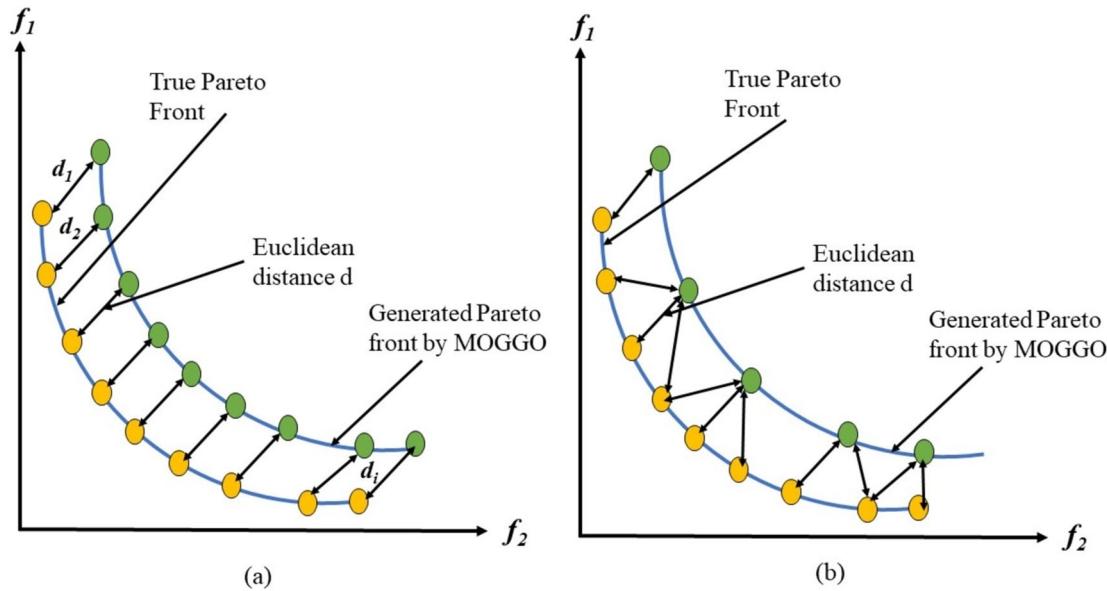


Fig. 9 **a** GD and **b** IGD for MO optimization MOGGO algorithm with reference point A

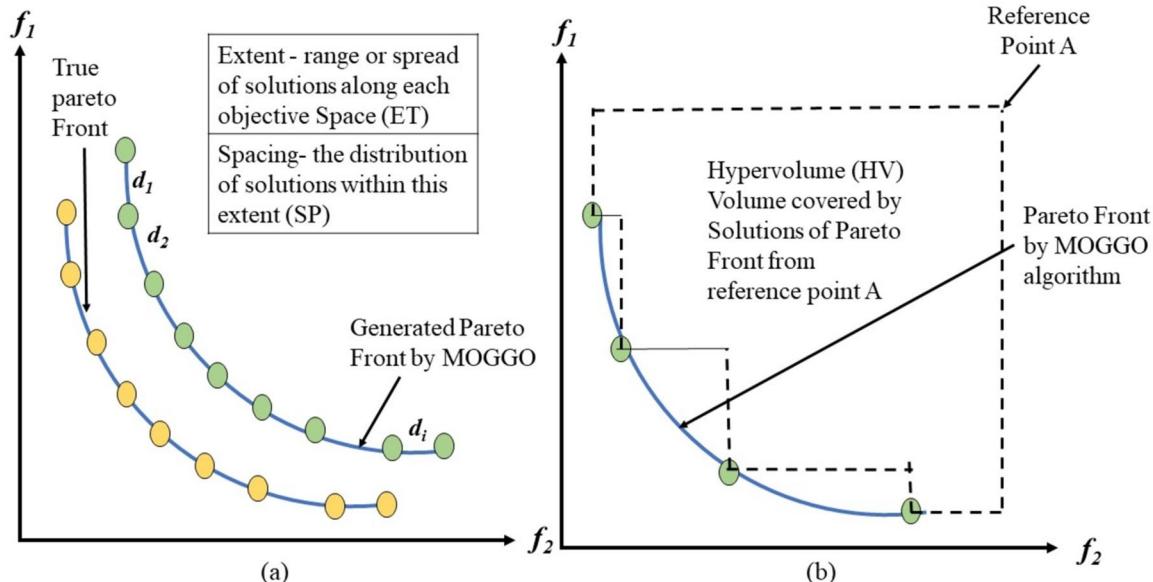


Fig. 10 Concept of **a** spacing, extent, and **b** hypervolume

$$\beta = \frac{\mu_M}{\sigma_M} = \frac{\mu_{Sy} - \sum_{i=1}^n k_i \mu F_i}{\sqrt{\sigma_{Sy}^2 + \sum_{i=1}^n (k_i \sigma F_i)^2}} \quad (3)$$

where, \$\mu_M\$ is a limit state function of the mean value of resistance and load magnitude, while \$\sigma_M\$ represents the standard deviation of resistance and standard deviation of load magnitude, \$k_i\$ is the ratio between stress occurring on

each member to the \$i\$th external load, which is a contribution of \$F_i\$ to stress on the element, in which the index \$I\$ represents \$i^{th}\$ specific component on which the loads is applied. \$\sigma_{sy}\$ and \$\sigma_{Fi}\$ are the variances of \$Sy\$ and \$Fi\$, respectively, as per Eq. 3. A higher value of the reliability index (\$\beta\$) indicates more safety, and therefore, it is set as an objective function that can be maximized during the optimization process. Solving

Eq. 4 in a linear finite element system can achieve truss static analysis.

$$[K_{ij}]\{u_i\} = \{F_{ex,i}\} \quad (4)$$

where, $[K_{ij}]$ is $N \times N$ stiffness matrix, $\{u_i\}$ is a $N \times 1$ The nodal displacement vector is the degree of freedom, and N is the degree of freedom. The equation above already addresses the boundary conditions. Stresses in the elements Se can be calculated using Eq. 5 as per the relation.

$$S_e = [T]\{u_i\} = [T][K_{ij}]^{-1}\{F_{ex,i}\} \quad (5)$$

S_e is a $N_e \times 1$ vector containing truss element stresses, e is an element index N_e is the number of elements and $[T]$ is a $N_e \times N$ stress transformation matrix.

3.2 The proposed multi-objective greylag goose optimizer

Figure 1 depicts the flowchart of the proposed algorithm. The flowchart visually represents the algorithm's key steps, including initialization, population update, constraint handling, selection, and archive update. Each step enhances the algorithm's ability to efficiently explore the search space and identify high-quality solutions. This algorithm incorporates an archive and non-dominated sorting. Retaining non-dominated solutions with an archive during optimization reduces the possibility of discarding more favorable solutions. Consequently, this algorithm exhibits enhanced exploration or diversity.

4 Formulation of the truss design problem and its MO compliance

4.1 Truss structures

All Contemplated truss structural problems [45] To evaluate MOGGO's performance, shape, size, and topology are considered simultaneously for multi-objective reliability-based optimization. Table 3 shows the design considerations of all the truss structures with loading conditions, material properties, shape, size, and topological information.

- The first structural problem is a 45-bar truss composed of 45 bars, as shown in Fig. 2. Vertical downward loads of 44.483 KN act on nodes 7, 8, and 9. These loads exert force vertically downwards on the specified nodes, imposing stress and strain on the truss elements. The truss design process encompasses both size and topology optimization, with a total of 45 design variables. These

Table 5 Parameter setting of all considered algorithms

Parameters	MOBA	NSGA-II	MOWCA	MOGOA	MODA	MOALO	DEMO	MOGGO
Population size	100	100	100	100	100	100	100	100
Maximum number of generations	250	250	250	250	250	250	250	250
Archive size	100	100	100	100	100	100	100	100
Other related parameters	Loudness: 0.5 Pulse rate: 0.5 Frequency range: [0, 2]	Crossover probability: -0.9	a (control parameter: -linearly decreased from 2 to 0)	C _{max} (separation weight): 0.1a (alignment weight): 0.1c (cohesion weight): -1	Beta: -4 Crossover probability: -0.9	Initial exploration & exploitation group size: 50% of the population each		
		Mutation probability: -0.5		C _{min} (mini- mum com- fort zone): -0.00004	Gamma: -2 (F): -0.5	Differential weight (enemy distraction weight): 0.1e		

Table 6 The hypervolume (HV)

		MOBA	NSGA-II	MOWCA	MOGOA	MODA	MOALO	DEMO	MOGGO
45-bar truss	average	3912.54	4681.49	4352.71	3810.26	2977.96	3834.35	2596.14	4854.58
	max	4223.59	4826.20	4787.83	4210.19	3206.29	4105.86	2799.32	4957.15
	min	3484.47	4484.00	3532.75	3306.04	2730.12	3497.45	2470.53	4752.10
	std	168.72	86.34	293.70	200.26	125.91	138.40	85.94	54.43
	Fried- man rank	4.43	2.07	3.30	5.10	7.03	5.10	7.97	1.00
15-bar truss	average	32,257.11	27,306.42	33,965.34	36,853.52	32,704.89	35,553.14	32,657.83	38,230.27
	max	34,222.40	28,911.27	36,256.40	37,205.38	34,951.98	36,444.80	36,297.95	38,305.98
	min	29,498.83	25,830.45	30,737.58	35,952.06	30,750.78	34,044.35	30,138.29	38,139.34
	std	1091.68	861.62	1272.10	284.58	1018.65	663.37	1232.88	44.54
	Fried- man rank	6.13	8.00	4.40	2.03	5.67	3.20	5.57	1.00
25-bar truss	average	9891.09	10,368.47	10,089.26	10,291.13	10,011.72	10,243.72	9007.85	10,890.08
	max	10,482.52	10,647.83	10,524.13	10,510.59	10,438.42	10,687.08	10,018.11	10,904.40
	min	9135.99	10,045.77	8590.62	9994.66	9282.83	9619.02	8286.83	10,873.79
	std	385.26	185.30	503.43	142.78	290.59	294.18	489.84	6.73
	Fried- man rank	5.80	3.27	4.60	4.00	5.47	4.07	7.80	1.00
39-bar truss	average	10,257.03	8700.90	11,854.14	12,572.04	11,042.87	11,496.27	10,085.86	13,532.48
	max	11,759.98	9861.18	12,672.23	12,892.09	11,800.90	12,421.07	10,716.73	13,682.05
	min	5638.30	7723.96	9364.09	12,217.08	9964.18	10,524.65	9378.14	13,328.04
	std	1416.21	558.34	633.18	189.87	395.64	488.16	286.33	99.94
	Fried- man rank	5.97	7.87	3.23	2.17	5.00	4.07	6.70	1.00
68-bar truss	average	188,828.09	134,487.33	238,345.97	243,285.49	201,007.82	218,424.38	202,399.01	270,778.85
	max	209,438.84	149,095.36	257,375.18	251,545.21	222,562.13	241,069.21	216,717.63	278,134.42
	min	141,723.47	116,180.26	211,299.37	233,053.92	180,112.48	195,783.72	181,342.22	259,222.01
	std	12,343.99	8428.69	9571.70	4287.07	10,326.93	10,686.94	8876.67	4758.02
	Fried- man rank	6.67	8.00	2.80	2.33	5.57	4.20	5.43	1.00
224-bar truss	average	3,469,511.69	3,655,196.36	4,330,771.98	4,664,454.78	3,492,158.53	4,419,604.29	3,724,839.21	4,973,946.47
	max	3,997,440.53	3,820,510.86	4,564,951.83	4,799,246.97	3,788,997.31	4,703,435.26	4,407,979.76	5,026,388.72
	min	2,648,427.41	3,513,521.64	3,939,465.25	4,493,106.81	3,154,789.27	3,936,235.29	3,379,308.83	4,909,795.34
	std	284,037.66	85,631.74	149,173.26	76,191.50	140,949.16	175,567.85	202,018.20	27,299.00
	Fried- man rank	7.00	6.13	3.63	2.10	7.10	3.30	5.73	1.00
Average Fried- man	6.00	5.89	3.66	2.96	5.97	3.99	6.53	1.00	
Overall Fried- man rank	7	5	3	2	6	4	8	1	

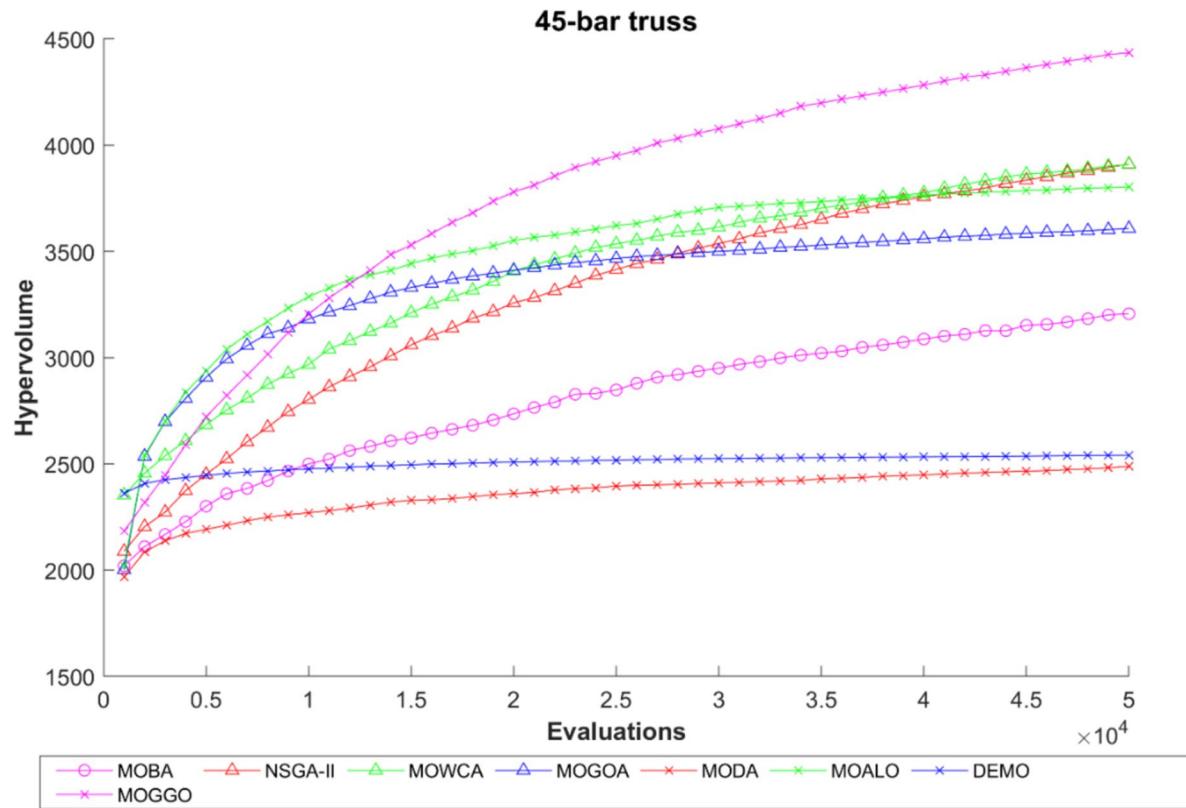


Fig. 11 Comparative hypervolume evolution for 45-bar truss

variables dictate the dimensions and configurations of the truss elements, determining their lengths, cross-sectional areas, and connection patterns. By exploring various combinations of size and topology configurations, the optimization process seeks to identify the most efficient and reliable truss design for the given loading conditions and structural requirements.

