IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 17, 2021, accepted June 5, 2021, date of publication June 9, 2021, date of current version June 18, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3087739*

# MOPGO: A New Physics-Based Multi-Objective Plasma Generation Optimizer for Solving Structural Optimization Problems

**SUMIT KUMAR[1], PRADEEP JANGIR[2], GHANSHYAM G. TEJANI[3], MANOHARAN PREMKUMAR[4], (Member, IEEE), AND HASSAN HAES ALHELOU[5], (Senior Member, IEEE)**

[1]Australian Maritime College, College of Sciences and Engineering, University of Tasmania, Launceston, TAS 7248, Australia
[2]Rajasthan Rajya Vidyut Prasaran Nigam, Sikar 332025, India
[3]Department of Mechanical Engineering, School of Technology, GSFC University, Vadodara 391750, India
[4]Department of Electrical and Electronics Engineering, Dayananda Sagar College of Engineering, Bengaluru 560078, India
[5]Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia 2230, Syria

Corresponding authors: Hassan Haes Alhelou (alhelou@ieee.org) and Manoharan Premkumar (mprem.me@gmail.com)

**ABSTRACT** This paper proposes a new Multi-Objective Plasma Generation Optimization (MOPGO) algorithm, and its non-dominated sorting mechanism is investigated for numerous challenging real-world structural optimization design problems. The Plasma Generation Optimization (PGO) algorithm is a recently reported physics-based algorithm inspired by the generation process of plasma in which electron movement and its energy level are based on excitation modes, de-excitation, and ionization processes. As the search progresses, a better balance between exploration and exploitation has a more significant impact on the results; thus, the crowding distance feature is incorporated in the proposed MOPGO algorithm. Also, the proposed posteriori method exercises a non-dominated sorting strategy to preserve population diversity, which is a crucial problem in multi-objective meta-heuristic algorithms. In truss design problems, minimization of the truss's mass and maximization of nodal displacement are considered objective functions. In contrast, elemental stress and discrete cross-sectional areas are assumed to be behavior and side constraints, respectively. The usefulness of MOPGO to solve complex problems is validated by eight truss-bar design problems. The efficacy of MOPGO is evaluated based on ten performance metrics. The results demonstrate that the proposed MOPGO algorithm achieves the optimal solution with less computational complexity and has a better convergence, coverage, diversity, and spread. The Pareto fronts of MOPGO are compared and contrasted with multi-objective passing vehicle search algorithm, multi-objective slime mould algorithm, multi-objective symbiotic organisms search algorithm, and multi-objective ant lion optimization algorithm. This study will be further supported with external guidance at https://premkumarmanoharan.wixsite.com/mysite.

**INDEX TERMS** Constraints optimization problems, crowding distance, meta-heuristics, non-dominated sorting, numerical optimization, Pareto front, structure optimization.

## I. INTRODUCTION

Design problems in physics and technology are mostly linked to more than one objective requiring a trade-off between these competing objectives to achieve optimal solutions [1]. These are multi-objective (MO) design problems in which programmers try to balance cost and performance. Structure optimization is one of the most vital challenges associated with engineering designs due to many diverse objectives under multiple constraints. The structure total weight reduction combined with maximum deflection objective is perhaps the most studied optimization case in truss bar design problems [2,3]. Researchers established numerous methodologies to tackle such engineering problems; nevertheless, these studies often failed to find optimal solutions from the established Pareto-front [4]. An optimal design can be achieved by creating a fine balance among various diverse goals by properly addressing two undertakings. First, a standard optimization technique should be developed to achieve a high-efficiency solution by finding a middle ground between

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

various design objectives, while the second task is to choose the best choice out of all those given to you. As a result, the best option should come from the Pareto optimal points, and a decision-maker should choose it [5]. The MO optimization problems are highly complicated due to many and diverse objectives [6]. Besides single-objective problems, these design problems have more than one optimal solution, referred to as a Pareto-optimal set. Thus, MO design issues require a powerful optimization methodology to resolve these challenging problems. One of the very active areas of research in optimization nowadays is the field of meta-heuristics (MHs) because of their high flexibility and effectiveness in solving complex problems like discrete, discontinuous, noisy, dynamic, and non-differentiable. In the last four decades, many new MHs have been proposed and applied in many optimization tasks. Genetic Algorithm (GA) [7], Differential Evolution (DE) [8], the Particle Swarm Optimization (PSO) algorithm [9], and Ant Colonies (ACO)[10] are a few of the influential MHs that grasp more attention of scholars from diverse field. Moreover, apart from a single objective application, many MHs were also developed and implemented for MO design optimization problems. Few notable state-of-the-art techniques are: NSGA-II [11], SPEA2 [12], PESA-II [13], PAES [14], MO Heat Transfer Search (MOHTS) [15], MO cuckoo search [16], MOPSO [17], decomposition-based MO evolutionary algorithm [18], and MOGA [19]. While the outcomes generated by MHs are not necessarily the optimal results, they can be accomplished in a reasonable period.

