Erciyes Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 41, Sayı 2, 2025

Erciyes University
Journal of Institue Of Science and Technology
Volume 41, Issue 2, 2025

Kafes Yapıların NSGA-II ve SHAMODE Algoritmaları ile Çok-amaçlı Optimizasyonu

Ibrahim Behram UGUR[*1]

[*1] Şırnak Üniversitesi Mühendislik Fakültesi İnşaat Mühendisliği, ŞIRNAK

(Alınış / Received: 20.04.2025, Kabul / Accepted: 02.06.2025, Online Yayınlanma / Published Online: 30.08.2025)

Anahtar Kelimeler
SHAMODE,
NSGA-II,
Çok-amaçlı optimizasyon
Pareto çözüm
Kafes yapı

Öz: Bu çalışmada, büyük ölçekli çok-amaçlı kafes yapı optimizasyonunda Başarı Geçmişine Dayalı Uyarlamalı Çok Amaçlı Diferansiyel Evrim (SHAMODE) ve İkinci Nesil Sıralamalı Genetik Algoritma (NSGA-II) yöntemlerinin performansları incelenmiştir. Amaç, kafes sistemin yapısal ağırlığını ve maksimum düğüm noktası deplasmanlarını minimize ederken, gerilme ve deplasman sınırlayıcılarını da sağlamaktır. Bu iki yöntem ile yapılan optimizasyon sonucunda elde edilen Pareto çözümlerin kalitesi ve dağılımı, Hiperhacim (HV), Nesilsel Uzaklık (GD), Ters Nesilsel Uzaklık (IGD) ve Aralık-Uzunluk Oranı (STE) performans ölçütleri kullanılarak değerlendirilmiştir. Farklı çalıştırmalar sonucu elde edilen en iyi ve ortalama değerler incelendiğinde, SHAMODE'un, NSGA-II'ye kıyasla daha yüksek HV ve daha düşük GD ve IGD değerleri ürettiği görülmüştür. Ayrıca, SHAMODE daha düşük STE değeri ile daha dengeli bir çözüm dağılımı sağlamıştır. Bu sonuçlar, SHAMODE'un karmaşık yapısal optimizasyon problemleri için etkili ve sağlam bir yöntem olduğunu ortaya koymaktadır.

Multi-Objective Optimization of Truss Structures Using NSGA-II and SHAMODE Algorithms

Keywords
SHAMODE
NSGA-II
Multi-objective optimization
Pareto front
Truss

Abstract: This study investigates the performance of Success-History Adaptive Multi-Objective Differential Evolution (SHAMODE) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) methods in solving a large-scale, multi-objective truss optimization problem. The objective is to minimize both the structural weight and the maximum nodal displacement, subject to stress and displacement constraints. Four widely used performance metrics including Hypervolume, Generational Distance (GD), Inverted Generational Distance (IGD), and Spacing-to-Extent (STE) are employed to evaluate the quality and distribution of the Pareto fronts obtained. Results from multiple independent runs show that SHAMODE consistently produces superior Pareto fronts, as evidenced by higher HV values and significantly lower GD and IGD scores compared to NSGA-II. Furthermore, SHAMODE achieves a more uniform distribution of solutions, as indicated by its lower STE values. These findings demonstrate SHAMODE's effectiveness and robustness in handling complex structural optimization problems

*İlgili Yazar, email: behramugur@sirnak.edu.tr

1. Introduction

Truss optimization has emerged as a dynamic and extensively studied area within structural engineering research, characterized by a rich variety of problem formulations and advanced algorithmic developments. Over recent decades, interest in truss optimization has grown significantly, driven by the dual goals of structural efficiency and economic feasibility. The types of optimization problems encountered in truss design are commonly classified as single-objective, multi-objective, and many-objective, with each category representing a progressively higher level of complexity and closer alignment with real-world engineering requirements. While single-objective formulations primarily focus on traditional targets such as minimizing structural mass or compliance, real-world design often necessitates balancing several conflicting performance criteria.

An extensive range of metaheuristic algorithms has been proposed for solving multi-objective optimization problems, particularly in structural engineering applications. Among the most established approaches are the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [1], the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [2], and the Pareto Archived Evolution Strategy (PAES) [3], all of which have been widely adopted for their ability to generate diverse and well-converged Pareto fronts. More recent developments have introduced nature-inspired methods such as the Multi-Objective Dragonfly Algorithm (MODA) [4], the Multi-Objective Grasshopper Optimization Algorithm (MOGOA) [5], and the Multi-Objective Salp Swarm Algorithm (MSSA) [6], which leverage biological behaviors to balance exploration and exploitation. Additionally, advanced population-based strategies such as the Unrestricted Population-Size Evolutionary Multi-Objective Optimization Algorithm (UPS-EMOA) [7], the Multi-Objective Multi-Verse Optimizer (MOMVO) [8], have shown promising results in handling high-dimensional and constrained design problems. Recent efforts have also focused on hybrid and adaptive algorithms, including the Success History-based Adaptive Multi-Objective Differential Evolution (SHAMODE) and its extension incorporating Whale Optimization (SHAMODE-WO) [9], as well as the Multi-Objective Meta-Heuristic with Iterative Parameter Distribution Estimation (MMIPDE) [10].

Multi-objective structural optimization of trusses commonly focuses on weight and displacement, but additional objectives are increasingly examined. Noilublao and Bureerat [11] integrated topology, shape, and sizing, assessing mass, compliance, frequencies, and transmissibility. Kaveh and Laknejadi [12] introduced a graph-based representation for truss layouts, incorporating specialized genetic operators. Tejani et al. [13] proposed a Multi-Objective Adaptive Symbiotic Organisms Search (MOASOS) with a two-archive method to balance exploration and exploitation. Mokarram and Banan [14] developed Fast Convergent Multi-Objective Particle Swarm Optimization (FC-MOPSO), enhancing leader selection for diversity and rapid convergence. Techasen et al. [15] considered reliability-based design by minimizing structural mass and reliability cost through multiple evolutionary algorithms. Vargas et al. [16] employed Generalized Differential Evolution 3 (GDE3) with an Adaptive Penalty Method, outperforming standard GDE3 and Non-dominated Sorting Genetic Algorithm II (NSGA-II). Kaveh and Mahdavi [17] extended Colliding Bodies Optimization into Multi-Objective Colliding Bodies Optimization (MOCBO), demonstrating efficient exploration and ranking in multi-objective spaces. Panagant et al. [18] compared fourteen metaheuristics, including NSGA-II and the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), revealing distinct performance strengths. Carvalio et al. [19] tested Differential Evolution variants on complex objectives like global stability and natural frequencies, incorporating topology and shape variables. Lemonge et al. [20] modified GDE3 to handle novel combinations of objectives, including critical load factors. Eid et al. [21] introduced the Multi-Objective Spiral Water Cycle Algorithm (MOSWCA), inspired by water cycle dynamics, integrating a hyperbolic spiral movement for stronger exploitation. Kumar et al. [22] developed the Multi-Objective Multi-Verse Optimizer with a Two-Archive Strategy (MOMVO2arc), leveraging dual-archive maintenance of diversity and convergence, surpassing NSGA-II and MOEA/D in benchmark truss problem. In this study, a three-dimensional steel truss structure comprising 264 members, originally proposed for the ISCSO-2024 [23] competition, is investigated as a benchmark for multi-objective optimization. The problem formulation involves two conflicting objectives: minimizing the structural weight and minimizing the maximum nodal displacement. With 264 design variables, the problem presents a high-dimensional and computationally challenging optimization task. To address this, two prominent evolutionary algorithms, NSGA-II and SHAMODE, are employed for generating Pareto-optimal solutions.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of the multi-objective optimization algorithms employed in this study. Section 3 presents the formulation of the multi-objective truss optimization problem. Section 4 outlines the details of the numerical experiment. In Section 5, the results are presented, including Pareto fronts, hypervolume values, and spacing metrics used for performance evaluation. Finally, Section 6 concludes the study.