- The second truss in question is a 15-bar truss, depicted in Fig. 3, comprising 15 ground elements. A load of 44.483 kN is applied to node 8, exerting a downward force. This load direction influences the truss structure's stress distribution and deformation characteristics. The optimization process involves considering shape and topological/size aspects, with 8 shape-type and 15 topological/size-type design variables, as specified in Table 3.
- The third problem entails a 25-bar 3-D truss, illustrated in Fig. 4, which exhibits symmetry in the x-z and y-z planes. This symmetrical arrangement simplifies design optimization by reducing the required unique design variables. In this case, 13 design variables are utilized for topology, shape, and size (TSS) optimization. These variables encompass various aspects of the truss structure's geometry, including its elements' lengths, angles, and cross-sectional areas. By employing a multi-objective optimization approach, the objective is to identify the

optimal combination of design variables that minimize the structural mass while maximizing reliability and performance under applied loads.

- The fourth truss structure is a 39-bar configuration, depicted in Fig. 5, featuring 28 design variables. These variables govern various aspects of the truss's topology, shape, and size, allowing for comprehensive optimization. The material properties and loading conditions pertinent to this structure are outlined in Table 3, providing essential parameters for the optimization process. Considering these inputs, the optimization algorithm can systematically explore the design space to identify configurations that minimize mass while ensuring structural reliability under specified loading conditions. The many design variables offer flexibility in tailoring the truss geometry to meet performance objectives, such as maximizing reliability or minimizing structural mass.
- The fifth truss structure, illustrated in Fig. 6, comprises a 68-bar configuration in a 2-D layout. This truss exhibits a more complex geometry than previous structures, offering increased design flexibility and optimization challenges. Two distinct load cases are specified for this truss, as detailed in Table 3, providing essential input parameters for the optimization process. Multiple load cases necessitate a comprehensive analysis to ensure

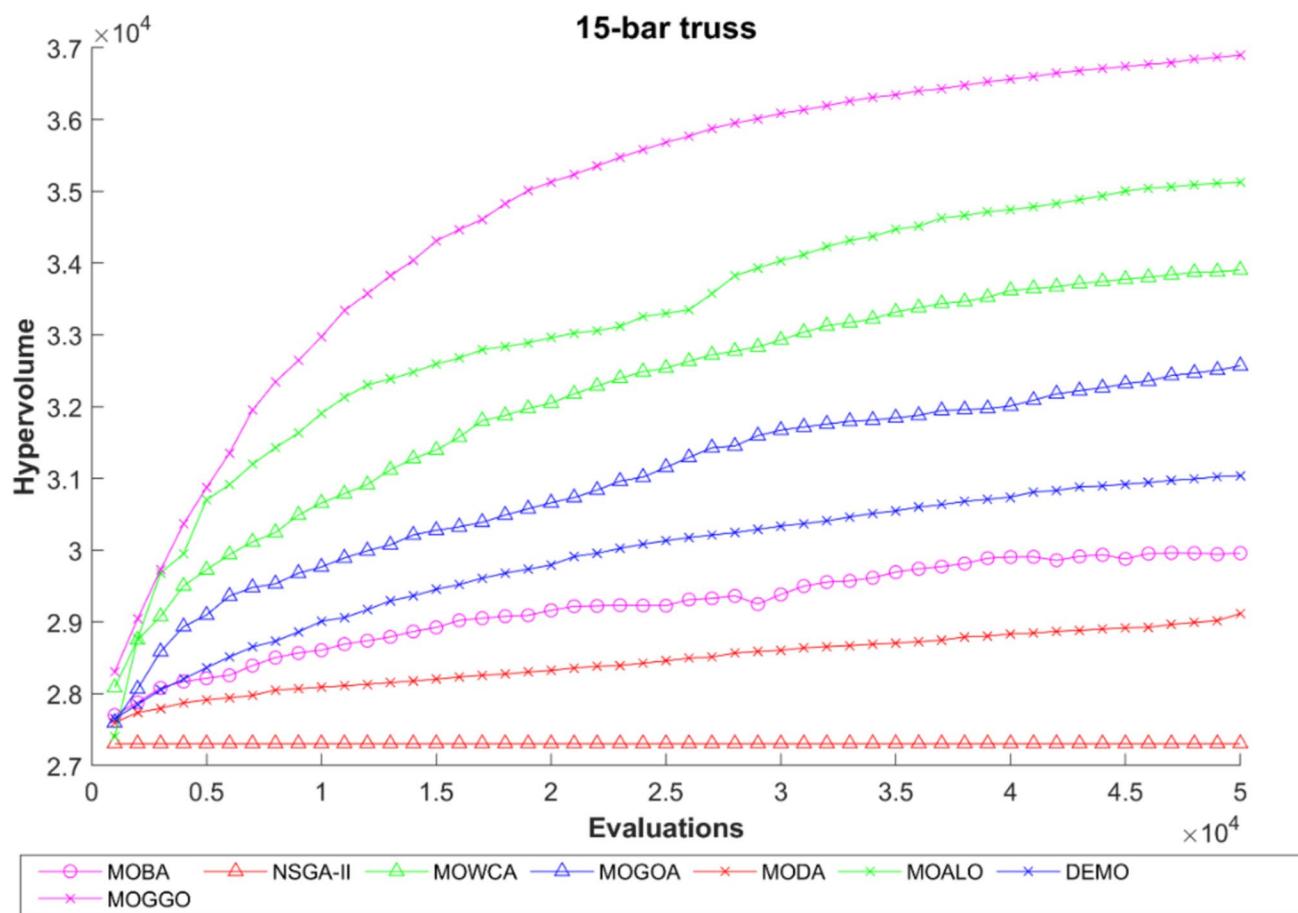


Fig. 12 Comparative hypervolume evolution for 15-bar truss

structural integrity and reliability under various operating conditions. With all design variables encompassing topology and size considerations, the optimization algorithm explores multiple design possibilities to achieve the desired performance objectives. The algorithm aims to identify Pareto-optimal solutions that balance mass minimization and structural reliability maximization by systematically evaluating different configurations and trade-offs between objectives. The algorithm uses rigorous optimisation to deliver optimized truss designs that meet stringent performance requirements across multiple load scenarios.

- The last truss structure, depicted in Fig. 7, presents a 3-D configuration of 224 bars, making it the study's most intricate and challenging design problem. This three-dimensional truss offers high structural complexity and geometric variability, requiring sophisticated optimization techniques for optimal performance. The loading conditions and material properties prescribed in Table 3 serve as critical inputs for the optimization process, guiding the algorithm in its search for efficient design solutions. With all the considered design

variables encompassing topology, shape, and size considerations, the optimization algorithm explores a vast design space to identify optimal truss configurations. The algorithm refines the design solutions through iterative optimisation iterations, ultimately delivering high-performance truss structures capable of withstanding operational demands while minimizing material usage and ensuring structural integrity.

4.2 Non-dominated solutions (NDS) and archiving strategy

Solutions in a population can be arranged using the non-dominated (ND) sorting method according to their Pareto dominance relationship. The first front starts with the identification of a group of ND solutions. The remaining solutions are then assessed to identify more ND solutions and create more fronts. The aforementioned iterative procedure persists until every solution has been classified. Solutions inside the same front are regarded as equal and share the

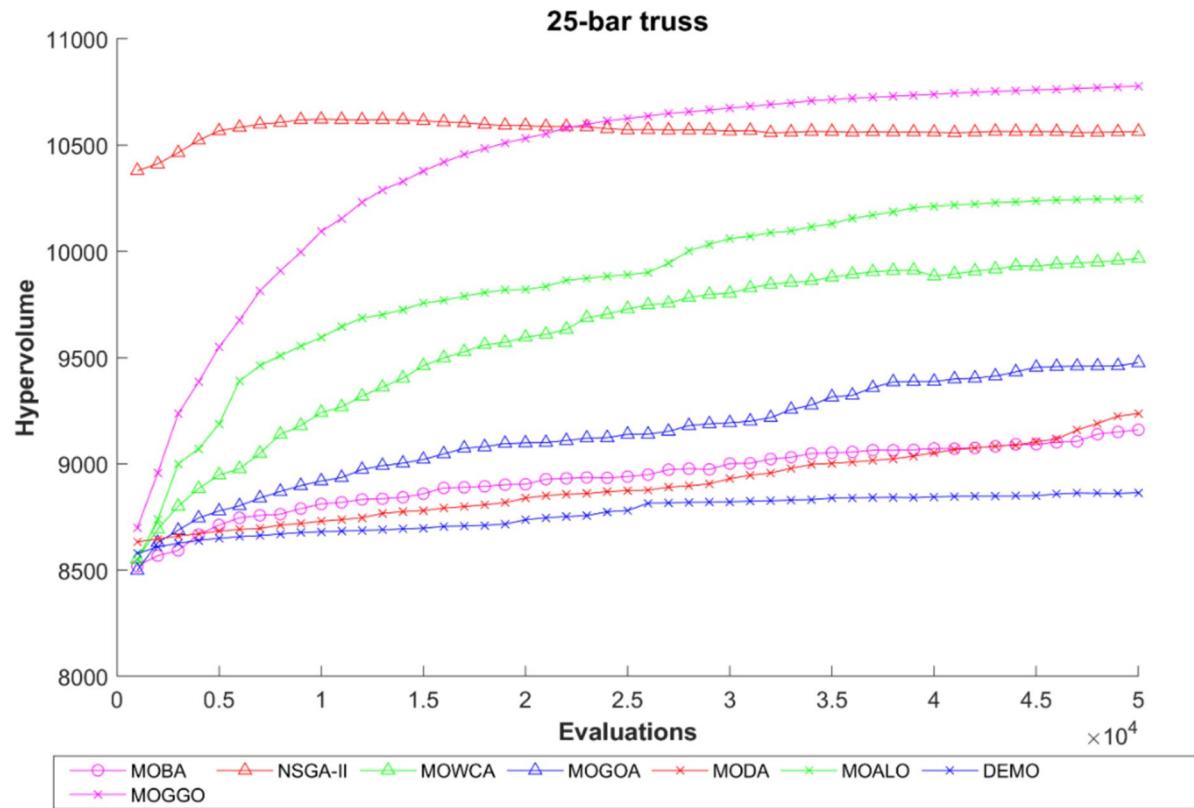


Fig. 13 Comparative hypervolume evolution for 25-bar truss

same rank after ND sorting. Reflecting their dominant relationship, solutions on higher fronts are ranked higher but are somewhat worse than those on lower fronts. Figure 8 shows Pareto dominance, which indicates a solution outperforms another in MO optimization if it is strictly superior in at least one objective and at least as good in all. The non-dominated solution (NDS) set, sometimes called the Pareto front or Pareto set, is the set of possible solutions not dominated by any other solution. This dominance relationship defines it. The efficient frontier of the optimization problem is formed by these non-dominated solutions, which show the best compromises between competing objectives. The feasible solution space contains solutions that meet the restrictions of the problem regardless of their optimality, whereas the objective space contains every possible combination of objective values.

An archive is a repository for the non-dominated (ND) solutions discovered during optimization. When a new solution is encountered, it is added to the archive if no existing solution dominates it. Conversely, if a new solution from the current population dominates one in the archive, the dominant solution is replaced with the new one [54]. In cases where the archive reaches its capacity and a new ND solution is generated, one of the existing solutions is substituted to accommodate the latest addition.

4.3 Empirical evaluations

Performance metrics evaluation is essential to compare the MOGGO algorithm accurately to other MO algorithms. Four well-known measures viz, Hypervolume indicator (HV) [55] which measures the portion of the target space that the non-dominated solution set occupies (higher values are better). The Generational Distance (GD) [56] quantifies the difference between the real Pareto-optimal front and the estimated one discovered throughout the search process (lower values are better). The Inverted Generational Distance (IGD) [42] is a metric used to evaluate the convergence of the obtained Pareto front to the actual Pareto front. Lower numbers indicate better convergence. It calculates the average distance between each point in the produced Pareto front and the nearest point in the actual Pareto front. The spacing to an extent (STE) [43] ratio is a composite statistic that combines the extent (ET) and spacing (SP) metrics. It makes it possible to evaluate a Pareto front's extent and spacing capabilities at the same time. As solutions are well-distributed over the front (low spacing) and cover a wide range of objectives (high extent), a lower STE value denotes a more efficient and non-dominated Pareto front. Accordingly, Table 4 describes the performance measurements applied to multi-objective algorithms.