Moreover, the potential to reach the Pareto front in a single run is the most striking feature of these MHs [20]–[23]. However, literature depicts the failure of these MHs while solving large-scale, complex, dynamic, and multi-constraints practical optimization problems. This is due to their slow convergence rate, local optima trap, a bunch of controlling parameters, the high time of computation, and incompetency in resolving non-trivial objective functions. These limitations were always tried to settle with further improvements and hybridizations of diverse techniques by many researchers. Few examples are improved MOHTS [24], adaptive MO Symbiotic Organisms Search (MOSOS) [25], Hybrid HTS and Passing Vehicle Search (PVS) [1], Grey Wolf Optimizer based on non-dominated sorting technique [26], enhanced chaotic JAYA algorithm [27], hybrid PSO-Multi Verse Optimizer [28], and many more. A fine balance between global diversification and local intensification is essential for MHs [4], [5]. In principle, the terminology diversification corresponds to search space exploration, while the expression intensification leads to the utilization of cumulative search knowledge. As mentioned, the harmony between diversification and intensification is crucial because the former helps in promptly identifying the high-quality solutions regions in the search arena. In contrast, the latter leads in minimal time in search areas that are either already being explored or that do not offer high-quality solutions [15], [22]. Today's quite burning question is the quest for even more powerful methods to efficiently solve the challenging and complex practical engineering design problems without compromising the local and global search rate. The emergence of new nature-inspired MHs is thus rising drastically, and many new algorithms were introduced and claimed to be efficient. Few prominent techniques are grey wolf [29], slime mould [30], marine predators [31], Jaya [32], harris hawks [33], equilibrium [34], sine cosine [35], ant lion [36], moth-flame [37], and chaotic gradient-based optimizer [38].

However, as per the prominent "No-Free-Lunch" theory [39], one MH cannot solve every problem effectively and efficiently. An MH may yield a good result in a specific design issue, but the same strategy might generate a feeble result in another challenge. There is no MH that provides optimal response for each problem, to put it another way. Hence, a search for a powerful and robust algorithm always prevails.

A recently proposed Plasma Generation Optimization (PGO) [40] has shown unique characteristics, including (i) no need for parameter adjustments since the optimizer is completely parameter-free; (ii) excellent capabilities for exploration with plasma ionization generation; (iii) exploitation capabilities gained by de-excitation phase; (iv) inferior solutions can be eliminated during the excitation phase. With these four benefits, it is clear to see how the PGO algorithm excels. Compared with other MHs, very few possess all four properties above, leading to more accurate results and reliable processes. These abilities and prospects encourage the authors to create a new non-dominated sorting-based (NDS) optimizer termed MOPGO, which is tested on various real-world structure optimization problems in this study. Many researchers have been drawn to the NDS-based paradigm because of its simplicity and suitability for MO design problems in the physical world [11], [26], [41], [42]. As a result, the current research combines the benefits of NDS methodology with PGO to produce a robust global optimization technique that balances local intensification and global diversification of search. Hence, the main contributions of this paper are as follows.

- A new NDS and Crowding Distance (CD) based MOPGO algorithm is proposed for solving MO problems.
- The framework of MOPGO lies on the foundation of PGO. Therefore, the MOPGO exploits the plasma generation phase to achieve equilibrium between search intensification and diversification.
- Demonstrates selecting one optimal solution from the Pareto fronts for a real-world application in practice through the fuzzy-based decision.
- Discusses three crucial aspects of the algorithm, i.e., optimization capability of the search algorithm, Pareto dominance, and solution diversity simultaneously.
- The MOPGO is evaluated using eight challenging real-world MO structural optimization design problems.
- Performance evaluation has been made qualitatively and quantitatively with MO Passing Vehicle Search

(MOPVS), MO Slime Mould Algorithm (MOSMA), MO Symbiotic Organisms Search (MOSOS), and MO Ant Lion Optimization (MOALO) algorithms.