2. Optimization algorithms

2.1. Non-dominated Sorting Genetic Algorithm – II (NSGA-II)

NSGA-II is a popular evolutionary algorithm for multi-objective optimization, known for its efficient non-dominated sorting approach and elitist selection mechanism. NSGA-II operates on a population of candidate solutions and evolves this population over a number of generations to approximate the Pareto-optimal front. The algorithm was originally proposed by Deb et al. [1] to address limitations of earlier multi-objective GAs by introducing three key features: fast non-dominated sorting, crowding-distance diversity preservation, and elitism.

NSGA-II maintains a population of size np and employs genetic operators (selection, crossover, mutation) similarly to a standard GA. The novelty lies in how solutions are evaluated and selected. Each individual solution is evaluated on all objective functions. A non-dominated sorting procedure classifies the population into Pareto fronts F_1, F_2... such that F_1 is the set of non-dominated solutions, F_2 is the set of solutions only dominated by those in F_1 and so forth. Solutions in lower-index fronts are better in the Pareto sense (rank 1 is best). Within each front, NSGA-II uses a crowding distance metric to estimate density: for each solution, the crowding distance is the average distance to its two nearest neighbors in objective space [1]. This provides a measure of how "isolated" a point is on that front; a larger crowding distance means the solution lies in a sparsely populated region of the front, which is desirable for diversity.

NSGA-II's selection is two-fold: an environmental selection that implements elitism, and a mating selection for reproduction. After each generation's variation (crossover/mutation), NSGA-II employs elitist survival selection by combining the parent and offspring populations (of total size 2×*npop*) and then filtering back to npop individuals for the next generation [1]. This is done by sorting the combined 2×*npop* pool into Pareto fronts and accepting front by front until the new population is filled. If the last accepted front Fi does not fit entirely only the most widely spaced solutions in F_i (those with largest crowding distance) are chosen to fill the remaining slots. This ensures the population retains a diverse spread of solutions and that no dominated solution survives (elitism). For mating selection (to create offspring), NSGA-II uses a binary tournament operator where two candidates are chosen and the one with better rank is selected (or if ranks equal, the one with a larger crowding distance). This preference ensures that parents with higher Pareto rank (or diversity) are more likely to reproduce.

***Algorithm 1**. Pseudo-code for NSGA-II*

```
Initialize population P of size npop
Evaluate the objective values for all individuals in P
Set generation counter iter ← 0
while iter <itermax do
   // Selection: Binary tournament based on rank and crowding distance
   for i = 1 to npop do
      Select parent1 and parent2 using tournament selection
      Apply crossover and mutation to generate offspringi
   end for
   Store offspring in Q                          // offspring population
   Evaluate objective values for all individuals in Q
   Combine P (parents) and Q (offspring) → R ← P ∪ Q
   Perform non-dominated sorting on R to get fronts F1, F2, ..., Fi
   Initialize new population Pnew ← Ø
   for each front Fi in F1, F2, ..., Fk do
      if (|Pnew| + |Fi| ≤ N) then
         Pnew ← Pnew ∪ Fi
      else
         Sort Fi by descending crowding distance
         Add top (npop - | Pnew |) individuals from Fi to Pnew
         break
      end if
   end for
   Update P ← Pnew
   iter ← iter + 1
end while
Return the non-dominated solutions in the final population P
```

2.2. Success History–based Adaptive Multi-Objective Differential Evolution (SHAMODE)

SHAMODE is a modern hybrid evolutionary approach that extends Differential Evolution with mechanisms for multi-objective optimization and self-adaptive parameter control. It was firstly proposed by Panagant et al. [9], originally to solve a reliability-based truss design problem involving simultaneous topology, shape, and sizing optimization. SHAMODE merges the DE search strategy with the non-dominated sorting and elitism concepts from algorithms like NSGA-II. Moreover, it introduces a success-history adaptive scheme to tune its mutation parameters F and crossover rate CR on the fly, thereby enhancing its robustness across different problem landscapes. In SHAMODE, candidate solutions are represented as vectors of design variables. DE generates new candidate solutions (offspring) through the mechanism of mutation and crossover applied to parent solutions [24].

In SHAMODE, the mutation operation is formulated as an extension of the *current-to-pbest*/1 strategy, incorporating both population-based and archive-based exploration. For each individual $x_{i,G}$in the current generation G, the corresponding mutant vector $v_{i,G}$ is computed using the following relation:

$$v_{i,G} = x_{i,G} + F_{i,G} \cdot (x_{pbest} - x_{i,G}) + + F_{i,G} \cdot (x_{r1,G} - \tilde{x}_{r2,G}) \tag{1}$$

where i represents the solution number in the population, x_{pbest} is randomly selected from best performing solutions, $F_{i,G}$ is a mutation scaling factor typically set within the range [0, 1], $x_{r1,G}$ *and* $\tilde{x}_{r2,G}$ are randomly selected from the current population and from the union of the current population and external archive.
Following mutation, a trial vector is produced by combining the mutant vector v_k and the original parent vector x_k through a crossover process. Each component of the trial vector is selected either from v_k or from x_k, based on a predefined crossover probability *CR*. This can be expressed as:

$$u_{i,j} = \begin{cases} v_{i,j} & if\ rand_j \leq CR \quad or\ j = j_{rand} \\ x_{i,j} & otherwise \end{cases} \tag{2}$$

where *j* indexes the decision variables, $rand_j$ is a uniformly distributed random number in the interval [0, 1], and j_{rand} is a randomly selected index to ensure at least one component is inherited from the mutant vector. In single-objective DE, selection is typically one-to-one: the trial vector $u_{i,j}$ replaces the parent $x_{i,j}$ if it exhibits superior fitness. However, in multi-objective optimization, as implemented in SHAMODE, selection is not performed in a one-to-one manner. Instead, SHAMODE follows a Pareto-based environmental selection strategy similar to that of NSGA-II. The parent and offspring populations are merged, and the next generation is formed through non-dominated sorting and crowding distance-based selection. This global competition allows all offspring to be evaluated relative to all parents and each other.