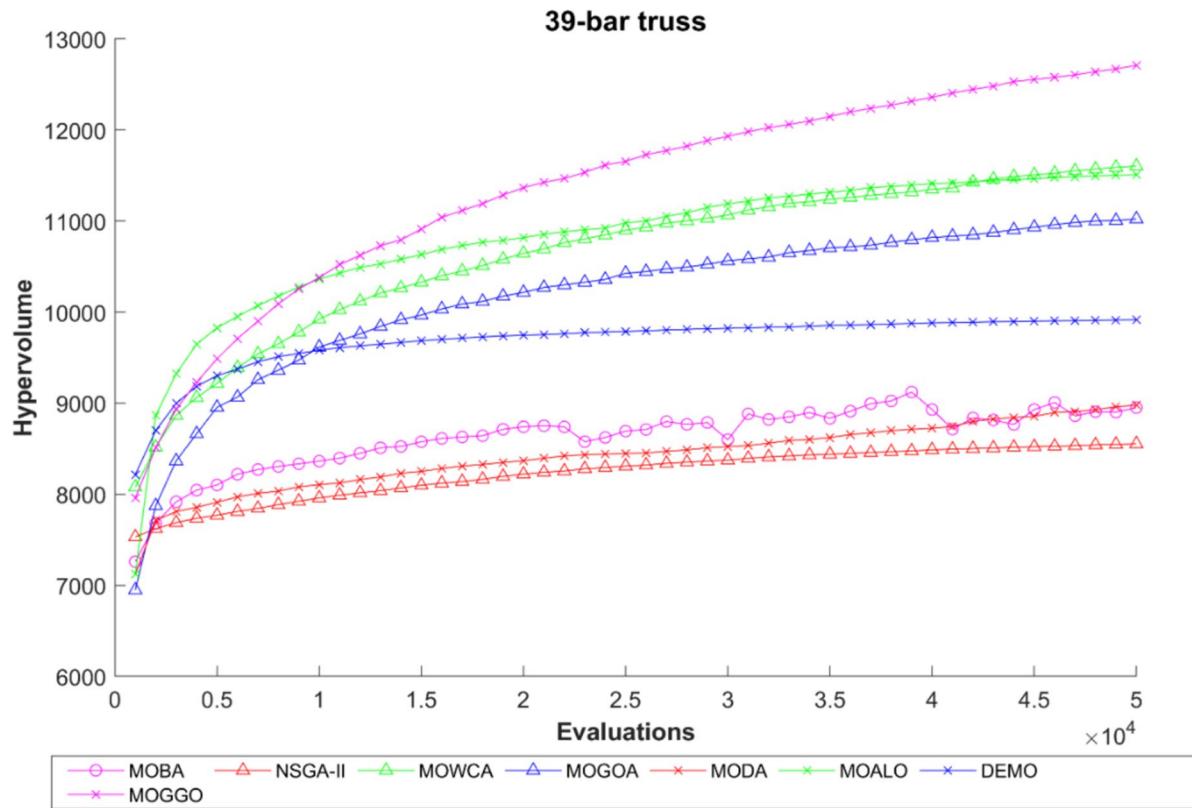


Fig. 14 Comparative hypervolume evolution for 39-bar truss

Figure 9 indicates pictorial representations of GD and IGD matrices for MO optimization. GD is a performance metric that calculates the average Euclidean distance between each point on the estimated Pareto front and the point closest to it on the true Pareto front in MO optimization. IGD, on the other hand, measures the average distance from points on the true Pareto front to their nearest points on the estimated Pareto front. Lower GD and IGD values suggest better convergence of the estimated front to the true front. By analyzing the figures depicting GD and IGD values over generations or function evaluations (FEs) for different MO optimization algorithms, we can assess their convergence behaviour and ability to approximate the true Pareto front effectively.

As per the visualization from Fig. 10a, Spacing-to-Extent (STE) is a metric used in MO optimization to evaluate the quality of non-dominated Pareto fronts. It combines two important aspects: extent and spacing, providing insights into the distribution and spread of solutions on the Pareto front. A lower value suggests a more uniformly distributed and evenly spaced set of solutions, indicating a higher quality Pareto front. HV value measures the volume of the objective space dominated by the solutions in the Pareto front. A higher HV indicates that the Pareto front occupies a more significant portion of the objective space, implying better

coverage and diversity of solutions. Figure 10b shows HV covered by the MOGGO algorithm with reference point A.

5 Results, analysis, and comparative study

5.1 Statistical results

Table 5 contains all considered MO algorithms with specific parameters in their original propositions. This approach ensures a fair and accurate comparison by utilizing the parameter settings originally optimized for each algorithm. Additionally, it is worth mentioning that the number of function evaluations is set to 25,000 for all methods. The evaluation results for the eight MO algorithms across six state-of-the-art truss structures, as depicted in Table 6, provide insights into their performance based on hypervolume measures. These measures include average, maximum, minimum, and standard deviation, offering a comprehensive view of each algorithm's efficacy in balancing the objectives of mass minimization and reliability maximization. The Friedman rank test further contextualizes these results by statistically comparing the performance of the MO algorithms and identifying any significant differences among them. This analysis highlights the trade-offs inherent in multi-objective

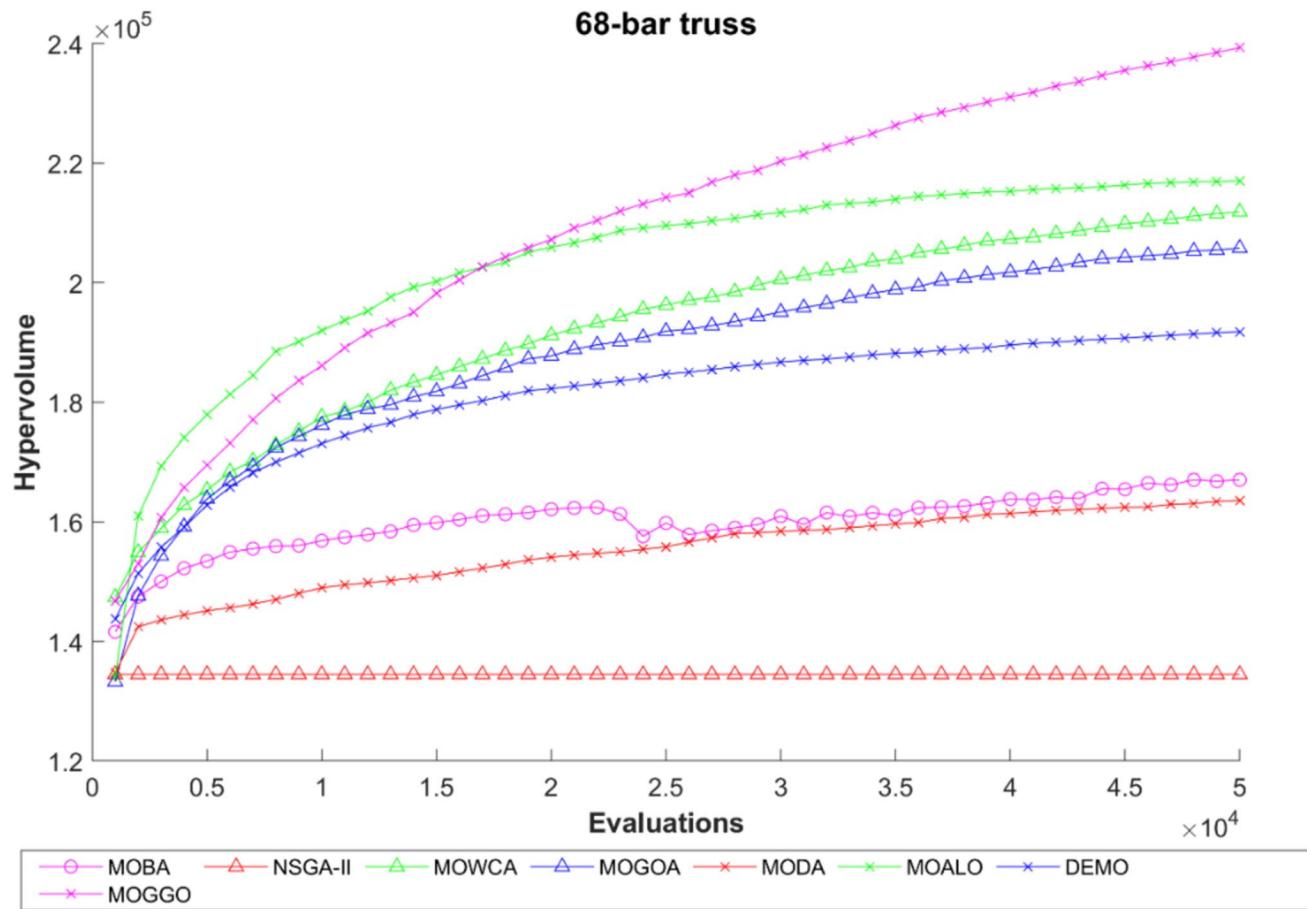


Fig. 15 Comparative hypervolume evolution for 68-bar truss

optimization, where algorithms must navigate between conflicting objectives to generate Pareto-optimal solutions. By considering average and variability metrics, researchers can assess each algorithm's overall performance, consistency, and robustness across different truss structures. Such insights are invaluable for guiding the selection and implementation of MO algorithms in real-world engineering applications, where achieving optimal designs necessitates careful consideration of multiple competing objectives.

5.1.1 Convergence analysis through hypervolume indicator

- HV values for the 45-bar truss demonstrate the highest values of maximum, average, and minimum 4957.15, 4854.58, and 4752.10, respectively, with a narrow standard deviation of 54.43 by MOGGO. NSGA-II is the second-best MO algorithm with second rank, while MODA and DEMO are the worst, with the lowest average HV values of 2977.96 and 2596.14, which shows poor convergence properties.

- Smaller truss structures like 15-bar MOGGO emerged as top performers and displayed stability with narrow standard deviations of 44.54 with rank one and highest average HV of 38,230.27, followed by MOGOA and MOALO.
- For the 25-bar truss again, MOGGO showcases outstanding and robust optimization capabilities with first Friedman's overall rank and highest hypervolume values of 10,890.08 average HV with lowest 6.73 std. NSGA-II and MOGOA are at the second and third rank with an average HV of 10,368.47 and 10,291.13, respectively.
- An average HV for 39-bar and 68-bar trusses are 13,532.48 and 270,778.8, with MOGGO securing the first rank in Friedman's rank test. MOGOA and MOWCA rank second and third, respectively. NSGA-II and MOBA are the worst performers, with the lowest average HV of 8700.90 and 10,257.03 for 39-bar and 134,487.3 and 188,828.0 for 68-bar.
- MOGGO's average HV for a 224-bar large truss is 4,973,946.47, the highest with the least variation of std 27,299. These results demonstrate the superior convergence characteristics of MOGGO. MOGOA and MOWCA are at the second and third rank, while DEMO

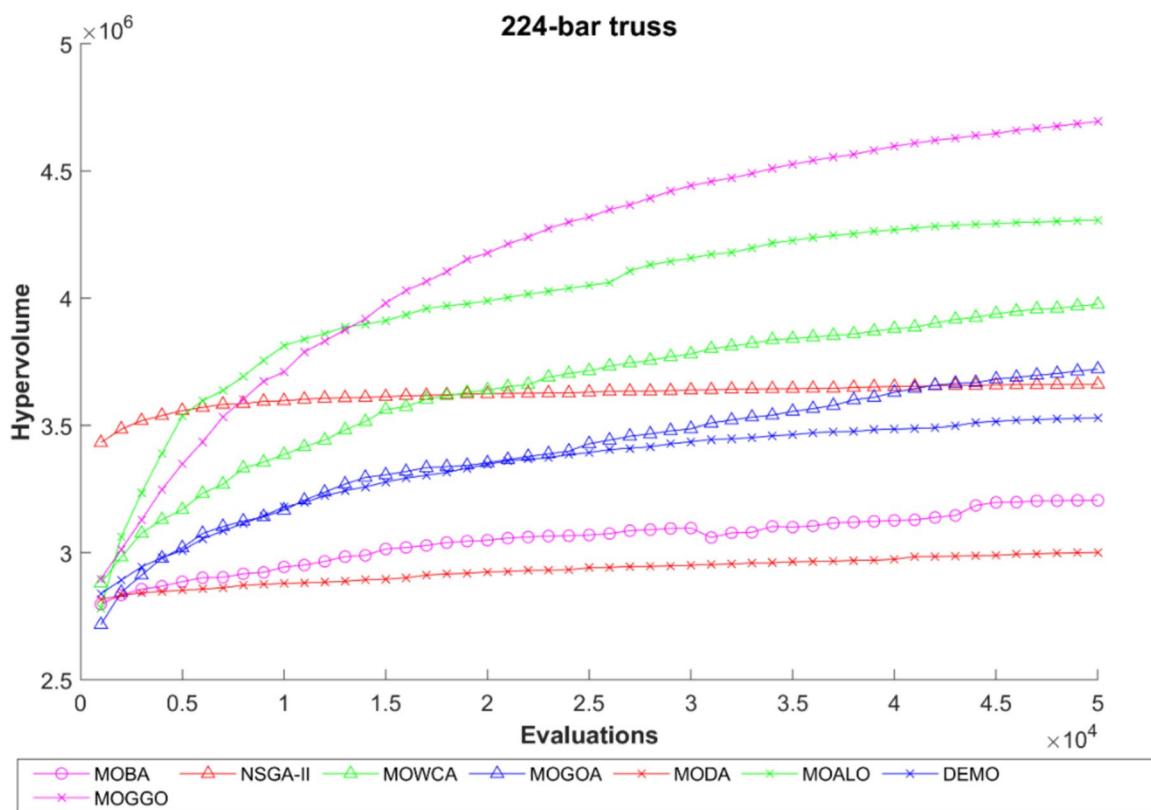


Fig. 16 Comparative hypervolume evolution for 224-bar truss

and MOBA are the worst performers in HV matrix. MOALO and NSGA-II are stable with fourth and fifth ranks, respectively.

- MOGOO exhibits its ability to balance mass reduction and reliability increase by producing well-distributed Pareto-optimal solutions, providing designers with a wide range of design options. Its hypervolume performance demonstrates its superiority and highlights its potential as a stable and dependable optimization method for truss structures, which will improve multi-objective optimization research in structural engineering.

5.1.2 Statistical analysis by hypervolume and Friedman's rank test

- For assessing MO optimization algorithms, Friedman's rank test is essential since it provides a non-parametric method that is resilient to assumptions about data distribution. It assigns a ranking to algorithms according to their performance on various parameters, taking importance and variability into account. It offers insightful information on the efficacy of algorithms and directs

prospective study or implementation decisions because it is versatile and comprehensible.