The rest of the paper is structured as follows: Section 2 provides details of the basic PGO algorithm. Section 3 discusses the formulation procedure of NDS based MOPGO algorithm. Section 4 presents the mathematical concepts of the selected MO optimization problems. The performance analysis and detailed description for all truss bar problems are presented in Section 5. Section 6 concludes the paper based on metrics and obtained Pareto fronts.

## II. PLASMA GENERATION OPTIMIZATION (PGO) ALGORITHM

Literature reveals that MHs prove to be an effective and efficient strategy in dealing with many challenging real-world optimization problems. They can find suitable solutions through a random search methodology inspired by nature. MHs are widely utilized due to their potential to achieve an optimal solution in most instances in the least computational time. PGO is the newly introduced physics-based algorithm that imitates the industrial plasma generation procedure and demonstrates its competitiveness with other state-of-the-art techniques in solving several constrained benchmarks and real-world problems [40]. The main steps of the PGO algorithm can be stated as follows:

### A. STEP 1: INITIALIZATION

The light beam strikes the molecules at the beginning of the procedure, creating a population of n random candidate solutions from electrons of various energy levels in the search space:

$$e_{i,j}^{0} = e_{j,min} + rand \times \left(e_{j,max} - e_{j,min}\right) \quad i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, d \tag{1}$$

where $e_{i,j}^{0}$ is the initial value of the $j$th variable of the $i$th candidate solution; $e_{j,min}$ and $e_{j,max}$ are respectively the minimum and maximum permissible values for the $j$th variable; *rand* is a random number uniformly distributed in the interval [0, 1]; $n$ is the number of electrons and $d$ is the number of design variables.

### B. STEP 2: IDENTIFYING EACH ELECTRON's PHYSICAL PROCESS

To determine an electron's new location, a random number between 0 and 1 is produced first. The randomly generated number specifies which physical process (excitation/de-excitation or ionization) should occur for the electron, so that:

$$\begin{aligned} &if\ rand_1 < EDR,\ \mathrm{X} \\ &otherwise,\ \mathrm{Y} \end{aligned} \tag{2}$$

in which $rand_1$ is a random number generated between 0 and 1; *EDR* is an excitation/de-excitation rate, $X$ is an excitation and/or de-excitation processes are occurred, and $Y$ is an ionization process that has occurred.

### C. STEP 3: EXCITATION AND/OR DE-EXCITATION PROCESSES ARE USED TO GENERATE A NEW ELECTRON POSITION

#### 1) EXCITATION PROCESS

Positive charge (protons) and negative charge (antiprotons) are the two charged particles that make up an atom (electron). There are electrons in atomic orbitals with various energy levels around the nucleus, based on the action of electron waves. Some of the electrons that are closer to the nucleus are positioned at lower energy levels. Electron beams interfere with the atom during excitation, increasing the energy level of atomic electrons. The movement of atomic electrons caused by collisions has a natural probability and is restricted to the atomic orbitals. In another way, electrons with lower energy levels migrate to higher energy levels during the excitation process. This movement can be defined in two ways:

$$\begin{aligned} stepsize_{i,j}^{Excitation} &= rand\ a \times \Delta x_{i,j} + rand\ b \\ &\quad \times \Delta x_{i,j} \times (1 - t) \\ t &= \frac{it}{Maxit} \end{aligned} \tag{3}$$

where $stepsize_{i,j}^{Excitation}$ is the step size of the $i^{\text{th}}$ atomic electron; *randa* and *randb* are random numbers uniformly distributed in $[0.6 + 0.1 \times \mathrm{t}, 1.4 - 0.1 \times t]$ and $[-\delta y_{i,j}, \delta y_{i,j}]$ intervals, respectively; $\Delta x_{i,j}$ and $\delta y_{i,j}$ are calculated as follows:

$$\Delta x_{i,j} = \begin{cases} x_{i,j} - x_{rs,j} & PCost_i < PCost_{rs} \\ x_{rs,j} - x_{i,j} & PCost_i \geq PCost_{rs} \end{cases} \tag{4}$$

where $x_{i,j}$ and $x_{rs,j}$ are the position of two compared electrons; Here, each electron compares with a randomly selected one ($x_{rs,j}$) except itself for possible improvement of energy level. $\delta y_{i,j}$ is obtained by simulating d-orbital with pear-shaped equation due to their similarity:

$$\delta y_{i,j} = \sqrt{\frac{\left| randa \times \left(\frac{|x_{i,j} - x_{rs,j}|}{e_{j,max} - e_{j,min}}\right)^3 - \left(\frac{|x_{i,j} - x_{rs,j}|}{e_{j,max} - e_{j,min}}\right)^4 \right|}{2 \times iteration}} \tag{5}$$

Eq. (5) shows that as iteration increases, the search domain around better electrons shrinks. The excitation method demonstrates the algorithm's ability to intensify by reducing the size of d-orbitals.