A distinguishing feature of the SHAMODE algorithm is its ability to adapt the control parameters of Differential Evolution (DE), specifically the mutation factor F and the crossover rate CR, throughout the optimization process based on historical success information. This self-adaptive mechanism is inspired by the SHADE algorithm, originally introduced by Tanabe and Fukunaga [25] and aims to eliminate the need for manual parameter tuning. To implement this mechanism, SHAMODE maintains an external memory archive commonly denoted as A which stores the historical memory of scaling factors (M_F) and (M_{CR}) associated with successful offspring, i.e., individuals that have either replaced their parents or have been included in the next generation through Pareto-based selection.

At the end of each generation, SHAMODE identifies which trial solutions have survived (i.e., selected into the next population). The corresponding control parameters used to generate those individuals are then recorded in the memory archive. Over time, this success history is used to update the sampling strategy for the control parameters in the following manner:

$$F_i = randc_i(\mu_F, 0.1) \tag{3}$$

$$CR_i = randn_i(\mu_{CR}, 0.1) \tag{4}$$

$randc_i$ *and* $randn_i$ denote random values generated based on Cauchy and normal distributions with means (μ_F, μ_{CR}) and variances ($\sigma_F^2, \sigma_{CR}^2$). The mean values of μ_F and μ_{CR} for each individual are randomly selected from the memories M_F and M_{CR} respectively [9]

The values of F_i *and* CR_i adjusted dynamically using a success-history adaptation method, building on the approach introduced in SHADE. While SHADE relies on a weighted Lehmer mean, SHAMODE simplifies this by applying the standard Lehmer mean, better suited for multi-objective settings where all non-dominated solutions are treated equally. Full implementation details are available in the original study by Panagant et al. [9].

Algorithm 2. Pseudo-code for SHAMODE
Initialize population P of size npop
Initialize memory M_F and M_{CR} to store successful F and CR values
Set generation counter iter ← 0
while *iter<itermax*
  for *each individual x_i* ***in*** *population P* ***do***
    Select F_i and CR_i from memory using success-history adaptation

Generate mutant vector v_i
Generate trial vector u_i
end for
Evaluate all trial vectors in Q // offspring population
Combine P (parents) and Q (offspring) → R ← P ∪ Q
Perform non-dominated sorting on R into fronts F_1, F_2, ..., Fi
Initialize new population P_{new} ← Ø
for** each front F_i in F_1, F_2, ..., Fi **do
***if** (|P_{new}| + |Fi| ≤ npop) then*
P_{new} ← P_{new} ∪ Fi
else
Sort Fi by descending crowding distance
Add top (npop - |P_{new}|) individuals from F_i to P_{new}|)
break
end if
end for
Identify successful individuals from Q that entered P_{new}
Store their F_i and CR_i values in memory by updating M_F and M_{CR}
Update P ← P_{new}
Increment generation counter: iter← iter + 1
end while
***Return** the non-dominated solutions in final population P*

3. Multi-objective Truss Optimization Formulation

Truss optimization problem is formulated as a constrained, multi-objective optimization task in which the goal is to minimize the total weight of the structure and nodal displacement.

$$f_1(A) = W = \sum_{i=1}^{n} \rho \; A_i \, L_i \tag{5}$$

$$f_2(A) = \max|d_j^{(ld)}| \tag{6}$$

Here, n is the number of truss members, A_i is the cross-sectional area, L_i is the length, and ρ is the material density. $d_j^{(l)}$ is the displacement vector of node j under load case ld.
The problem can be expressed as given below:

$$Find \quad A = [A_1, A_2, \dots \; A_n] \qquad \text{to minimize } \{f_1(A), f_2(A)\} \quad subjected\ to\ \; g_j(A) \le 0 \qquad j=1,2,\dots,m$$

Here, $g_j(A)$denotes the constraint functions (e.g., axial stress, buckling, or displacement limits), m is the total number of constraints. The truss design must comply with the AISC-LRFD (1994) provisions for both strength and stability. For members under axial tension, the design axial force must not exceed the available tensile strength:

$$P_u \le \phi_t \cdot P_n \tag{7}$$

P_u *and* P_n are the ultimate and nominal tensile force, ϕ_t *is* the resistance factor for tension and taken as 0.9. P_n is calculated as $P_n = A \cdot F_y$, with F_y representing the yield stress of the steel.

For members under axial compression, the design axial force must not exceed the critical buckling force (P_{cr}):

$$P_u \le \phi_c \cdot P_{cr} \tag{8}$$

ϕ_c *is* the resistance factor for compression and taken as 0.85. P_{cr} is calculated as $P_{cr} = A \cdot F_{cr}$, with F_{cr} representing the critical buckling stress is found as given equation:

$$F_{cr} = \begin{Bmatrix} \left(0.658^{\lambda_c{}^2}\right) F_y & if\ \lambda_c \le 1.5 \\ \left[\dfrac{0.877}{\lambda_c{}^2}\right] F_y & if\ \lambda_c > 1.5 \end{Bmatrix} \tag{9}$$

The critical slenderness ratio denoted *as* λ_c defines the transition point between elastic and inelastic buckling behavior in compression members.

To evaluate the performance of the proposed optimization algorithms in generating high-quality Pareto fronts, four established multi-objective performance indicators are implemented: Hypervolume (HV), Generational Distance (GD), Inverted Generational Distance (IGD), and Spacing-to-Extent ratio (STE). HV is the area (or volume in many-objective problems) between a reference point and the obtained Pareto front. It visually corresponds to the space dominated by non-dominated solutions and is influenced by the location of the reference point, typically set using the worst objective values observed across all algorithms and runs. The HV is calculated as:

$$HV = \text{volume}\left(\bigcup_{i=1}^{|P|} V_i\right) \tag{10}$$

where V_i is the hypercube defined between the ith non-dominated solution and the reference point. A larger HV value indicates a better approximation of the true Pareto front. GD measures the average Euclidean distance from each solution in the obtained Pareto front to its nearest point on a true pareto front. IGD is a metric which measures the average distance from each point in the reference front to its nearest solution in the obtained front. Both GD and IGD require a reference front, which ideally corresponds to the true Pareto front of the optimization problem. However, since the true front is typically unknown, a reference front is approximated by aggregating all non-dominated solutions from multiple runs (*M1*) of all compared algorithms (*M2*). The combined pool is then filtered to retain only the non-dominated solutions, which are used as the reference front [18]. Smaller values of GD and IGD indicate better algorithm performance by reflecting closer convergence to the reference front and improved coverage of the Pareto-optimal solution space. These metrics are calculated as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{|P|} d_i^2}}{|P|} \tag{11}$$

$$IGD = \frac{\sqrt{\sum_{i=1}^{|P\prime|} d_i'^2}}{|P'|} \tag{12}$$

STE is a composite metric used to assess both the distribution uniformity and the range of the obtained solutions. It is computed as the ratio of Spacing (SP) to Extent (ET):