- The performance of the MOGOO algorithm was compared with several other algorithms, namely MOBA, NSGA-II, MOWCA, MOGOA, MODA, MOALO, and DEMO, on various truss problems as tabulated in Table 6, showing supremacy of the MOGOO algorithm over the competitors. For the 45-bar truss problem, MOGOO obtained a hypervolume of 4854.58, thus beating MOBA by 24.55%, NSGA-II by 3.68% and other competitors. In truss problem 15-bar, MOGOO reached an average hypervolume of 38,230.27 through a significant improvement of 2.26%, surpassing the next best performer, MOGOA, besides surpassing all other algorithms. In the 25-bar truss, the performance is consistent with the achievement of 10,890.08 by MOGOO. 39-bar truss problem finds outstanding excellence of MOGOO as it recorded hypervolume at 13,532.48, showing an impressive improvement of 7.19% over the second-best algorithm. Results for MOGOO were more emphasized in large-scale problems, such as the 68-bar truss, where it presented a hypervolume of 270,778.85,

Table 7 The generational distance (GD) metric

GD		MOBA	NSGA-II	MOWCA	MOGOA	MODA	MOALO	DEMO	MOGGO
45-bar truss	average	3.0289	0.7051	0.5281	1.1533	7.3325	8.7499	49.8671	0.2446
	max	14.4367	5.0654	3.7209	3.7672	18.3734	13.3242	59.9402	0.3138
	min	0.4497	0.1319	0.1080	0.2220	1.1770	3.6360	42.2521	0.1903
	std	3.4559	0.9239	0.7410	1.1235	4.8662	2.4193	4.2322	0.0346
	Friedman rank	5.00	2.80	2.33	3.80	6.00	6.60	8.00	1.47
15-bar truss	average	18.9600	14.0733	1.4017	2.9394	4.4160	5.8970	6.6332	1.6906
	max	105.4592	34.5502	2.2255	5.5568	9.6469	15.2318	15.1087	3.3681
	min	2.2890	3.1056	1.1475	0.7750	2.0076	1.0906	3.5347	0.8476
	std	20.8109	7.7149	0.2092	1.2117	1.8202	3.5961	3.0224	0.7444
	Friedman rank	6.87	7.10	1.70	3.20	4.53	4.97	5.83	1.80
25-bar truss	average	1.1561	0.1231	0.5523	0.5577	1.0656	0.5645	0.6404	0.4472
	max	10.6842	0.1976	0.6861	0.8039	2.7680	1.5857	1.1735	0.5249
	min	0.4229	0.0378	0.4197	0.3757	0.4670	0.3433	0.4053	0.3877
	std	1.8139	0.0387	0.0545	0.1120	0.6898	0.2266	0.1568	0.0360
	Friedman rank	7.10	1.00	4.60	4.47	6.43	4.10	5.63	2.67
39-bar truss	average	7.0648	0.3666	0.6251	1.3798	4.8830	4.0142	9.9340	0.5631
	max	19.0464	1.6803	2.4764	5.2711	12.6249	16.0202	16.8346	1.7968
	min	0.8189	0.1793	0.4148	0.4080	0.4246	0.4754	3.7504	0.3642
	std	6.8412	0.3450	0.3791	1.3304	3.2839	3.5627	2.6883	0.3478
	Friedman rank	6.27	1.43	3.27	3.67	5.87	5.80	7.47	2.23
68-bar truss	average	343.9785	796.9648	44.3723	78.0524	137.5329	107.3220	156.9399	22.3753
	max	630.3463	1648.9223	125.5724	119.1143	220.9216	193.0133	194.5052	84.9741
	min	163.6073	421.6257	9.7456	12.6007	71.0324	45.4109	127.1854	6.5260
	std	131.8165	295.3131	35.9241	25.3743	36.8497	43.1678	16.8618	22.6862
	Friedman rank	7.00	8.00	2.00	3.23	4.87	3.93	5.63	1.33
224-bar truss	average	10,806.5270	4102.0292	501.0272	1774.0337	1833.4600	3370.0551	2966.1246	125.0942
	max	27,454.0909	8348.4973	2956.9189	5163.8551	5132.4234	7773.5917	8147.7490	572.9118
	min	406.4211	1760.1309	131.3614	86.8031	192.3356	91.3164	263.8307	71.0997
	std	7611.8733	1820.0046	775.4188	1497.4566	1397.4482	1581.2115	2017.8424	119.8561
	Friedman rank	7.0000	6.2333	2.4667	4.2000	4.0333	5.6000	5.3333	1.1333
Average Friedman		4.90	3.32	2.05	2.82	3.97	3.88	4.74	1.33
Overall Friedman rank		8	4	2	3	6	5	7	1

surpassing MOBA by 43.65%. In the case of the problem presented in the 224 bar truss, MOGGO attained a hypervolume at 4,973,946.47 and, thus, improved through the MOBA algorithm by 43.36%. Overall, it can be concluded that the MOGGO outperforms its counterparts by achieving higher hypervolume values across the Pareto front in truss optimization problems with better convergence and diversity.

- Friedman's rank test for all MO algorithms for considered truss structures for HV suggests MOGGO's super-

rior performance and best convergence characteristics consistently archive higher HV across different truss configurations, indicating its effectiveness in exploring the trade-off between conflicting objectives.

Figures 11, 12, 13, 14, 15, 16 illustrate the evolution of hypervolume with a maximum of 50,000 FEs across all considered truss structures and the eight multi-objective optimizers. Analysis of these figures indicates a consistent trend wherein MOGGO achieves the highest hypervolume

Table 8 The inverted generational distance (IGD) metric

		MOBA	NSGA-II	MOWCA	MOGOA	MODA	MOALO	DEMO	MOGGO
45-bar truss	Average	7.6214	1.2957	3.6734	6.7322	16.7361	3.7619	22.3156	0.6826
	Max	10.1582	1.9714	10.7712	11.4180	20.3824	11.2994	25.4408	1.7136
	min	5.4767	0.4303	1.0519	3.5548	12.3766	0.9212	17.3609	0.2400
	std	1.4198	0.4843	2.0237	2.1502	1.8033	2.4649	1.9426	0.3777
	Friedman rank	5.47	1.93	3.60	5.17	7.03	3.70	7.97	1.13
15-bar truss	Average	39.4563	76.3680	44.3660	13.0249	38.8638	15.2166	35.5630	14.7746
	max	59.3767	93.4949	66.3311	24.0377	55.5567	30.2448	58.8039	28.6132
	min	21.0341	57.5808	27.9009	7.0393	18.6074	6.5465	9.3053	2.6761
	std	10.2232	9.4533	10.3767	4.4979	8.9461	5.7334	10.4613	7.1719
	Friedman rank	5.63	8.00	5.83	1.73	5.43	2.27	4.90	2.20
25-bar truss	average	7.1620	10.0250	16.7523	4.2132	5.3538	7.6281	13.5473	4.9263
	max	13.8313	16.5806	27.2105	7.4363	11.8706	17.8868	20.0193	9.1501
	min	1.8848	5.7123	5.1307	2.6399	1.5421	3.3720	5.1815	0.7953
	std	3.3922	2.1914	5.1697	1.3792	2.3983	3.0487	4.2061	1.6777
	Friedman rank	4.03	5.63	7.30	2.17	2.90	4.43	6.83	2.70
39-bar truss	average	20.2322	20.3931	22.7502	15.4648	14.7341	11.3858	26.9369	15.8616
	max	42.7980	28.9046	36.8781	22.8912	22.2948	22.1517	34.0564	20.4518
	min	6.2254	8.9604	9.5589	9.6179	7.3351	3.3661	18.8445	10.1935
	std	8.5495	4.5146	6.0282	3.5292	3.6264	5.0736	3.2681	2.3720
	Friedman rank	4.93	5.40	5.93	3.27	3.17	2.17	7.40	3.73
68-bar truss	average	795.6945	1012.5836	890.6633	731.8677	576.4156	310.5720	569.4769	904.8492
	max	1114.2020	1169.1985	1071.3148	984.3934	968.7361	619.0119	783.6686	1143.9163
	min	328.3469	894.1950	558.8164	530.4065	329.7721	110.2345	398.3886	402.7928
	std	165.9649	61.5397	142.1793	119.2913	170.4415	107.2931	95.1719	198.1130
	Friedman rank	4.97	7.33	5.87	4.47	3.07	1.17	2.73	6.40
224-bar truss	average	10,682.6234	4895.1331	4290.1493	1831.1159	11,390.7979	2978.7889	9358.5746	651.6563
	max	14,288.8584	5918.7557	7421.0203	2943.9710	14,503.9812	7572.5579	12,635.7637	2547.6902
	min	6459.6625	3773.4083	2643.1064	1123.6395	8808.4645	853.0110	3682.3837	108.1813
	std	2016.9178	561.4246	1199.2997	448.0953	1245.7627	1488.1791	1765.9674	591.1667
	Friedman rank	7.03	4.67	4.00	2.27	7.50	3.07	6.40	1.07
Average Friedman		4.01	4.12	4.07	2.38	3.64	2.10	4.53	2.15
Overall Friedman rank		5	7	6	3	4	1	8	2

values as FEs progress. This suggests that MOGGO effectively explores diverse regions of the search space and evaluates a wide range of potential solutions for all truss structures. The highest hypervolume measures attained by MOGGO reflect its superior performance in MO-reliability optimization, underscoring its ability to generate solutions that offer significant improvements in both mass minimization and reliability maximization objectives. Overall, these findings highlight the efficacy of MOGGO in navigating complex optimization landscapes and identifying high-quality solutions that balance competing objectives across various truss structures.

5.1.3 Effectivity analysis by generational distance

Table 7 presents the results of the GD metric, a significant measure used to assess the disparities between the Pareto optimal front and ND solutions across different truss configurations. A reduced GD score signifies an outstanding, non-dominated front.

- For the 45-bar and 15-bar truss, MOGGO's average GD values are 0.2446 and 1.6906, respectively, the lowest among all MO optimizers with Friedman rank 1. DEMO and MOALO are the worst, with higher GD values suggesting a higher distance between true Pareto and Pareto generated by MO algorithms.
- Similarly, for 25-bar and 39-bar truss, average GD matrices by MOGGO are 0.4472 and 0.5631 with very few

Table 9 The spacing—to—extent (STE) metric

		MOBA	NSGA-II	MOWCA	MOGOA	MODA	MOALO	DEMO	MOGGO
45-bar truss	average	0.0351	0.0112	0.0129	0.0038	0.0125	0.0160	0.0226	0.0051
	max	0.0748	0.0413	0.0245	0.0068	0.0319	0.0313	0.0525	0.0111
	min	0.0114	0.0054	0.0067	0.0014	0.0042	0.0064	0.0121	0.0033
	std	0.0172	0.0067	0.0047	0.0016	0.0069	0.0060	0.0087	0.0019
	Friedman rank	7.4333	4.0333	4.8000	1.3000	4.3333	5.4667	6.7000	1.9333
15-bar truss	average	0.0266	0.1037	0.0162	0.0068	0.0119	0.0211	0.0084	0.0082
	max	0.1081	0.2199	0.0733	0.0098	0.0211	0.0407	0.0198	0.0225
	min	0.0060	0.0258	0.0050	0.0035	0.0064	0.0096	0.0038	0.0051
	std	0.0194	0.0497	0.0132	0.0017	0.0043	0.0078	0.0048	0.0033
	Friedman rank	6.1000	7.9333	4.5333	2.2333	4.1000	6.0000	2.3667	2.7333
25-bar truss	average	0.0129	0.0445	0.0126	0.0068	0.0090	0.0293	0.0071	0.0068
	max	0.0278	0.0961	0.0292	0.0132	0.0160	0.0546	0.0269	0.0094
	min	0.0081	0.0102	0.0055	0.0007	0.0050	0.0069	0.0045	0.0059
	std	0.0045	0.0266	0.0055	0.0032	0.0025	0.0143	0.0042	0.0007
	Friedman rank	5.6667	7.5000	5.0333	2.5667	4.0667	6.5667	2.1333	2.4667
39-bar truss	average	6.67E+18	4.41E-02	0.0142	0.0056	0.0111	0.0215	0.0090	0.0059
	max	1.00E+20	8.36E-02	0.0389	0.0128	0.0243	0.0543	0.0154	0.0131
	min	0.0147	0.0020	0.0050	0.0017	0.0039	0.0077	0.0056	0.0044
	std	2.54E+19	2.46E-02	0.0073	0.0023	0.0048	0.0108	0.0023	0.0018
	Friedman rank	7.1333	7.1000	4.6667	1.7333	4.0333	5.9000	3.5333	1.9000
68-bar truss	average	547,343.3	3,131,442	0.0134	0.0043	0.0134	0.0170	0.0060	0.0080
	max	16,420,296	91,008,136	0.0312	0.0069	0.0373	0.0295	0.0118	0.0227
	min	0.0208	0.0111	0.0072	0.0014	0.0029	0.0068	0.0038	0.0043
	std	2,997,922	16,605,896	0.0064	0.0011	0.0085	0.0062	0.0017	0.0043
	Friedman rank	7.3000	7.5667	4.7667	1.3333	4.4000	5.3000	2.3000	3.0333
224-bar truss	average	3.33E+18	3.01E-02	0.0108	0.0044	0.0393	0.0214	0.0291	0.0080
	max	1.00E+20	7.38E-02	0.0198	0.0071	0.1185	0.0423	0.1159	0.0176
	min	0.0289	0.0049	0.0058	0.0020	0.0075	0.0077	0.0086	0.0050
	std	1.83E+19	1.53E-02	0.0038	0.0013	0.0318	0.0101	0.0232	0.0036
	Friedman rank	7.4667	5.6333	3.1667	1.1333	5.8000	5.1333	5.3000	2.3667
Average Friedman		5.14	4.97	3.37	1.29	3.34	4.30	2.79	1.80
Overall Friedman rank		8	7	5	1	4	6	3	2

Table 10 The overall Friedman ranks

	MOBA	NSGA-II	MOWCA	MOGOA	MODA	MOALO	DEMO	MOGGO
45-bar truss	5.58	2.71	3.51	3.84	6.10	5.22	7.66	1.38
15-bar truss	6.18	7.76	4.12	2.30	4.93	4.11	4.67	1.93
25-bar truss	5.65	4.35	5.38	3.30	4.72	4.79	5.60	2.21
39-bar truss	6.08	5.45	4.28	2.71	4.52	4.48	6.28	2.22
68-bar truss	6.48	7.73	3.86	2.84	4.48	3.65	4.03	2.94
224-bar truss	7.13	5.67	3.32	2.43	6.11	4.28	5.69	1.39
Average Friedman rank	4.64	4.21	3.06	2.18	3.86	3.32	4.24	1.51
Overall Friedman rank	8	6	3	2	5	4	7	1

standard deviations of 0.0360 and 0.3478. NSGA-II and MOGGO are the best two algorithms for 25-bar and 39-bar. MOBA, MODA, and DEMO are the last three on the list with higher GD values, as shown in Table 7.