#### 2) DE-EXCITATION PROCESS

Because of the interaction of electron beams with gas atoms, a percentage of excited electrons lose energy by releasing light, causing their locations to shift from high-energy to low-energy. This method can be mathematically expressed as follows:

$$\begin{aligned} &if\ rand_1 < EDR\ and\ rand_2 < DR,\ X \\ &otherwise,\ Y \\ &X = NDRS = ceil\ (DRS \times d) \\ &Y = De - excitation\ process\ does\ not\ occur \end{aligned} \tag{6}$$

in which $rand_1$ and $rand_2$ are random numbers generated between 0 and 1; The De-excitation Rate ($DR$) parameter specifies whether or not the de-excitation process occurs. $DRS$ denotes the de-excitation rate for each excited-state electron. Ceil is the operator that rounds the obtained value toward positive infinity; this parameter is a regulating parameter that determines how many dimensions each electron can perform for de-excitation. To perform the de-excitation procedure, the following equation is used to obtain the set of dimensions of each electron chosen at random:

$$K = randsample\,(d, NDRS) \tag{7}$$

where K returns NDRS values sampled uniformly, without replacement, from the integer 1 to d (number of design variables), in the following, changing atomic electrons position are occurred based on steps which are obeyed from normal distribution as:

$$stepsize_{i,k}^{De-excitation} = stepsize_{i,k}^{Excitation} + rand \times \left(e_{k,max} - e_{k,min}\right) \tag{8}$$

where $rand$ is a normal distributed random number with mean 0 and variance 1.

### D. STEP 4: BASED ON THE IONIZATION MECHANISM, GENERATE A NEW ELECTRON POSITION

Atoms collide with high-energy electron beams. Any of these atomic electrons are torn from their atoms and thrown into the plasma. Immersed electrons collide with other atoms due to their high kinetic energy. As a result, the atoms in question are excited and become ions. The movement of electrons, which follows the laws of levy flight, is used to model this operation. To put it another way, the trajectories of electrons immersed in plasma follow the levy distribution and can be mathematically simulated as follows:

$$stepsize_{i,j}^{Ionization} = rand \times S_{i,j} \times \Delta x_{i,j} \times (1 - t) \tag{9}$$

where $rand$ is a normally distributed rand number with mean 0 and variance 1;$t$ and $\Delta x_{i,j}$ are obtained as per Eqs. (3)-(4), respectively, and $S_{i,j}$ can be calculated as follows:

$$S_{i,j} = \frac{rand_1}{|rand_2|^{\frac{1}{\beta}}} \times \sigma \tag{10}$$

where $\beta$ is a constant equal to 1.5 in this study; $rand_1$ and $rand_2$ are normally distributed random numbers, and $\sigma$ is calculated as follows:

$$\sigma = \left( \frac{\Gamma\,(1+\beta) \times sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}}$$
$$\Gamma\,(x) = (x-1)\,! \tag{11}$$

Because of the extremely long jump made in levy flight, the ionization process demonstrates the algorithm's diversification capability.

**Algorithm 1** Procedure of the PGO Algorithm

```
Set algorithm parameters (n, Maxit, EDR, DR, DRS).
Evaluate the initial candidate solutions.
while(termination condition not met) do
 for i: 1 to n
   Select randomly one other electron among n
   electrons (e.g. rs), except (i).
   if rand1 < EDR (Excitation process occurs)
    Generate new solution i based on the excitation
    process using Eqs. (4) and (12).
    if rand2 < DR (De-excitation process occurs)
     Calculate NDRS and K using Eqs. (7) and (8)
     Changing some dimensions of the newly
    generated solution i using Eqs. (9) and (12).
    end if
   else (Ionization process occurs)
   Generate new solution i based on the ionization
   process using Eqs. (10) and (12).
   end if
   Checking lower and upper bounds of design variables
   for i^th electron.
   Evaluating the objective function for i^th electron.
   Updating the position of the i^th electron using Step 5
 end for
 end while
end
```