$$SP = \frac{1}{|P|-1}\sum_{i=1}^{|P|}\left(d_i - \bar{d}\right)^2 \tag{13}$$

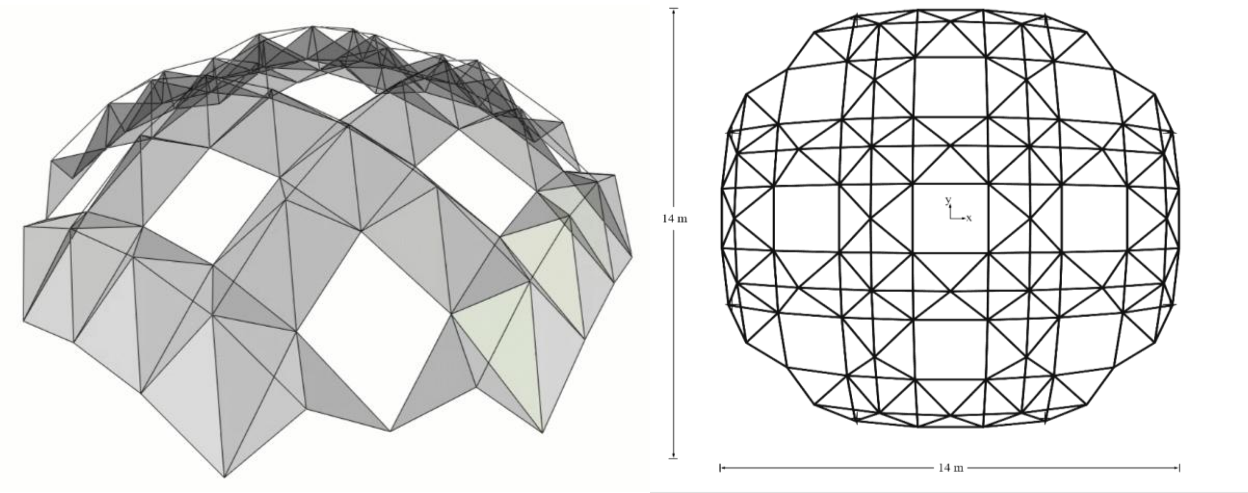
$$ET = \sqrt{\sum_{n=1}^{M} (f_n^{\max} - f_n^{\min})^2} \tag{14}$$

$$STE = \frac{SP}{ET} \tag{15}$$

where d_i is the Euclidean distance between the *i-th* solution and its nearest neighbor in the obtained front, $\bar{d}$ is the mean of all d_i, M is the number of objective functions, $f_n^{\max}$ *and* $f_n^{\min}$ are the maximum and minimum values of the n-th objective. A lower STE value indicates that the Pareto front is both uniformly spaced and well-extended, which is desirable for providing decision-makers with a diverse set of trade-off solutions.

4. Numeric Example: 264-Bar Dome-like Space Truss

In this study, the optimization problem of the International Student Competition in Structural Optimization (ISCSO 2024) [23] is considered as a benchmark example. The structure consists of a dome-like truss composed of 264 members and 88 nodes, of which 80 are free and 8 are designated as supports. The truss geometry is illustrated in Fig. 1. The material of the truss is steel with density ρ = 7.85 ton/m^3, Young's modulus E = 200 GPa, and yield strength Fy = 248.2 MPa.



(a) **(b)**

Figure 1. 268-bar dome-like truss geometry a) 3D view b) Top view [23]

A fixed topology is assumed, meaning that the nodal connectivity and geometry are predefined and remain unchanged throughout the optimization process. The structure is required to safely withstand multiple load cases, while two conflicting objectives are to be minimized: the total structural weight and the maximum nodal displacement [23].

This structure is subjected to three loading scenarios:

Load Case 1: 11 kN applied at each free node in the global *+x* direction
Load Case 2: 11 kN at each free node in the global *+y* direction
Load Case 3: 14 kN at each free node in the global *-z* direction

Since no member grouping is applied, each truss member is sized independently. As a result, there are 264 design variables, denoted as {*S1, S2, … S264*} each representing the cross-sectional section choice for one truss member. The cross-section selection is restricted to a predefined catalog of 37 commercially available circular hollow sections (Table 1) which are characterized by varying outer diameters and wall thicknesses. These are standard pipe profiles commonly used in steel structures. The variable *Si* is an *integer* between 1 and 37 indicating the section ID chosen for member-*i.* A value of 1 corresponds to the smallest pipe section and 37 to the largest. The total number of possible designs is 37^{264}, resulting in a vast and highly complex search space.

Table 1. Available Circular Hollow Sections [23]

ID	Section Name	Area (cm^2)	ID	Section Name	Area (cm^2)
1	PIPE 1/2" STD	1.6129	20	PIPE 3-1/2" XS	23.7419
2	PIPE 1/2" XS	2.0645	21	PIPE 2-1/2" XXS	25.9999
3	PIPE 3/4" STD	2.1484	22	PIPE 5" STD	27.7419
4	PIPE 3/4" XS	2.7935	23	PIPE 4" XS	28.4516
5	PIPE 1" STD	3.1871	24	PIPE 3" XXS	35.2903
6	PIPE 1" XS	4.1226	25	PIPE 6" STD	35.9999
7	PIPE 1-1/4" STD	4.3161	26	PIPE 5" XS	39.4193
8	PIPE 1-1/2" STD	5.1548	27	PIPE 4" XXS	52.2580
9	PIPE 1-1/4" XS	5.6839	28	PIPE 6" XS	54.1934
10	PIPE 1-1/2" XS	6.9032	29	PIPE 8" STD	54.1934
11	PIPE 2" STD	6.9032	30	PIPE 5" XXS	72.9031
12	PIPE 2" XS	9.5484	31	PIPE 10" STD	76.7740
13	PIPE 2-1/2" STD	10.9677	32	PIPE 8" XS	82.5805
14	PIPE 3" STD	14.3871	33	PIPE 12" STD	94.1934
15	PIPE 2-1/2" XS	14.5161	34	PIPE 6" XXS	100.6450
16	PIPE 2" XXS	17.1613	35	PIPE 10" XS	103.8708

17	PIPE 3-1/2" STD	17.2903	36	PIPE 12" XS	123.8707
18	PIPE 3" XS	19.4838	37	PIPE 8" XXS	137.4191
19	PIPE 4" STD	20.4516			

The optimization problem is formulated as a bi-objective design task with two objectives. The first objective is to minimize the total structural weight, aiming to reduce material usage and construction cost; however, this often results in a more flexible structure. The second objective is to minimize the maximum nodal displacement, promoting a stiffer structural response with smaller deflections under loading. These objectives inherently conflict, as achieving lower displacements typically requires the use of larger or heavier cross-sections, which in turn increases the total weight of the structure.

The optimization thus aims to find a Pareto front of designs ranging from very lightweight but flexible solutions to very stiff but heavy solutions. Each solution is evaluated by running the structural analysis provided by a MATLAB function in ICSCO-2024. Constraint handling is performed using a dominance-based approach, in which any solution that violates constraints is considered to be dominated by all feasible solutions. Throughout the optimization process, it is ensured that the final Pareto front comprises only non-dominated, feasible solutions. In practice, when comparing two solutions, the feasible one is always preferred over the infeasible one. Alternatively, infeasible solutions may be penalized by assigning artificially high objective function values. Both SHAMODE and NSGA-II are configured to prioritize feasibility, either by discarding infeasible candidates or by applying a constraint-domination rule during selection.