- For large trusses, viz. 68-bar and 224-bar, MOGGO emerged first in Friedman's rank test with average GD values of 22.3753 and 125.0942. MOWCA and MOGOA are at the second and third ranks with an average of

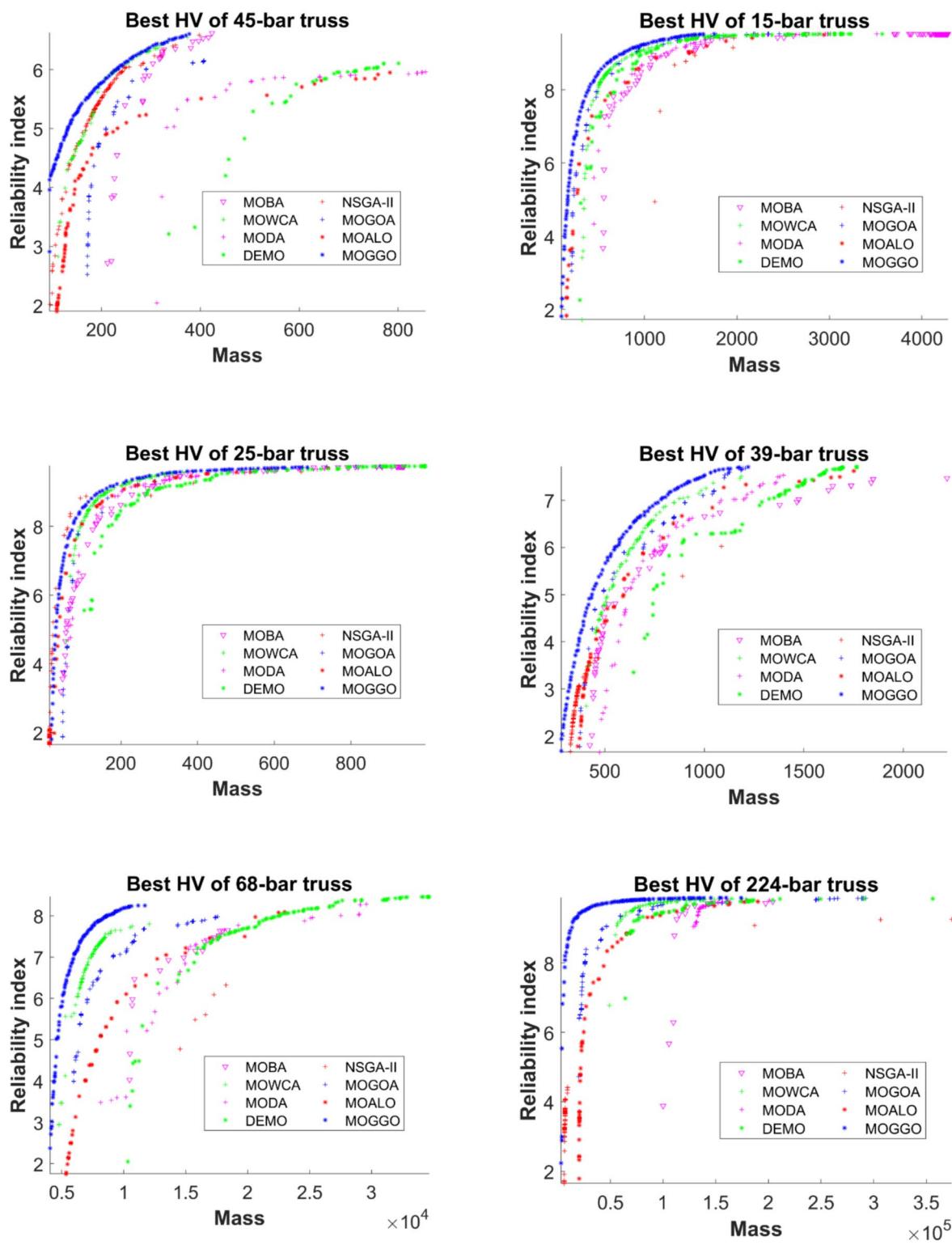


Fig. 17 Best Pareto fronts of the considered truss structures

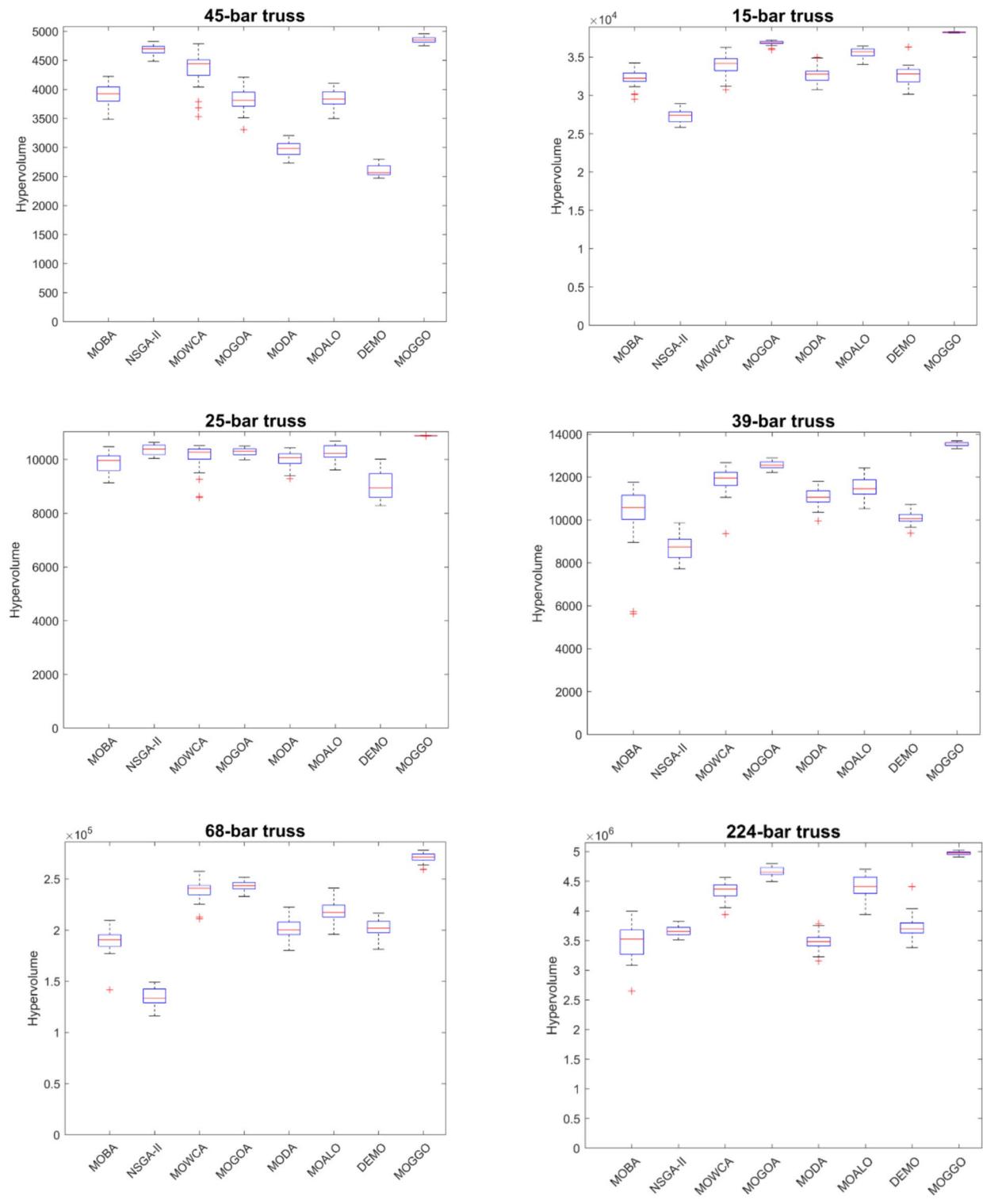


Fig. 18 Boxplots of the considered truss structures

44.3723 (68-bar), 501.0272 (224-bar), 78.0524 (68-bar), and 1174.0337(224-bar), respectively.

- Overall, MOGGO emerged as very efficient, with better convergence and high-quality solutions. NSGA-II is the

worst performer for a 68-bar truss with a very high std of 295.3131, and MOBA is for a 224-bar truss with a high std of 7611.8733. higher std indicates more significant variability in the distance between solutions obtained by

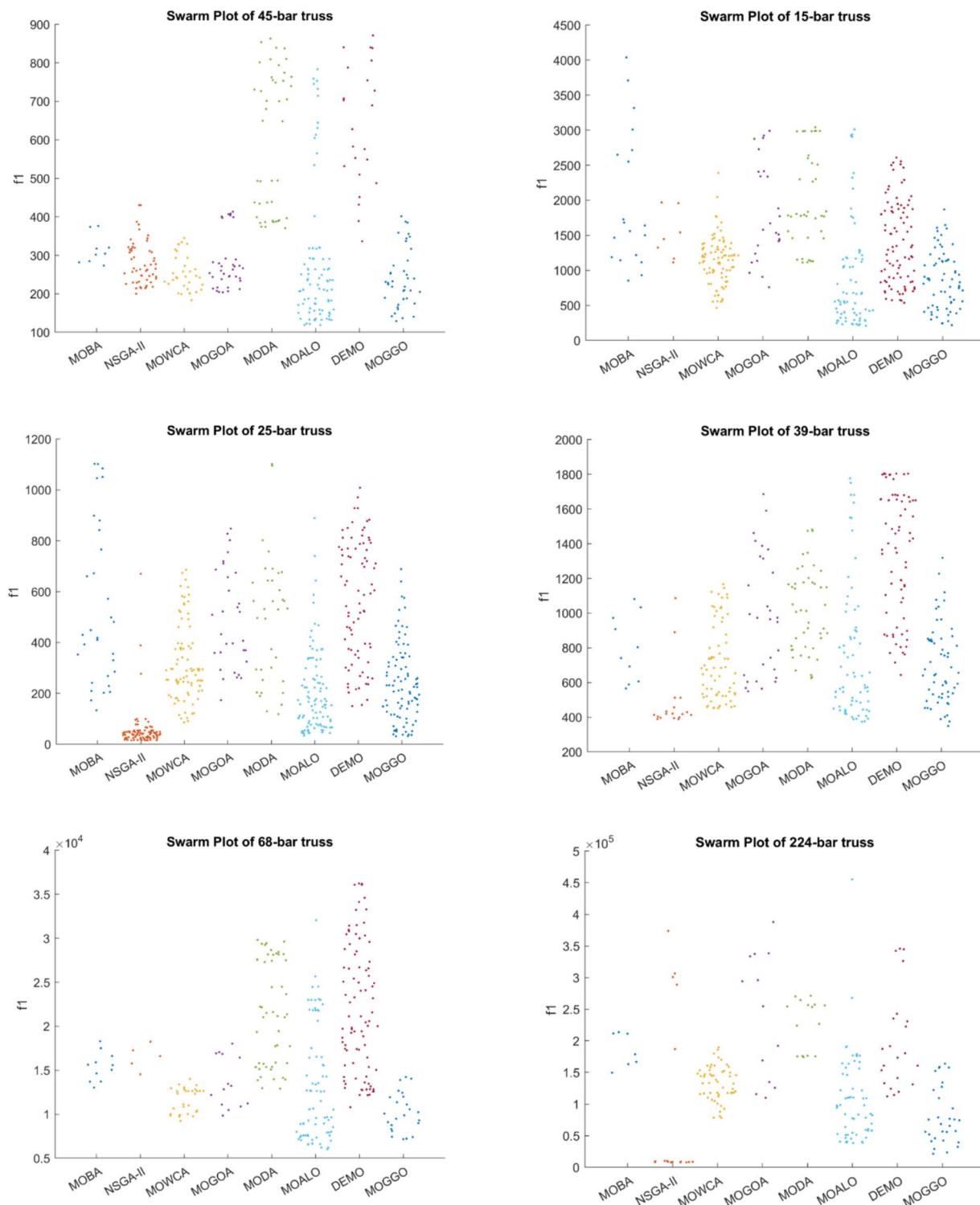


Fig. 19 Swarm plots of Structural mass for the considered truss structures

the MO algorithm and the true Pareto front, showcasing poor convergence and diversity. A high std value alongside it implies that the algorithm's performance fluctuates

significantly from one run to another. It may also indicate instability and unpredictable behavior of the MO algorithms in finding the finest Pareto fronts.