### E. STEP 5: UPDATING THE ELECTRON's POSITION

In the previous step, Eqs. (3), (8), and (9) were used to determine the new position of the electron (Eq. 12). The newly produced electron is then compared to the one generated in the previous iteration. It is substituted if the newly created one is better. The best electron in each iteration, on the other hand, is compared to the best electron obtained thus far, and if the best electron is better than the best electron obtained thus far, it is substituted.

$$e_{i,j}^{iteration+1} = e_{i,j}^{iteration} + \begin{cases} stepsize_{i,j}^{Excitation} \\ stepsize_{i,k}^{De-excitation} \\ stepsize_{i,j}^{Ionization} \end{cases}$$
$$\begin{cases} if\ rand_1 < EDR \\ if\ rand_1 < EDR\ and\ rand_2 < DR \\ if\ rand_1 \geq EDR \end{cases} \tag{12}$$

### F. STEP 6: CHECKING TERMINATION CONDITION

The optimization process is terminated if the number of iterations exceeds the maximum number of iterations set as the stopping criterion, and the best solution found thus far will be announced. Otherwise, the process loops back to Step 2 for the next iteration. The pseudocode for the PGO algorithm is given in *Algorithm 1*.

## III. MULTI-OBJECTIVE PLASMA GENERATION OPTIMIZATION (MOPGO) ALGORITH

In this paper, the NDS and CD-based PGO algorithm is proposed for the MO optimization problem. The NDS method handles the Pareto dominance issue and tries to keep the non-dominated solution into the next generation, and the CD technique maintains the diversity of solutions [43]. Apart from the Pareto dominance and diversity research issue, this research also addressed another MO algorithm-related subject that how to select one Pareto solution from the Pareto frontier for a real-world application in practice through the fuzzy-based decision and computational complexity measurement. This paper contributes a new MOPGO algorithm that confronts three crucial aspects, i.e., optimization capability of the search algorithm, Pareto dominance, and solution diversity concurrently. By this suggested framework, the search algorithm ensures a proper balance between global diversification and local intensification. The NDS comprises the subsequent phases.

- First, determining the non-dominated solution.
- Second, the application of the NDS approach.
- Calculating non-dominated ranking (NDR) of all non-dominated solutions.

The NDR process happens between two fronts. The solutions in the first front give a '0' index since it is not dominated by any solutions, at the same time, the solutions in the second front are dominated by at least one solution in the first front. Such NDR of the solutions is equal to the number of solutions that dominate them. The concept of two populations is further explained as follows. Firstly, for each solution obtained from the basic search method (i.e., PGO) or initially generated random population $P_o$, all the objectives from the objective vector $F$ are evaluated. In addition, a domination count $n_p$ defined as the number of solutions dominating the solution $p$ and $S_p$ which is a set of solutions dominated by solution $p$ are calculated. Secondly, all the solutions $p$ is assigned a domination count zero and are put in the first non-dominated level, also known as Pareto Front *(PF),* and their non-domination rank ($NDR_p$) is set to 1. Thirdly, for each solution $p$ with $n_p = 0$, each member $q$ of the set $S_p$ is visited and its domination count $n_q$ is reduced by one. While reducing $n_q$ count if it falls to zero, the corresponding solution $q$ is put in second non-domination level and $NDR_q$ is set to 2. The procedure is repeated for each member of the second non-domination level to obtain the third non-domination level, and subsequently, the procedure should be repeated until the whole population is sorted into different non-domination levels.

The CD mechanism is utilized to maintain diversity between the developed solutions. In the crowding distance approach for maintaining diversity among the obtained solutions, firstly, the population is sorted according to the value of each objective function in ascending order called ''sorting by fitness.'' The concept of ''sorting by fitness'' is explained in detail with one simple example in Appendix. An infinite crowding distance is then assigned to the boundary solutions, $i = 1$ and $i = l$, of each objective. Here $l$ is the total number of solutions in a particular non-dominated set. The boundary solutions are the minimum *(i =1)* and maximum ($i = l$) function values. Except for the boundary solutions, all the other solutions of the sorted population *(i = 2 to l-1)* for each objective $j$ ($j = 1, 2, \ldots, m$) are assigned, and the CD mechanism is defined as follows.