Since the true Pareto front of the optimization problem is unknown, a reference front was constructed following the approach described in Section 3. Specifically, all non-dominated solutions obtained from 30 independent runs of NSGA-II and SHAMODE were aggregated. Duplicate solutions were removed, and a final non-dominated sorting was applied to extract the reference front. This approximated front was then used as a common baseline to compute performance metrics such as GD and IGD.

5. Results and Discussion

The truss optimization problem was solved using both the NSGA-II and SHAMODE algorithms under identical experimental settings. For each algorithm, a population size of *npop*=80 was adopted, and the number of generations was specified to satisfy the evaluation budget of 10^6 function evaluations as conditioned in the competition [23]. Each algorithm was independently executed 30 times to ensure statistical reliability of the results. The initial solutions were constructed by randomly assigning section IDs to each member.

The NSGA-II algorithm was configured with standard settings as described in the original study Ref. [1], including binary tournament selection, a crossover probability of 0.9, and a mutation rate defined as $1/ndim$. SHAMODE was implemented according to its original formulation. Initial values of M_F *and* M_{CR} set to 0.5, and a historical memory size H=5 was used to guide the adaptive adjustment of control parameters as suggested in SHADE [25]. The maximum size of the external archive A was set to 1.4×*npop,* following the recommendation in L-SHADE [26]. It should be noted that the reference point used in the Hypervolume (HV) calculation was (*F1, F2)* = (57319, 10.5), as specified in Ref. [23]

Both algorithms employed the same constraint-handling strategy, in which any solution violating one or more constraints was considered inferior to all feasible solutions. This was enforced through either explicit removal of infeasible solutions or by applying a constraint-domination principle during selection. To assess robustness, each algorithm was run independently 30 times.

The HV results over 30 independent runs, including the mean, minimum, and standard deviation, are summarized in Table 2. As shown, SHAMODE achieved both a higher mean HV of 460 068.23 and a best HV of 451 693.92, outperforming NSGA-II, which yielded a best HV of 427 576.10 and a mean HV of 419 394.50. Given that higher HV reflects better front extension and quality, SHAMODE yielded approximately 7.6% higher hypervolume values compared to NSGA-II, indicating its stronger performance in generating well-distributed Pareto-optimal solutions for the multi-objective truss optimization problem. Additionally, the standard deviation of NSGA-II' is lower than that of SHAMODE indicating that NSGA-II produced more consistent results across independent runs, despite SHAMODE achieving higher average and better HV values.

Table 2. Hypervolume values from 30 independent runs.

Run no	SHAMODE	NSGA-II	Run no	SHAMODE	NSGA-II
1	453087.0	419404.9	16	452166.4	418808.5
2	447849.8	417153.0	17	441671.8	422215.2
3	458956.0	415173.6	18	459158.0	413460.7
4	446232.7	418293.5	19	460068.2	420869.1
5	450712.4	422778.8	20	449188.6	426207.9
6	455086.1	419189.7	21	452157.6	417076.3
7	454098.9	420856.1	22	458475.5	419432.9
8	454376.8	410742.5	23	448146.5	427576.1
9	452833.3	417231.7	24	443105.7	423951.3
10	443396.7	422455.9	25	453509.3	418885.3
11	451249.8	421705.4	26	449677.8	420469.3
12	446786.2	413480.4	27	459200.2	422722.0
13	450593.6	418117.9	28	444479.2	426849.8
14	447118.4	418545.4	29	457714.7	412677.4
15	454774.0	412798.2	30	454946.2	422707.3
Best HV	460068.2	427576.1			
Mean HV	451693.9	419394.5			
Std	5094.8	4171.8			

Pareto fronts obtained by NSGA-II and SHAMODE on the 264-bar truss design problem. HV plots with reference point and corresponding Pareto fronts obtained by NSGA-II and SHAMODE are presented in Figures 1 and 2, respectively. Each point corresponds to a feasible truss design, plotted by its total weight on horizontal axis and maximum nodal displacement on vertical axis. In the figures, the designs corresponding to the lowest weight with the highest displacement and the highest weight with the lowest displacement are highlighted within the same plot, using arrows pointing to the respective solutions on the Pareto front. Lower-left points are preferable, which denotes light and stiff designs. When comparing figures, it can be clearly seen that SHAMODE is able to extend the front slightly toward lighter weights and lower deflections, indicating a broader trade-off coverage.

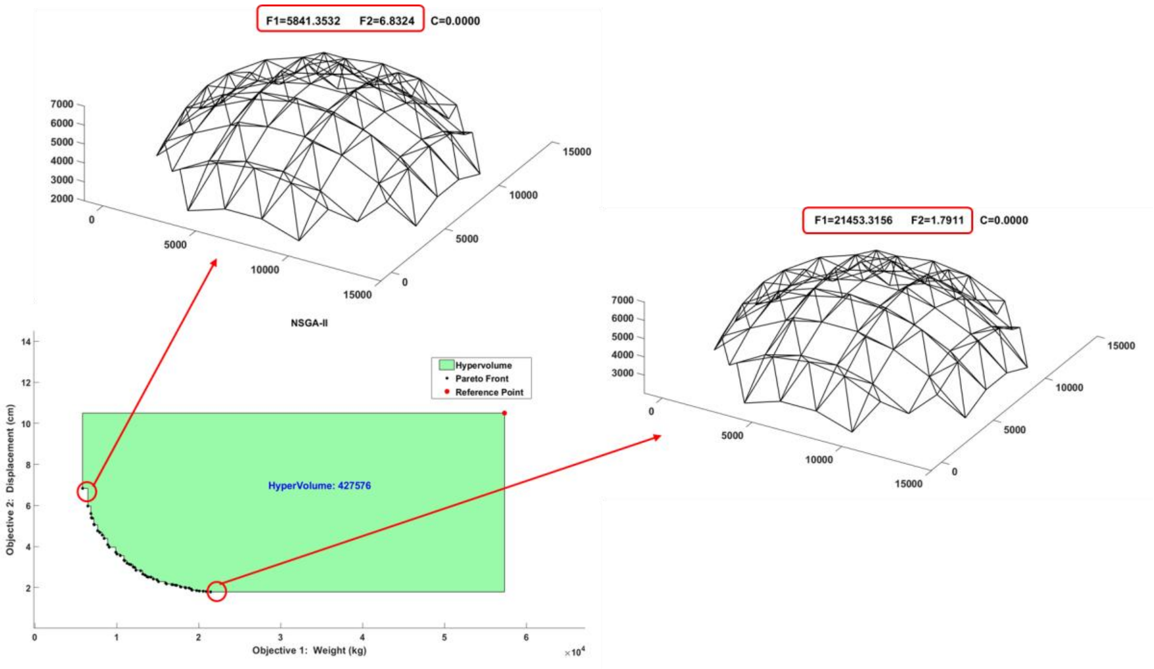


Figure 2. Hypervolume plot, best Pareto front, and representative structural designs obtained for the 264-bar truss using NSGA-II