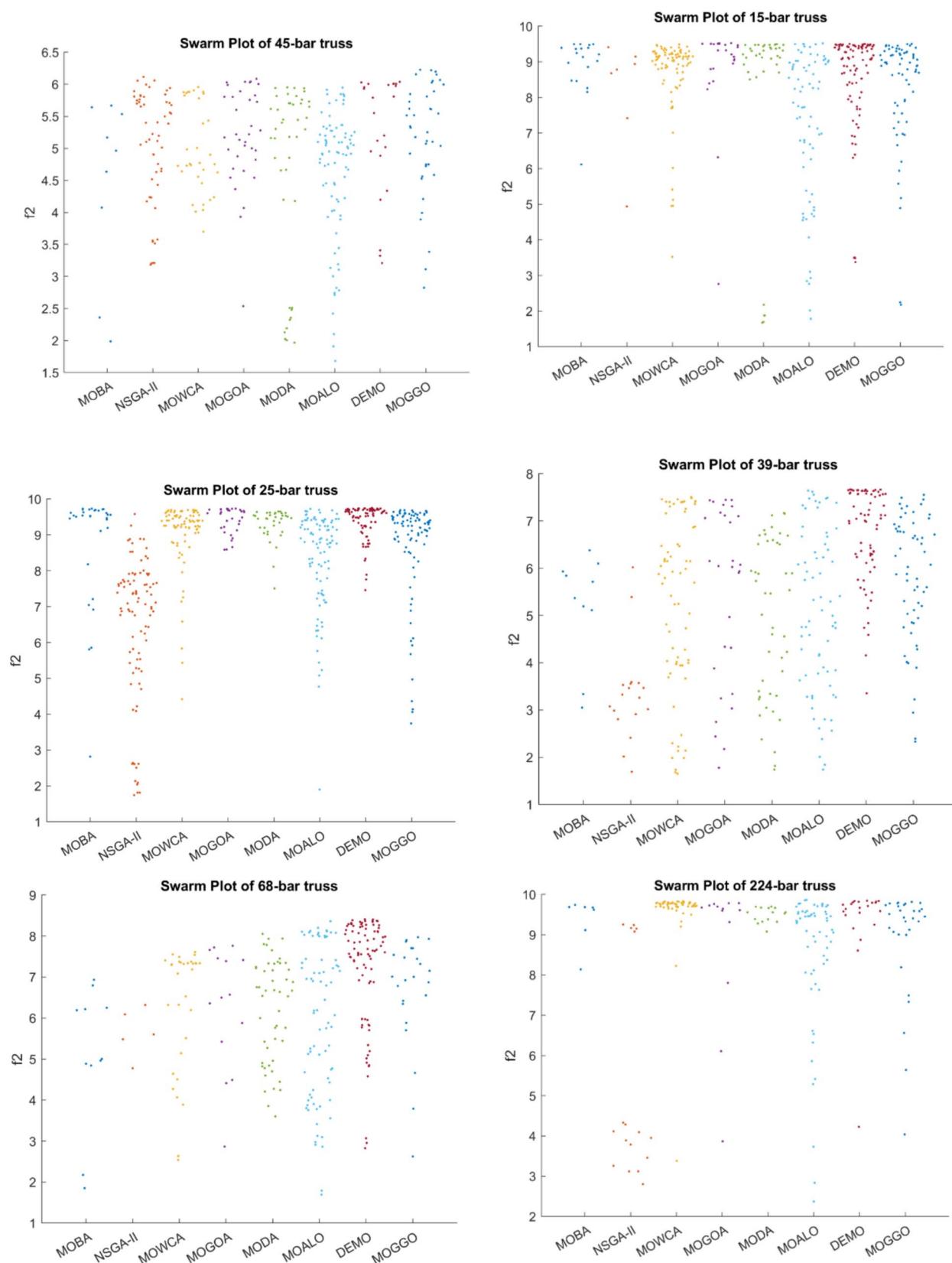


Fig. 20 Swarm plots for reliability for the considered truss structures

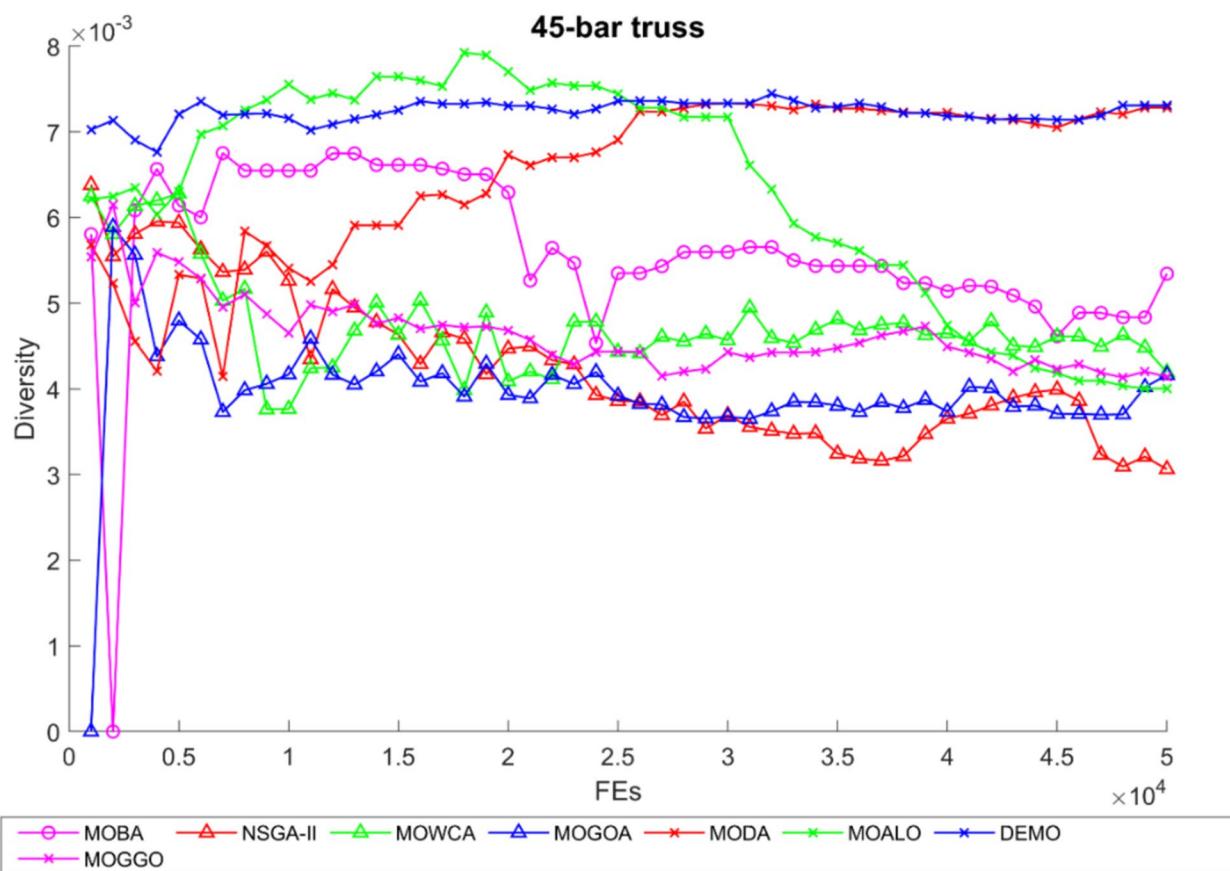


Fig. 21 The diversity curve for a 45-bar truss

5.1.4 Diversity analysis by inverted generational distance

As per Table 8, the IGD values showcased considered algorithms for all structures in which lower values suggest better and superior non-dominating fronts. The IGD average values for MOGGO are 0.6826, 14.7746, 4.9263, 15.8616, 904.8492, 651.6563, for 45-bar to 224-bar respectively, which are lower compared with other well-known considered optimizers. MOGGO and MOALO are the two best algorithms, and they indicate generating superior Pareto fronts and elucidating their strengths and limitations regarding convergence and diversity. Lower IGD values suggest the ability of the MO algorithm to quantify how well solutions are distributed across the objective space, and lower IGD indicates better diversity. This is important because a wide range of solutions gives decision-makers more excellent options and enlightens them about the trade-offs between competing goals. These observations offer insightful information about the field of multi-objective optimization for challenging reliability-based truss design problems.

A comparison of the performance of the MOGGO algorithm with that of several others has been performed, such as MOBA, NSGA-II, MOWCA, MOGOA, MODA, MOALO, and DEMO. The average IGD of the algorithms is in the Table 8. For the 45-bar truss problem, the average IGD for the MOGGO was 0.6826, a 91.05% improvement over MOBA with an average IGD of 7.6214. The average IGD achieved by MOGGO on the 15-bar truss problem is 14.7746, better than that of MOBA at an IGD of 39.4563 with a success rate of 62.62%. On the 25-bar truss problem, there was an improvement in the IGD of 4.9263 for MOGGO compared to MOBA's 7.1620, with a success percentage of 31.67%. On the 39-bar truss problem, the average IGD of 15.8616 for MOGGO came out better than that of MOBA at 20.2322, with improvements of 21.79%. In the 68-bar truss problem, MOGGO improved to an average IGD of 904.8492 with an 11.67% improvement from MOBA at 1021.9936. Regarding the 224-bar truss problem, an average IGD of 651.6563 was recorded for MOGGO, translating to a 66.63% improvement over MOBA's massive 10,806.5270. Overall, Friedman's rank was 2.15 in favor of MOGGO, focusing on its competitive performance in achieving lower IGD values on all test truss configurations. Results indicate that

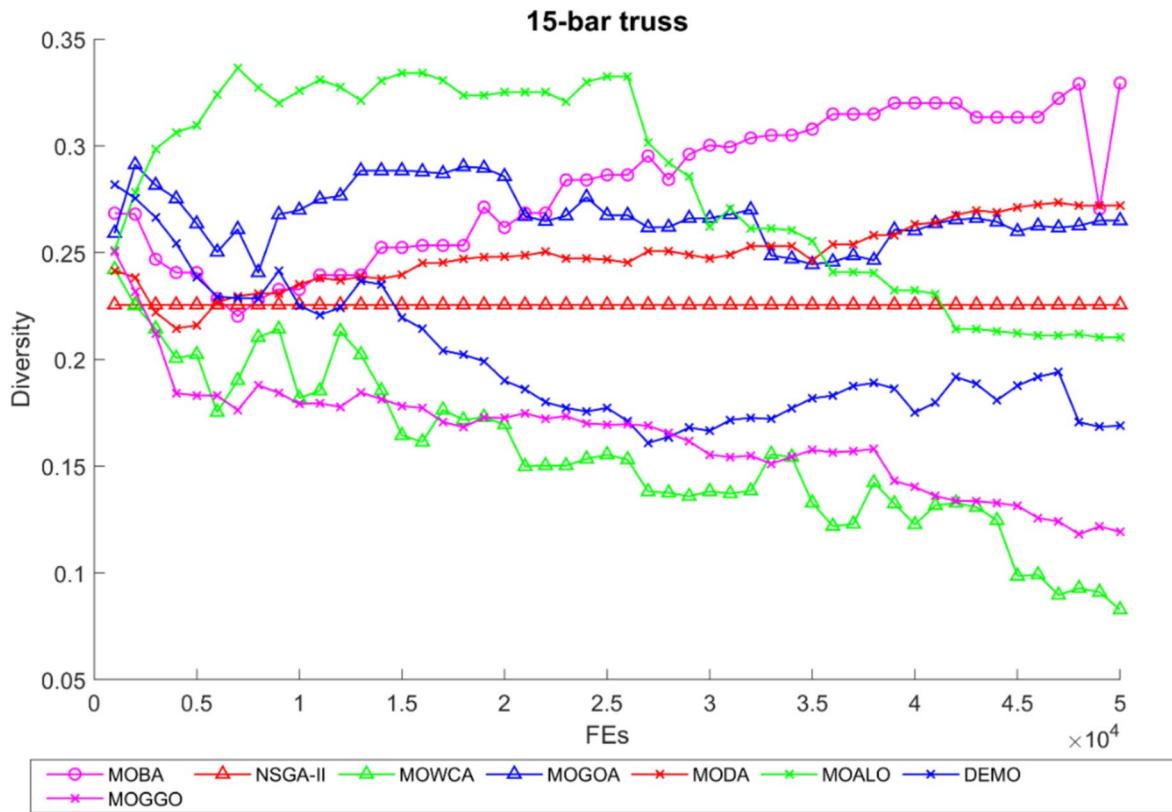


Fig. 22 The diversity curve for a 15-bar truss

MOGGO sustains superior convergence behavior at every step and effectively traverses the multi-modal landscape of multi-objective truss optimization.

5.1.5 Quality assessment of ND solutions by spacing to extent

The STE metric assesses both spacing and extent concurrently, providing crucial insights into the quality of non-dominated fronts. The distribution or arrangement of solutions along the Pareto front is referred to as the spacing aspect of the STE metric. It measures the degree to which the solutions are dispersed evenly along the front. A more evenly spaced distribution indicates better coverage of the objective space and possibly greater diversity among the solutions. Conversely, the coverage or range of the Pareto front in the objective space is referred to by the extension aspect of the STE measure. It quantifies how much the solutions on the front cover the objective space. The Pareto front covers a broader range of objective values to a greater extent, which might be advantageous for offering an extensive selection of trade-off solutions. As per Table 9, a lower STE score signifies a superior, more balanced spacing-to-extent ratio for all considered trusses, indicating a more optimal non-dominated front. From the findings, it can be seen that

MOGGO and MOGOA have the lowest value of average STE and are the top two MO optimization algorithms with a 95% significance level in Friedman's statistical test. The lower value of STE for all trusses indicates a more balanced ratio with more optimal non-dominated fronts.

Table 10 presents a comprehensive overview of the Friedman rank across all truss structures for the evaluated techniques. Among these algorithms, MOGGO achieves the lowest average Friedman's score and secures the top rank compared to MOBA, NSGA-II, MOWCA, MOGOA, MODA, MOALO, and DEMO. MOGGO demonstrates notably superior convergence rates compared to other prominent multi-objective optimization algorithms. Compared to the 45-bar truss problem results, MOGGO earned a rank of 1.38. That indicates an improvement of 75.31% over MOBA with a rank of 5.58. For the 15-bar truss problem, the ranking of 1.93 by MOGGO depicts an improvement of 68.75% compared to the 7.76 rank of NSGA-II. In the case of the 25-bar truss problem, the ranking of 2.21 by MOGGO shows an enhancement of 61.03% compared to the ranking of MOBA as 5.65. For the 39-bar truss problem, the rank that MOGGO attained was 2.22, hence 63.49% better than the rank of 6.08 achieved by MOBA. For the 68 bar truss problem, the rank trend of achieving better rank continued where MOGGO obtained a rank of 2.94, thus a 54.75% improvement relative

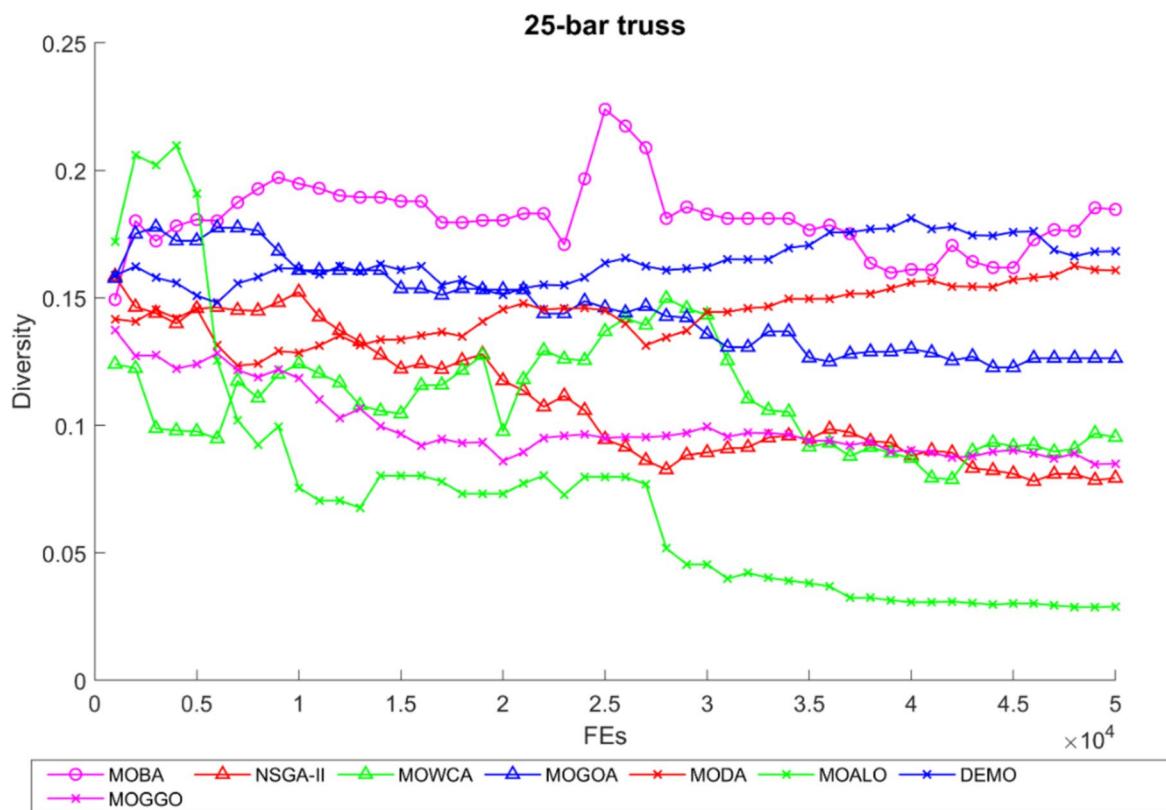


Fig. 23 The diversity curve for a 25-bar truss

to the rank of 6.48 attained by MOBA. For the 224 bar truss problem, the rank of 1.39 was achieved by MOGGO, thus an improvement of 80.46% concerning the rank of MOBA, which was 7.13. The overall rank assigned by the Friedman procedure to MOGGO is 1.51, which once again confirms the position of this algorithm to be the best for the given set in one test run and has been persistently performing in all the different truss configurations. These tests thus prove the superiority of convergence as well as the capability of holding low ranks of MOGGO in the multi-objective optimization tasks. This dominance of MOGGO is statistically significant, supported by Friedman's rank test at a 95% confidence level, highlighting its superior performance relative to the other algorithms examined in the study. In summary, MOGGO exhibits the highest HV values, showcasing its thorough exploration and diverse solution set. Additionally, it consistently maintains the lowest GD and IGD across various scenarios, reflecting a favourable balance between convergence and diversity. MOGGO is the most effective algorithm for addressing reliability-based MO truss structure problems when considering all three metrics. To put it briefly, MOGGO stands out as the preferred option for these intricate structural problems due to its ability to generate a well-distributed set of near-optimal and balanced solutions.