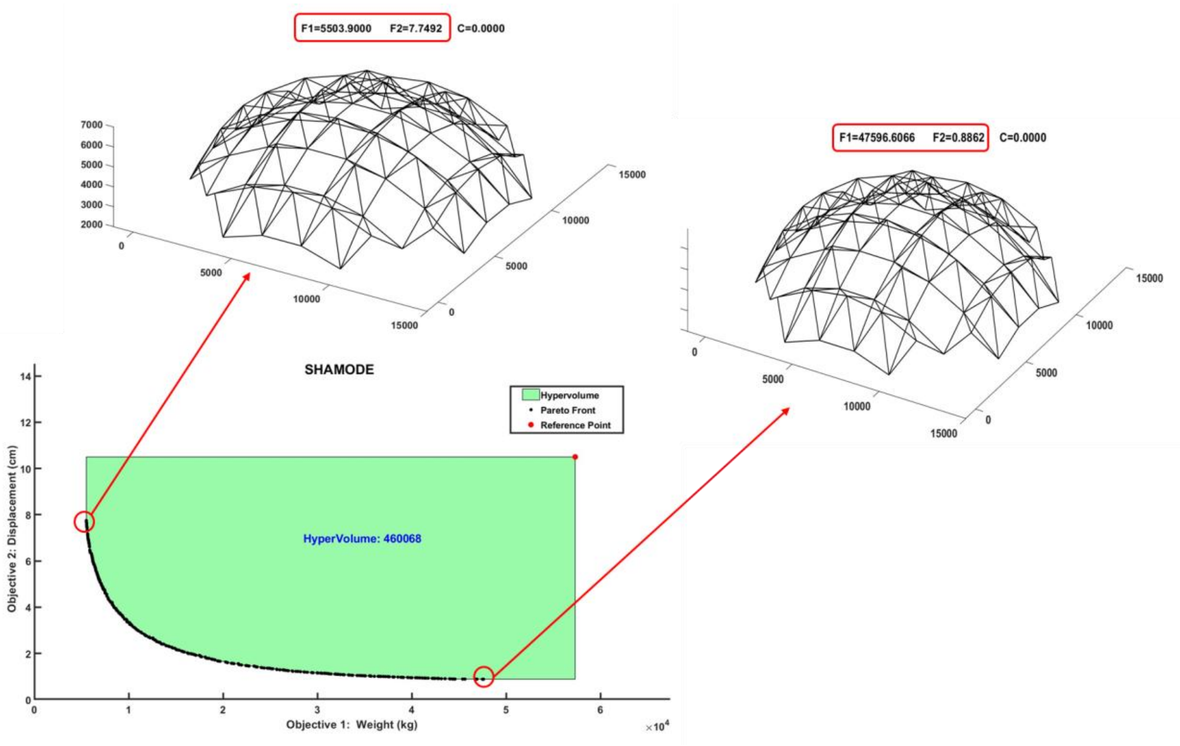


Figure 3. Hypervolume plot, best Pareto front, and representative structural designs obtained for the 264-bar truss using SHAMODE

Table 3 presents the performance metrics including GD, IGD, and STE for the SHAMODE and NSGA-II algorithms evaluated on the 264-bar truss optimization problem. In terms of convergence, SHAMODE outperforms NSGA-II, as reflected by its lower GD values (best: 2.15 vs. 3.68; mean: 3.53 vs. 6.32), indicating that SHAMODE's solutions are closer on average to the reference front. The IGD metric, which evaluates both convergence and coverage, further confirms SHAMODE's advantage. SHAMODE reports a significantly lower best IGD (11.80 vs. 478.29) and mean IGD (87.81 vs. 536.22), demonstrating a more comprehensive approximation of the reference front. Finally, the STE values which combine spacing and extent to assess uniformity show that SHAMODE provides a more uniformly distributed front (mean STE: 0.00158) compared to NSGA-II (mean STE: 0.00974). This suggests that SHAMODE's Pareto solutions are not only better in quality but also more evenly spaced. In summary, across all performance metrics, SHAMODE outperforms NSGA-II both in terms of solution quality and consistency, making it a more effective algorithm for the large-scale, multi-objective truss optimization problem considered, as shown in Figure 4.

Table 3. Performance Metrics for NSGA-II vs SHAMODE on the 264-bar Truss Problem

Algorithm		HV	GD	IGD	STE
SHAMODE	*Best*	460068.2	2.154979	11.79982	0.000493
	Mean	451693.9	3.529307	87.80671	0.001583
	Std	5094.797	0.862568	43.77879	0.000824
NSGA-II	*Best*	427576.1	3.675992	478.2892	0.006407
	Mean	419394.5	6.317392	536.2202	0.009739
	Std	4171.81	1.62877	32.1734	0.002277

To statistically assess the significance of the observed performance differences in HV values, a Wilcoxon signed-rank test was conducted using the results from 30 independent runs. The test yielded a p-value of 3.20×10^{-11}, indicating that the differences between SHAMODE and NSGA-II are statistically significant at the 0.05 level. These results confirm that SHAMODE consistently outperforms NSGA-II in terms of HV metric across repeated trials.