Figure 17 represents the best Pareto fronts achieved by all eight MO algorithms, including MOGGO, for comparing both objectives mass and reliability for all considered truss structures. These fronts consist of non-dominating solutions that minimize structural mass while maximizing the reliability of the structures with a probability of failure of not more than 5%. MOGGO generates uniform and good-quality fronts, which indicates a trade-off between conflicting objectives. MOGGO optimization algorithms systematically explore the design space to identify optimal Pareto fronts, evaluating numerous design alternatives to determine the best compromises. These Pareto fronts provide a range of feasible design options, empowering to make informed decisions tailored to their specific requirements and preferences.

5.2 Boxplots and swarm chart analysis

Figure 18 displayed a visual summary of the hypervolume distribution for all considered MO algorithms for all considered truss structures for reliability-based optimization. Boxplots help compare the central tendency and distribution of objective values produced by various algorithms. The median is shown by a line inside each box in the diagram,

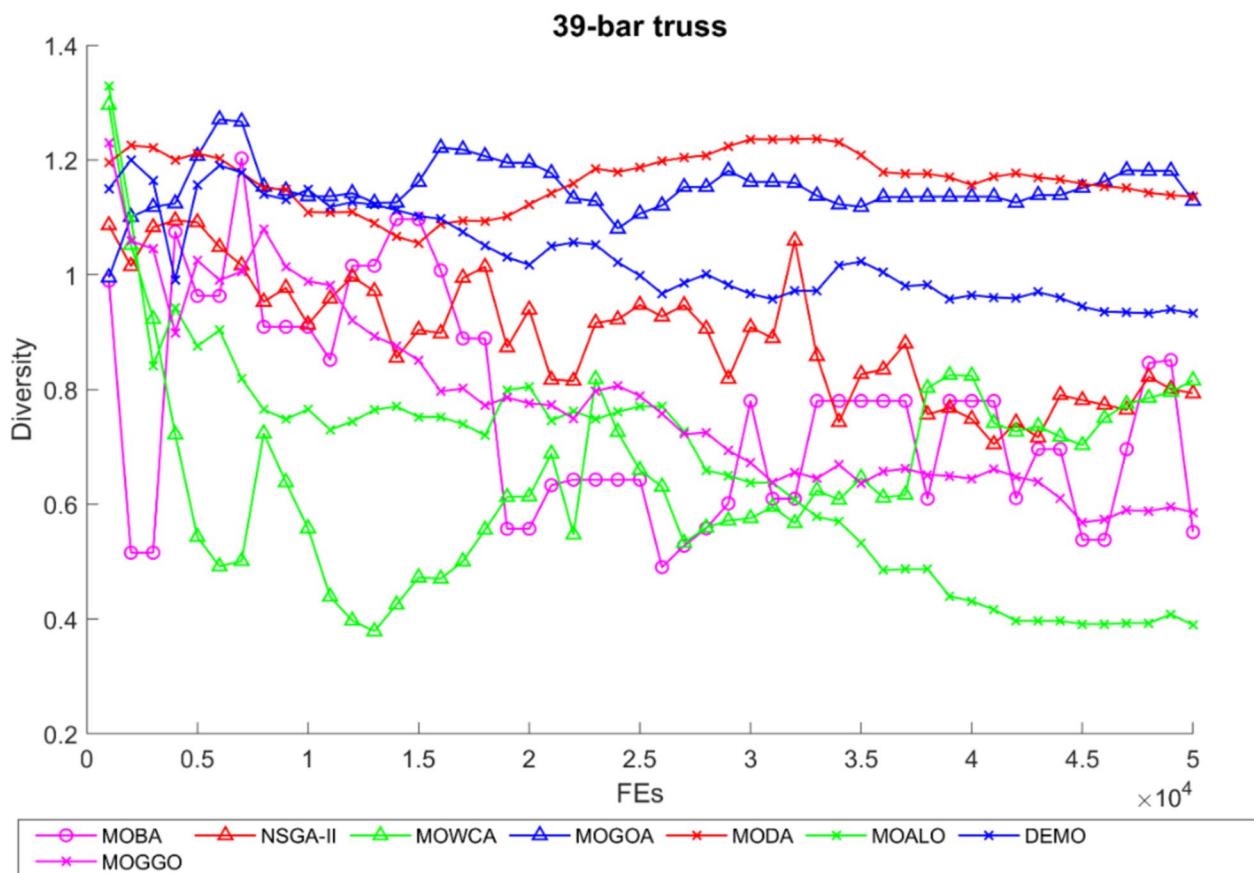


Fig. 24 The diversity curve for 39-bar truss

indicating the objective values' interquartile range (IQR). With outliers removed, the whiskers stretch to depict the range of the data. Boxplot analysis helps analyze solutions' overall distribution and discover differences in performance between different algorithms. The Boxplots generated by the MOGGO for all considered MO algorithms indicate low variability in the HV distributions as they are closely clustered. MOGGO consistently performs well with narrow Boxplots, indicating a high degree of convergence towards a specific region of the objective space, where the solutions produced by the MOGGO converge closely. It also shows stability and reliability in the performance of the MOGGO algorithm.

Figure 19 showcases the swarm plots of the first objective value (Structural mass). In contrast, Fig. 20 shows the second objective value (Reliability of the structure) of all the considered MO algorithms for all considered truss structures. These swarm plots provide valuable insights into the distribution and spread of both objective function values generated by eight MO algorithms for considered structures. Solution distribution by MOGGO for all considered trusses with the healthy spread of the data points indicates dispersion in the objective function values. Apart from this, the solutions generated by MOGGO do not have any outlier solution that deviates significantly from

the majority solution, and a wider spread suggests more significant variability. Also, they indicate insights into the trade-off between solutions clustering around specific values representing good convergence and spread of the solutions across the solution space, which means good diversity.

5.3 Convergence and diversity analysis by diversity curves

Figures 21, 22, 23, 24, 25, 26 represent diversity curves with all 50,000 FEs by all considered MO optimization algorithms for considered truss structures. Comparing diversity curves of different algorithms allows one to evaluate their ability to balance convergence and diversity. Algorithms maintaining higher diversity levels while converging towards the Pareto front are often considered more robust and effective.

- **Convergence Rate of MOGGO:** Diversity curves show how quickly or slowly the solutions converge towards the Pareto front as the number of FEs increases. A step decrease in diversity indicates rapid convergence, sug-

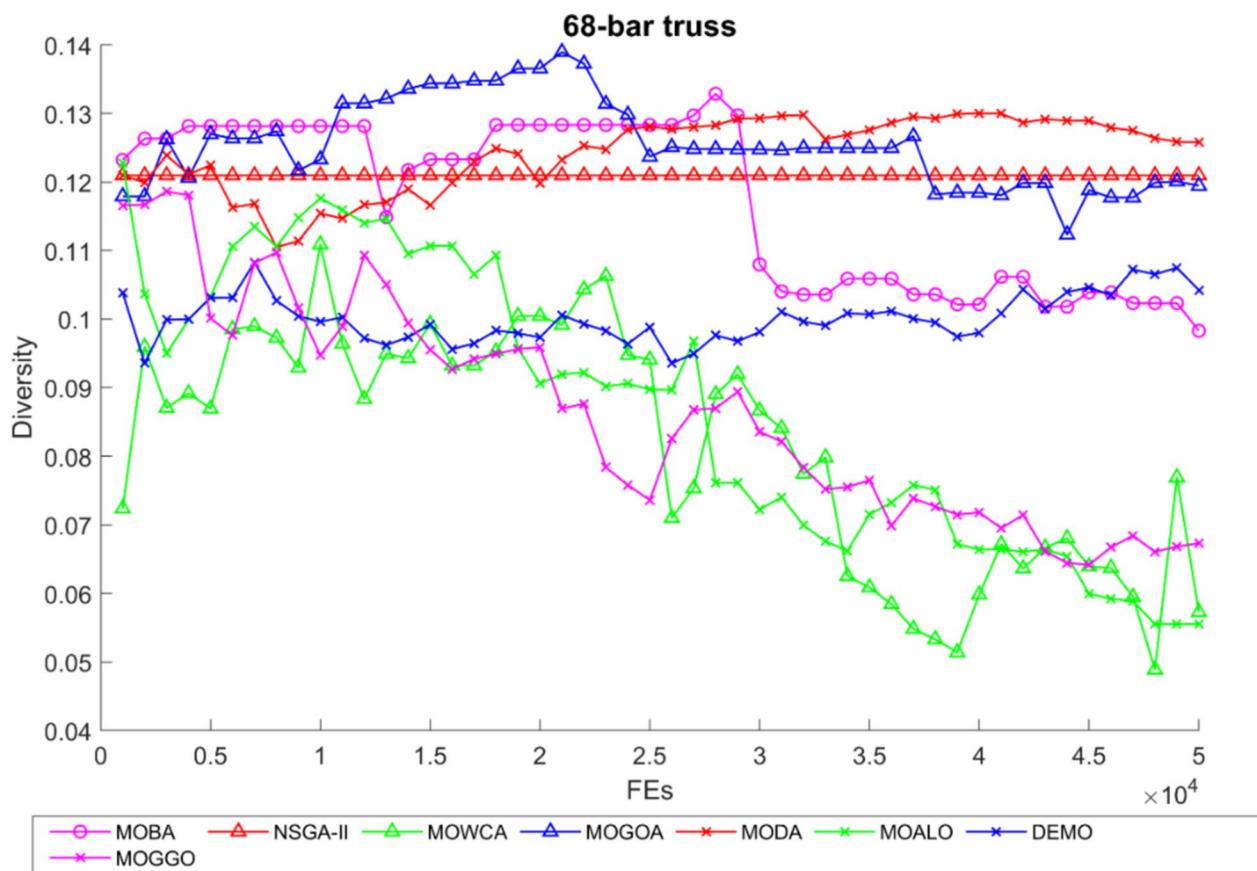


Fig. 25 The diversity curve for 68-bar truss

gesting that the algorithm is effectively refining solutions towards the Pareto front.

- Diversity maintenance by MOGGO: By observing fluctuations in the diversity curve, we can assess how well the algorithm maintains diversity among solutions while converging towards the Pareto front. A consistently decreasing diversity curve may indicate a lack of preservation, leading to premature convergence or suboptimal solutions. MOGGO manages diversity well in finding the optimal Pareto front and generating the best solutions.
- Identification of stagnation: The fluctuations in the diversity curve may signal stagnation or lack of progress in the optimization process. Detecting such patterns can prompt researchers to adjust algorithm parameters or explore alternative approaches to overcome stagnation and improve solution quality. MOGGO's diversity curve is smooth, and due to its decisive exploitation phase, it is without stagnation for most of the truss structures.
- Insights into optimization dynamics: Diversity curves provide insights into optimization dynamics, revealing how solutions explore and exploit the solution space over time. Understanding these dynamics can inform algorithmic improvements and guide the development of more

efficient optimization strategies. With two robust exploration and exploitation phases with dynamic connections, MOGGO emerged as a significant MO optimization algorithm for truss reliability-based structure design.

6 Conclusion

In this paper, we investigated the mass and reliability-based multi-objective optimization of truss structures using the MOGGO algorithm inspired by the dynamic behavior of Greylag geese and using non-dominated sorting and archiving. We aimed to evaluate how beneficial MOGGO is for enhancing several truss design problems. From the results of the experimentation, we were able to show that MOGGO can come up with Pareto-optimal solutions that are quite satisfactory in terms of both weight reduction and reliability improvement. Its optimization strategy, involving non-dominated sorting and archiving, helped to preserve solution diversification and quality and thus strengthen the algorithm. Friedman's rank test for

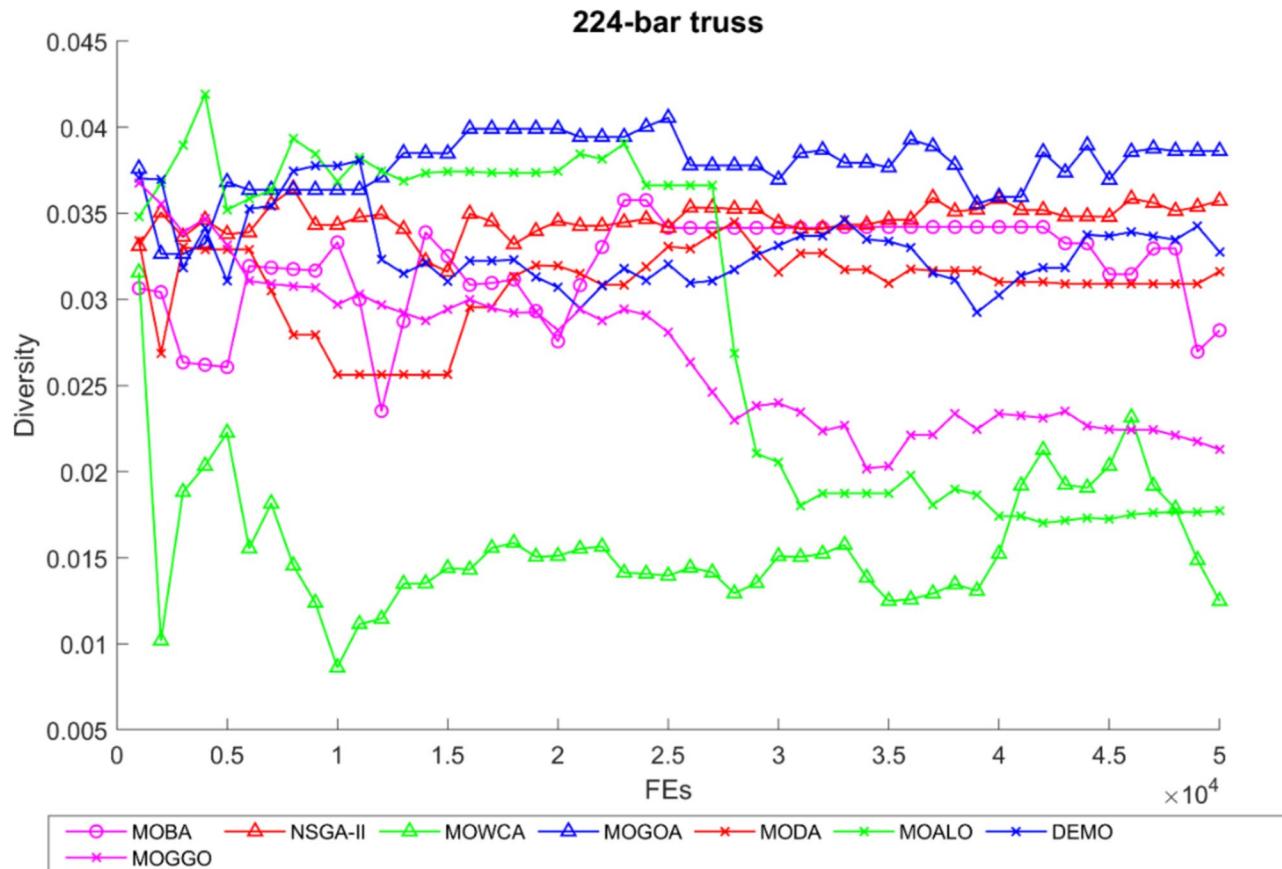


Fig. 26 The diversity curve for 224-bar truss

statistical validation put MOGGO on par with other algorithms and suggested the algorithm's capability to handle multi-objective optimization issues in structural engineering. Using established performance metrics and Friedman's rank test, we evaluated MOGGO's performance on six distinct truss structures with different shapes, sizes, and topologies (TSS). The results suggest that MOGGO performed well, ranking highly in several tests. However, while the algorithm showed promise regarding coverage, convergence, and solution diversification, its overall effectiveness should be further investigated, especially in more complex, real-world applications. Future research could explore hybridizing MOGGO with other optimization methods and testing its applicability in larger-scale, dynamic problems to better understand its practical impact and potential limitations in engineering optimization.