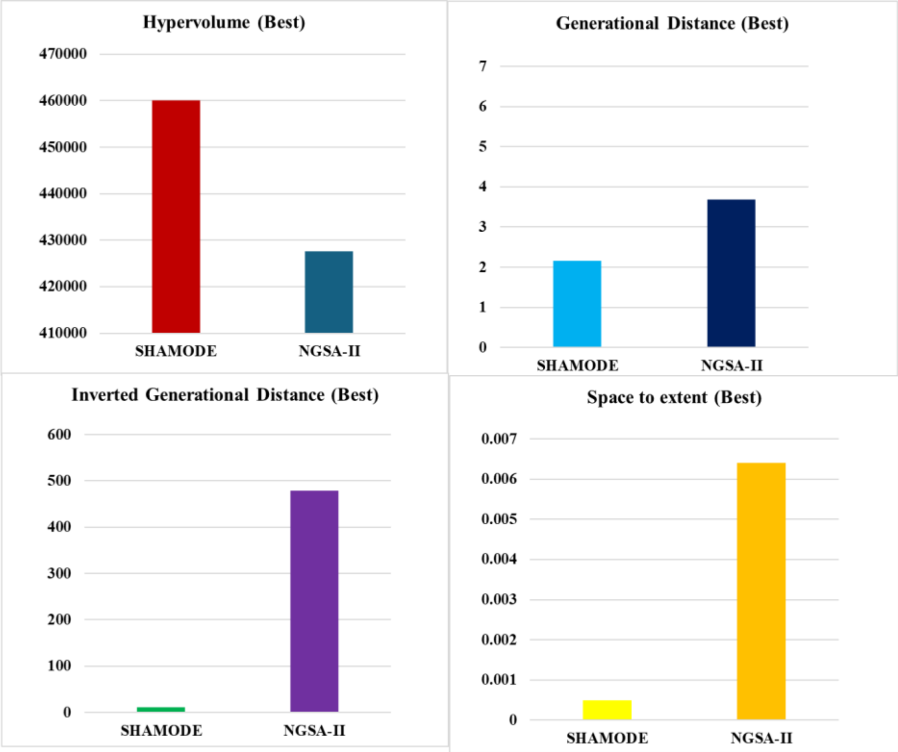


Figure 4. Comparison of NSGA-II and SHAMODE Using Best Performance Metric Values

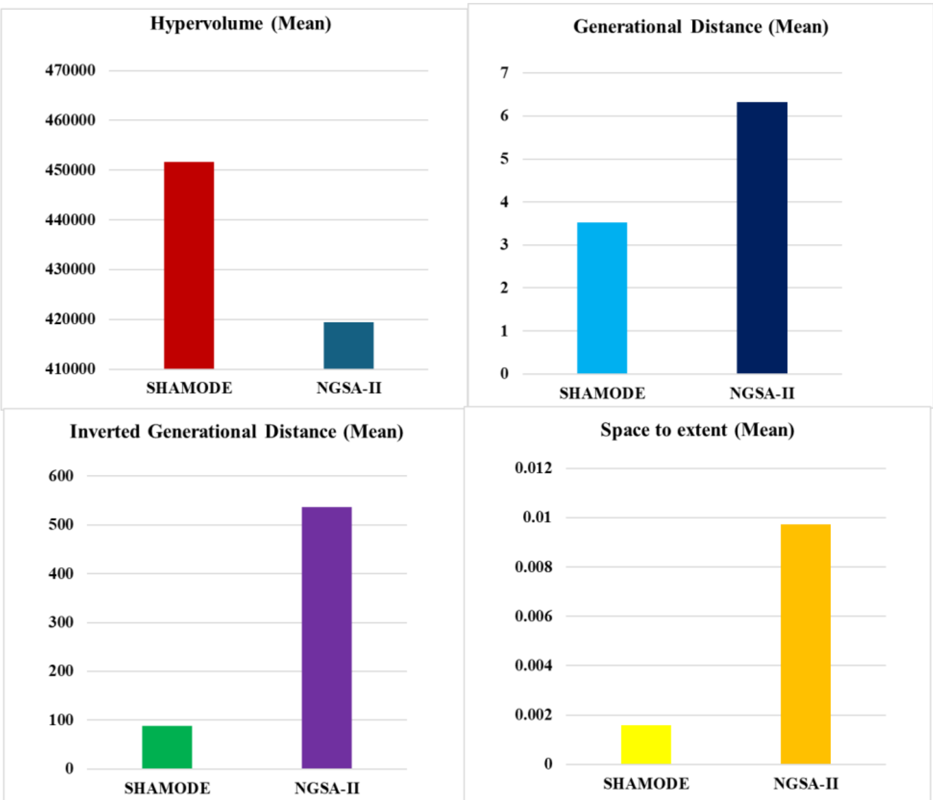


Figure 5. Comparison of NSGA-II and SHAMODE Using Mean Performance Metric Values

Figure 6 presents the HV convergence curves for SHAMODE and NSGA-II. Both algorithms exhibit a rapid initial increase in HV during the early iterations, which is typical as they quickly approximate the Pareto front. However, SHAMODE continues to improve steadily throughout the optimization, while NSGA-II's progress begins to plateau after approximately 4,000 iterations. This suggests that SHAMODE maintains a more effective balance between exploration and exploitation over time, enabling it to find better-distributed and more convergent solutions.

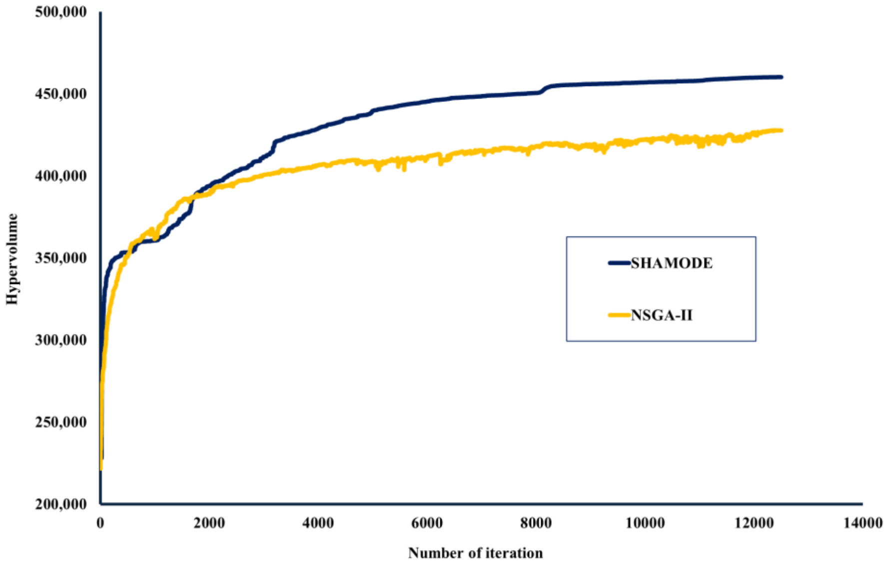


Figure 6. Hypervolume convergence trends of SHAMODE and NSGA-II

In addition to evaluating solution quality, the computational effort required by each algorithm was assessed. The average computation time over 30 independent runs was 6521.81 seconds for SHAMODE and 5507.12 seconds for NSGA-II. As expected, SHAMODE required more time due to its more complex structure, which includes adaptive control parameters and historical memory mechanisms. While this increases the computational burden, it contributes to the algorithm's improved search performance and robustness in generating high-quality Pareto fronts. A boxplot comparing the computation times of both algorithms is presented in Figure 7 to visually support this observation.

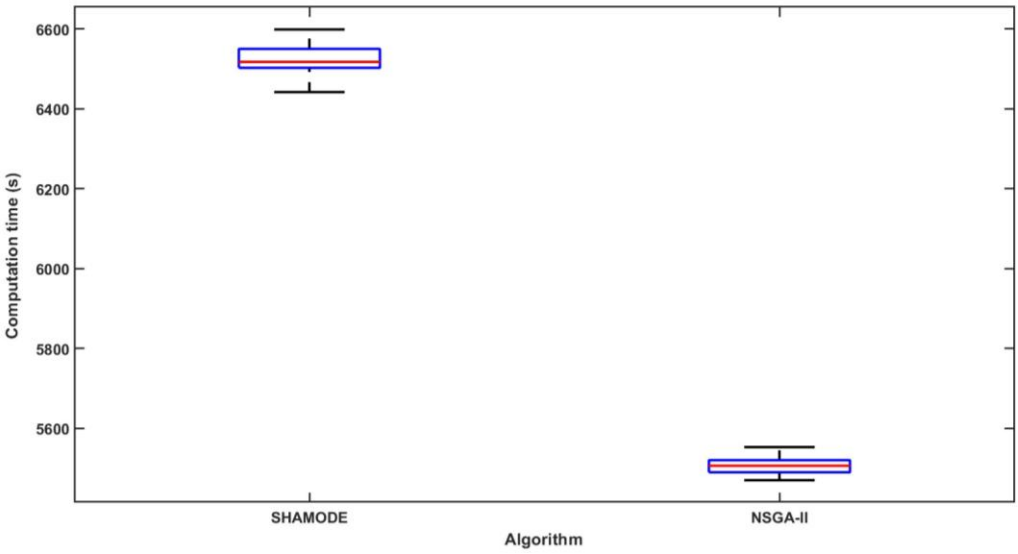


Figure 7. Comparison of computation times for SHAMODE and NSGA-II over 30 independent runs.

6. Conclusion

This study presented a comparative performance analysis of two evolutionary algorithms, SHAMODE and NSGA-II, for solving large-scale multi-objective truss optimization problems. Performance was evaluated using widely accepted metrics. Based on the 264-bar truss benchmark, SHAMODE consistently outperformed NSGA-II in terms of convergence, diversity, and distribution of Pareto-optimal solutions. SHAMODE achieved higher HV values, indicating better front extension, and demonstrated superior convergence behavior with lower GD and IGD scores. Furthermore, the algorithm provided more uniformly distributed solutions, as reflected by its favorable STE values. These findings were further supported by convergence trends and visualizations of Pareto fronts. Overall, the results highlight SHAMODE as a more effective and robust approach for addressing complex structural optimization problems involving multiple conflicting objectives. As future work, upgraded versions of the present algorithms could be developed and evaluated on more complex multi-objective truss optimization tasks.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, 2002, doi: 10.1109/4235.996017.

[2] F. Zhao, W. Lei, W. Ma, Y. Liu, and C. Zhang, "An improved SPEA2 algorithm with adaptive selection of evolutionary operators scheme for multiobjective optimization problems," *Mathematical Problems in Engineering*, vol. 2016. 2016. doi: 10.1155/2016/8010346.

[3] S. Rostami and F. Neri, "Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm," *Integr. Comput. Aided. Eng.*, vol. 23, no. 4, 2016, doi: 10.3233/ICA-160529.

[4] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, 2016, doi: 10.1007/s00521-015-1920-1.

[5] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," *Appl. Intell.*, vol. 48, no. 4, 2018, doi: 10.1007/s10489-017-1019-8.

[6] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[7] T. Aittokoski and K. Miettinen, "Efficient evolutionary approach to approximate the Pareto-optimal set in multiobjective optimization, UPS-EMOA," in *Optimization Methods and Software*, 2010, vol. 25, no. 6. doi: 10.1080/10556780903548265.

[8] S. Mirjalili, P. Jangir, S. Z. Mirjalili, S. Saremi, and I. N. Trivedi, "Optimization of problems with multiple objectives using the multi-verse optimization algorithm," *Knowledge-Based Syst.*, vol. 134, 2017, doi: 10.1016/j.knosys.2017.07.018.

[9] N. Panagant, S. Bureerat, and K. Tai, "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.*, vol. 60, no. 5, 2019, doi: 10.1007/s00158-019-02302-x.

[10] K. Wansasueb, N. Pholdee, N. Panagant, and S. Bureerat, "Multiobjective meta-heuristic with iterative parameter distribution estimation for aeroelastic design of an aircraft wing," *Eng. Comput.*, vol. 38, no. 1, 2022, doi: 10.1007/s00366-020-01077-w.

[11] N. Noilublao and S. Bureerat, "Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms," *Comput. Struct.*, vol. 89, no. 23–24, 2011, doi: 10.1016/j.compstruc.2011.08.010.

[12] A. Kaveh and K. Laknejadi, "A hybrid evolutionary graph-based multi-objective algorithm for layout optimization of truss structures," *Acta Mech.*, vol. 224, no. 2, 2013, doi: 10.1007/s00707-012-0754-5.

[13] G. G. Tejani, N. Pholdee, S. Bureerat, and D. Prayogo, "Multiobjective adaptive symbiotic organisms search for truss optimization problems," *Knowledge-Based Syst.*, vol. 161, 2018, doi: 10.1016/j.knosys.2018.08.005.

[14] V. Mokarram and M. R. Banan, "A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables," *Struct. Multidiscip. Optim.*, vol. 57, no. 2, 2018, doi: 10.1007/s00158-017-1764-7.

[15] T. Techasen, K. Wansasueb, N. Panagant, N. Pholdee, and S. Bureerat, "Simultaneous topology, shape, and size optimization of trusses, taking account of uncertainties using multi-objective evolutionary algorithms," *Eng. Comput.*, vol. 35, no. 2, 2019, doi: 10.1007/s00366-018-0629-z.

[16] D. E. C. Vargas, A. C. C. Lemonge, H. J. C. Barbosa, and H. S. Bernardino, "Differential evolution with the

adaptive penalty method for structural multi-objective optimization," *Optim. Eng.*, vol. 20, no. 1, 2019, doi: 10.1007/s11081-018-9395-4.

[17] A. Kaveh and V. R. Mahdavi, "Multi-objective colliding bodies optimization algorithm for design of trusses," *J. Comput. Des. Eng.*, vol. 6, no. 1, 2019, doi: 10.1016/j.jcde.2018.04.001.

[18] N. Panagant, N. Pholdee, S. Bureerat, A. R. Yildiz, and S. Mirjalili, "A Comparative Study of Recent Multi-objective Metaheuristics for Solving Constrained Truss Optimisation Problems," *Arch. Comput. Methods Eng.*, vol. 28, no. 5, 2021, doi: 10.1007/s11831-021-09531-8.

[19] J. P. G. Carvalho, É. C. R. Carvalho, D. E. C. Vargas, P. H. Hallak, B. S. L. P. Lima, and A. C. C. Lemonge, "Multi-objective optimum design of truss structures using differential evolution algorithms," *Comput. Struct.*, vol. 252, 2021, doi: 10.1016/j.compstruc.2021.106544.

[20] A. C. C. Lemonge, J. P. G. Carvalho, P. H. Hallak, and D. E. C. Vargas, "Multi-objective truss structural optimization considering natural frequencies of vibration and global stability," *Expert Syst. Appl.*, vol. 165, 2021, doi: 10.1016/j.eswa.2020.113777.

[21] H. F. Eid, L. Garcia-Hernandez, and A. Abraham, "Spiral water cycle algorithm for solving multi-objective optimization and truss optimization problems," *Eng. Comput.*, vol. 38, 2022, doi: 10.1007/s00366-020-01237-y.

[22] S. Kumar *et al.*, "A two-archive multi-objective multi-verse optimizer for truss design," *Knowledge-Based Syst.*, vol. 270, 2023, doi: 10.1016/j.knosys.2023.110529.

[23] ISCSO-2024," *International Student competition in Structural optimization*.

[24] J. M. P. Vieira, J. P. G. Carvalho, D. E. C. Vargas, É. C. R. Carvalho, P. H. Hallak, and A. C. C. Lemonge, "Many-Objective Truss Structural Optimization Considering Dynamic and Stability Behaviors," *Dynamics*, vol. 5, no. 1, p. 3, 2025.

[25] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for Differential Evolution," in *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2013. doi: 10.1109/CEC.2013.6557555.

[26] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, 2014. doi: 10.1109/CEC.2014.6900380