Author contributions N. M.: Methodology, Formal analysis, Data Curation, Writing—Original Draft, Writing—Review & Editing G.T.: Conceptualization, Methodology, Software, Formal analysis, Data Curation, Writing—Original Draft, Writing—Review & Editing, Supervision, Project administration P. P., Methodology, Formal

analysis, Data Curation, Writing—Original Draft, Writing—Review & Editing.

Funding No funding.

Data availability The datasets used and analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Tomar V, Bansal M, Singh P (2023) Metaheuristic algorithms for optimization: a brief review. Eng Proceed. <https://doi.org/10.3390/engproc2023059238>
2. Cui EH, Zhang Z, Chen CJ, Wong WK (2024) Applications of nature-inspired metaheuristic algorithms for tackling optimization problems across disciplines. Sci Rep 14(1):9403. <https://doi.org/10.1038/s41598-024-56670-6>
3. Kaveh A, Talatahari S, Khodadadi N (2022) Stochastic paint optimizer: theory and application in civil engineering. Eng Comput 38(3):1921–1952. <https://doi.org/10.1007/s00366-020-01179-5>

4. Oyelade ON, Ezugwu AE-S, Mohamed TIA, Abualigah L (2022) Ebola optimization search algorithm: a new nature-inspired metaheuristic optimization algorithm. *IEEE Access* 10:16150–16177. <https://doi.org/10.1109/ACCESS.2022.3147821>
5. Hubálovská M, Hubálovský Š, Trojovský P (2024) Botox optimization algorithm: a new human-based metaheuristic algorithm for solving optimization problems. *Biomimetics* 9(3):137. <https://doi.org/10.3390/biomimetics9030137>
6. El-kenawy ESM, Khodadadi N, Mirjalili S, Abdelhamid AA, Eid MM, Ibrahim A (2024) Greylag goose optimization: nature-inspired optimization algorithm. *Expert Syst Appl.* <https://doi.org/10.1016/j.eswa.2023.122147>
7. Azizi M, Baghazadeh Shishehgarkhaneh M, Basiri M, Moehler RC (2023) Squid game optimizer (SGO): a novel metaheuristic algorithm. *Sci Rep* 13(1):5373. <https://doi.org/10.1038/s41598-023-32465-z>
8. Meraihi Y, Gabis AB, Mirjalili S, Ramdane-Cherif A (2021) Grasshopper optimization algorithm: theory, variants, and applications. *IEEE Access* 9:50001–50024. <https://doi.org/10.1109/ACCESS.2021.3067597>
9. Kumar S, Tejani GG, Mehta P, Sait SM, Yildiz AR, Mirjalili S (2024) Optimization of truss structures using multi-objective cheetah optimizer. *Mech Based Des Struct Mach* 1:1–22. <https://doi.org/10.1080/15397734.2024.2389109>
10. Hashim FA, Mostafa RR, Hussien AG, Mirjalili S, Sallam KM (2023) Fick's law algorithm: a physical law-based algorithm for numerical optimization. *Knowl Based Syst* 260:110146. <https://doi.org/10.1016/j.knosys.2022.110146>
11. Rezaei F, Safavi HR, Abd Elaziz M, Mirjalili S (2023) GMO: geometric mean optimizer for solving engineering problems. *Soft comput* 27(15):10571–10606. <https://doi.org/10.1007/s00500-023-08202-z>
12. Abdollahzadeh B et al (2024) Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning. *Cluster Comput.* <https://doi.org/10.1007/s10586-023-04221-5>
13. Zhao W et al (2024) Electric eel foraging optimization: a new bio-inspired optimizer for engineering applications. *Expert Syst Appl* 238:122200. <https://doi.org/10.1016/j.eswa.2023.122200>
14. Amiri MH, Mehrabi Hashjin N, Montazeri M, Mirjalili S, Khodadadi N (2024) Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm. *Sci Rep* 14(1):5032. <https://doi.org/10.1038/s41598-024-54910-3>
15. Alzoubi S, Abualigah L, Sharaf M, Daoud MSh, Khodadadi N, Jia H (2023) Synergistic swarm optimization algorithm. *Comput Model Eng Sci* 0:1–10. <https://doi.org/10.32604/cmes.2023.045170>
16. Trojovska E, Dehghani M, Trojovsky P (2022) Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access* 10:49445–49473. <https://doi.org/10.1109/ACCESS.2022.3172789>
17. Zhao W, Wang L, Zhang Z, Mirjalili S, Khodadadi N, Ge Q (2023) Quadratic interpolation optimization (QIO): a new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems. *Comput Methods Appl Mech Eng.* <https://doi.org/10.1016/j.cma.2023.116446>
18. Khodadadi N, El-Kenawy E-SM, De Caso F, Alharbi AH, Khafaga DS, Nanni A (2023) The mountain gazelle optimizer for truss structures optimization. *Appl Comput Intell* 3(2):116–144. <https://doi.org/10.3934/aci.2023007>
19. Abdel-Basset M, Mohamed R, Zidan M, Jameel M, Abouhawwash M (2023) Mantis search algorithm: a novel bio-inspired algorithm for global optimization and engineering design problems. *Comput Methods Appl Mech Eng.* <https://doi.org/10.1016/j.cma.2023.116200>
20. Jia H, Rao H, Wen C, Mirjalili S (2023) Crayfish optimization algorithm. *Artif Intell Rev* 56:1919–1979. <https://doi.org/10.1007/s10462-023-10567-4>
21. Abdel-Basset M, Mohamed R, Azeem SAA, Jameel M, Abouhawwash M (2023) Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowl Based Syst.* <https://doi.org/10.1016/j.knosys.2023.110454>
22. Ouyang H, Chen J, Li S, Xiang J, Zhan Z-H (2023) Altruistic population algorithm: a metaheuristic search algorithm for solving multimodal multi-objective optimization problems. *Math Comput Simul* 210:296–319. <https://doi.org/10.1016/j.matcom.2023.03.004>
23. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization
24. Sadollah A, Eskandar H, Bahreininejad A, Kim JH (2015) Water cycle algorithm for solving multi-objective optimization problems. *Soft comput* 19(9):2587–2603. <https://doi.org/10.1007/s00500-014-1424-4>
25. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarrah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. *Appl Intell* 48(4):805–820. <https://doi.org/10.1007/s10489-017-1019-8>
26. Robič T, Filipič B (2005) DEMO: differential evolution for multi-objective optimization. In: Zitzler E, Coello Coello CA, Hernandez Aguirre A (eds) Evolutionary multi-criterion optimization. Springer Berlin Heidelberg, Berlin, pp 520–533
27. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27(4):1053–1073. <https://doi.org/10.1007/s00521-015-1920-1>
28. Yang X-S (2011) Bat algorithm for multi-objective optimisation. *Int J Bio-Inspired Comput* 3(5):267–274. <https://doi.org/10.1504/IJIBC.2011.042259>
29. Khodadadi N, Soleimanian Gharehchopogh F, Mirjalili S (2022) MOAVOA: a new multi-objective artificial vultures optimization algorithm. *Neural Comput Appl* 34(23):20791–20829. <https://doi.org/10.1007/s00521-022-07557-y>
30. Pereira JLJ, Oliver GA, Francisco MB, Cunha SS Jr, Gomes GF (2022) Multi-objective lichtenberg algorithm: a hybrid physics-based meta-heuristic for solving engineering problems. *Expert Syst Appl* 187:115939. <https://doi.org/10.1016/j.eswa.2021.115939>
31. Kumar S, Tejani GG, Pholdee N, Bureerat S (2021) Multi-objective passing vehicle search algorithm for structure optimization. *Expert Syst Appl.* <https://doi.org/10.1016/j.eswa.2020.114511>
32. Tejani GG, Kumar S, Gandomi AH (2021) Multi-objective heat transfer search algorithm for truss optimization. *Eng Comput* 37(1):641–662. <https://doi.org/10.1007/s00366-019-00846-6>
33. Azizi M, Talatahari S, Khodadadi N, Sareh P (2022) Multiobjective atomic orbital search (MOAOS) for global and engineering design optimization. *IEEE Access* 10:67727–67746. <https://doi.org/10.1109/ACCESS.2022.3186696>
34. Nouhi B, Khodadadi N, Azizi M, Talatahari S, Gandomi AH (2022) Multi-objective material generation algorithm (MOMGA) for optimization purposes. *IEEE Access* 10:107095–107115. <https://doi.org/10.1109/ACCESS.2022.3211529>
35. Khodadadi N, Azizi M, Talatahari S, Sareh P (2021) Multi-objective crystal structure algorithm (MOCryStAl): introduction and performance evaluation. *IEEE Access* 9:117795–117812. <https://doi.org/10.1109/ACCESS.2021.3106487>
36. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II
37. Mashruid N, Tejaniid GG, Patelid P, Khisheid M (2024) Optimal truss design with MOHO: a multi-objective optimization perspective. *PLoS ONE.* <https://doi.org/10.1371/journal.pone.0308474>

38. Mashru N, Tejani GG, Patel P (2024) Many-objective optimization of a 120-bar 3D dome truss structure using three metaheuristics. In: Taler J, Venkat Rao R (eds) Advanced engineering optimization through intelligent techniques. Springer Nature, Singapore, pp 231–239
39. Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. *Expert Syst Appl* 90:484–500. <https://doi.org/10.1016/j.eswa.2017.07.043>
40. Kumar S, Jangir P, Tejani GG, Premkumar M (2022) A decomposition based multi-objective heat transfer search algorithm for structure optimization. *Knowl Based Syst*. <https://doi.org/10.1016/j.knosys.2022.109591>
41. Mashru N, Patel P, Tejani GG, Kaneria A (2023) Multi-objective thermal exchange optimization for truss structure. In: Advanced engineering optimization through intelligent techniques: select proceedings of AEOTIT 2022, Springer, pp. 139–146
42. Kumar S et al (2023) A two-archive multi-objective multi-versatile optimizer for truss design. *Knowl Based Syst*. <https://doi.org/10.1016/j.knosys.2023.110529>
43. Tejani GG, Pholdee N, Bureerat S, Prayogo D (2018) Multiobjective adaptive symbiotic organisms search for truss optimization problems. *Knowl Based Syst* 161:398–414. <https://doi.org/10.1016/j.knosys.2018.08.005>
44. Fan M, Chen J, Xie Z, Ouyang H, Li S, Gao L (2022) Improved multi-objective differential evolution algorithm based on a decomposition strategy for multi-objective optimization problems. *Sci Rep* 12(1):21176. <https://doi.org/10.1038/s41598-022-25440-7>
45. Techasen T, Wansasueb K, Panagant N, Pholdee N, Bureerat S (2019) Simultaneous topology, shape, and size optimization of trusses, taking account of uncertainties using multi-objective evolutionary algorithms. *Eng Comput* 35(2):721–740. <https://doi.org/10.1007/s00366-018-0629-z>
46. Panagant N, Bureerat S, Tai K (2019) A novel self-adaptive hybrid multi-objective meta-heuristic for reliability design of trusses with simultaneous topology, shape and sizing optimisation design variables. *Struct Multidiscip Optim* 60(5):1937–1955. <https://doi.org/10.1007/s00158-019-02302-x>
47. Yadong W, Quan S, Weixing S, Qiang W (2019) Improve multi-objective ant lion optimizer based on quasi-oppositional and levy fly. In: 2019 Chinese control and decision conference (CCDC), pp. 12–17. <https://doi.org/10.1109/CCDC.2019.8832365>.
48. Meng Z, Li G, Wang X, Sait SM, Yıldız AR (2021) A comparative study of metaheuristic algorithms for reliability-based design optimization problems. *Arch Comput Methods Eng* 28(3):1853–1869. <https://doi.org/10.1007/s11831-020-09443-z>
49. Greiner D, Hajela P (2012) Truss topology optimization for mass and reliability considerations—Co-evolutionary multiobjective formulations. *Struct Multidiscip Optim* 45(4):589–613. <https://doi.org/10.1007/s00158-011-0709-9>
50. Park S, Choi S, Sikorsky C, Stubbs N (2004) Efficient method for calculation of system reliability of a complex structure. *Int J Solids Struct* 41(18–19):5035–5050. <https://doi.org/10.1016/j.ijsolstr.2004.04.028>
51. Stocki R, Kolaneck K, Jendo S, Kleiber M (2001) Study on discrete optimization techniques in reliability-based optimization of truss structures. *Comput Struct* 79(22–25):2235–2247. [https://doi.org/10.1016/S0045-7949\(01\)00080-3](https://doi.org/10.1016/S0045-7949(01)00080-3)
52. Kaveh A, Zaerreza A (2022) A new framework for reliability-based design optimization using metaheuristic algorithms. *Structures* 38:1210–1225. <https://doi.org/10.1016/J.ISTRUCL.2022.02.069>
53. Chun J (2021) Reliability-based design optimization of structures using complex-step approximation with sensitivity analysis. *Appl Sci (Switzerland)*. <https://doi.org/10.3390/app11104708>
54. Ahmadi A, Tiruta-Barna L, Capitanescu F, Benetto E, Marvuglia A (2016) An archive-based multi-objective evolutionary algorithm with adaptive search space partitioning to deal with expensive optimization problems: application to process eco-design. *Comput Chem Eng* 87:95–110. <https://doi.org/10.1016/J.COMPCHEMENG.2015.12.008>
55. Zitzler E, Technische E, Zürich H (1999) Evolutionary algorithms for multiobjective optimization: methods and applications
56. Houssein EH, Ahmed MM, Elaziz MA, Ewees AA, Ghoniem RM (2021) Solving multi-objective problems using bird swarm algorithm. *IEEE Access* 9:36382–36398. <https://doi.org/10.1109/ACCESS.2021.3063218>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.