$$CD_j^i = \frac{fobj_j^{i+1} - fobj_j^{i-1}}{fobj_j^{max} - fobj_j^{min}} \tag{13}$$

where $fobj_j^{max}$ and $fobj_j^{min}$ are the maximum and minimum values of $j^{\text{th}}$ objective function. In Eq. 13, the right-hand side term is the difference in values of objective function $j$ for two neighboring solutions ($i + 1$ and $i - 1$) of solution $i$. The diagrammatic illustration of an NDS-based approach is illustrated in Fig. 1. *Algorithm 2* shows the pseudocode of the MOPGO algorithm. The algorithm starts with defining the required parameters, including population size ($N_p$), termination criteria, and a maximum number of generation/maximum number of iteration ($Max_{it}$) to run the MOPGO algorithm. Then, a randomly generated parent's population $P_o$ in feasible search space region $S$ is created, and each objective function in the objective space vector $F$ for $P_o$ is assessed. Thirdly, the elitist-based CD and NDS are applied to $P_o$. Fourthly, a new population of $P_j$ is generated and combined with $P_o$ to get population $P_i$. This $P_i$ is sorted based on elitist non-domination and the obtained data of CD and NDR.

**Algorithm 2** Pseudocode of Multi-Objective Plasma Generation Optimization (MOPGO) Algorithm

**Step 1:** Initially Generate population ($P_o$) randomly in solution space (S)

**Step 2:** Evaluate objective space (F) for the generated population ($P_o$)

**Step 3:** Sort the based on the elitist non-dominated sort method and find the non-dominated rank (NDR) and fronts

**Step 4:** Compute crowding distance (CD) for each front

**Step 5:** Update solutions ($P_j$) using ***Algorithm 1***

**Step 6:** Merge $P_o$ and $P_j$ to create $P_i = P_o$ U $P_j$

**Step 7:** For $P_i$ perform ***Step 2***

**Step 8:** Based on NDR and CD sort $P_i$

**Step 9:** Replace $P_o$ with $P_i$ for $N_p$ first members of $P_i$

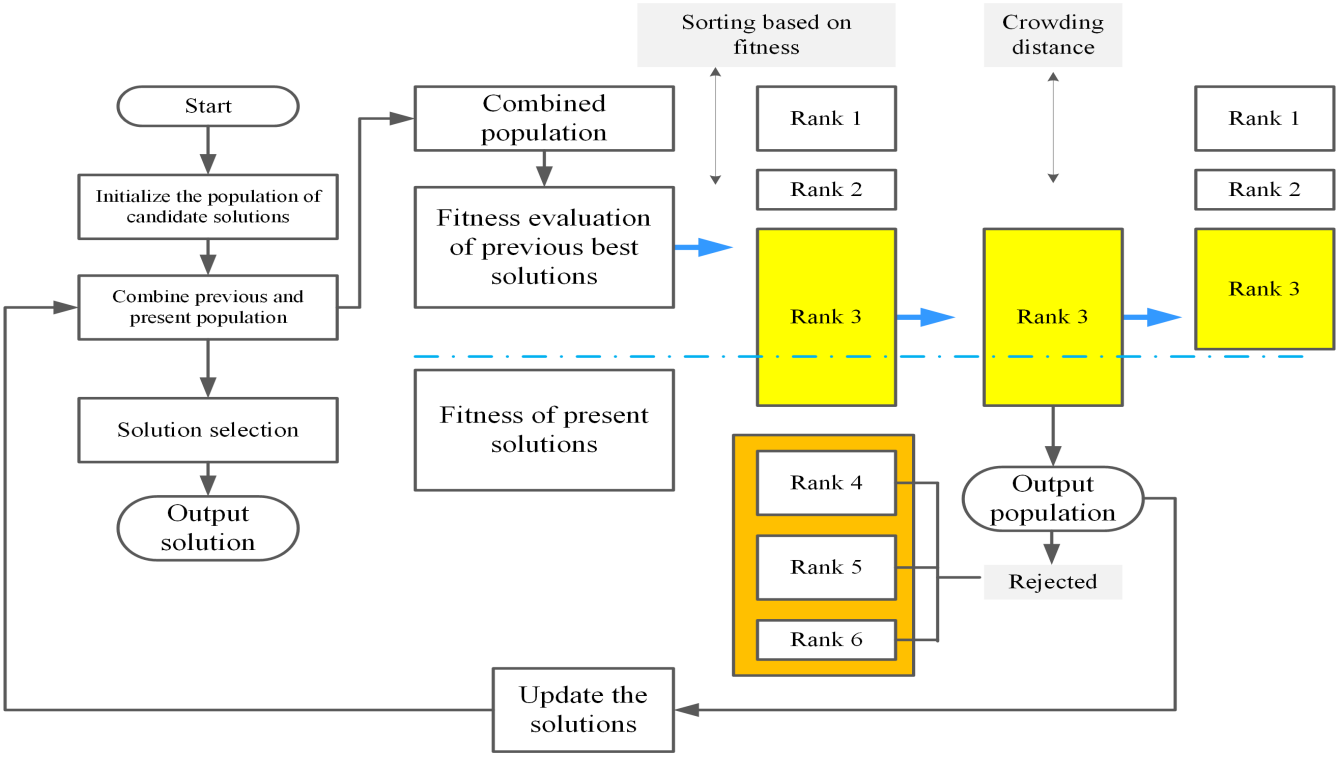


**FIGURE 1.** The procedure of non-dominated sorting approach.

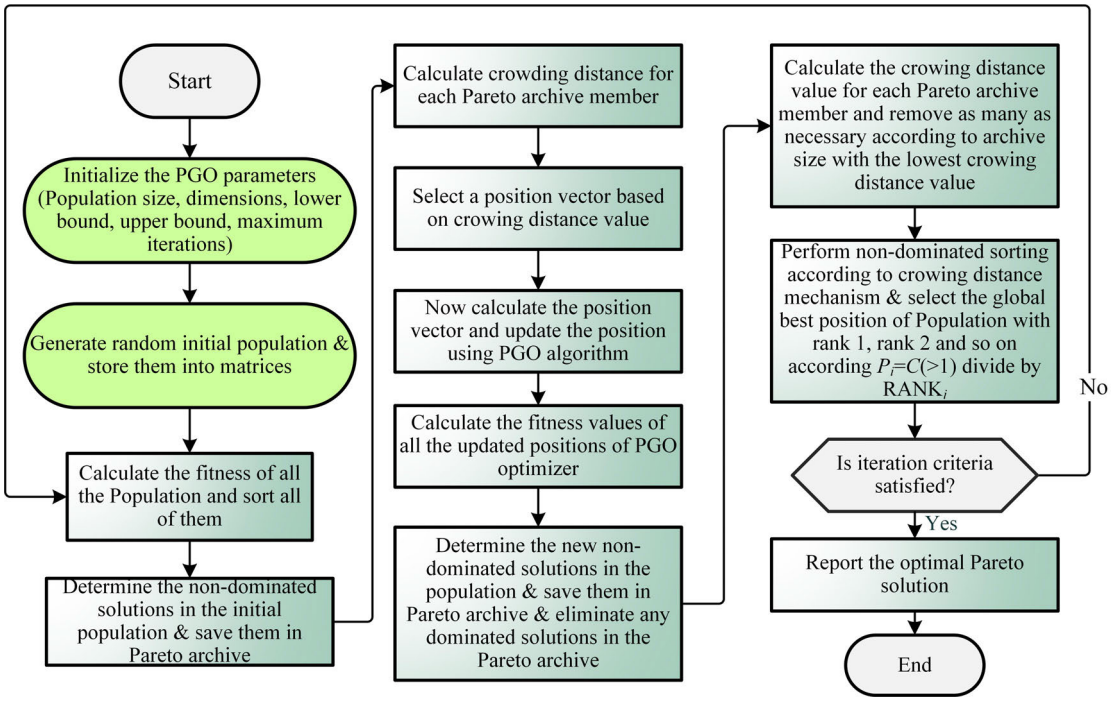


**FIGURE 2.** Flowchart of the MOPGO algorithm.

The best $N_p$ solutions are reviewed to create a new parent population. Lastly, this procedure is repeated until the termination criteria. The flowchart of MOPGO is shown in Fig. 2.

### *A. BEST COMPROMISE SOLUTION (BCS) BASED ON FUZZY DECISION*

To find the best solution that provides the best degree of satisfaction to each objective is pursued out of feasible

*Pareto*-optimal solutions and governed by fuzzy membership [44] functions $\mu_i$ defined as follows:

$$\mu_i^j = \begin{cases} 1, & \text{if } f_i^j \leqslant f_{\min}^j \\ \dfrac{f_{\max}^j - f_i^j}{f_{\max}^j - f_{\min}^j}, & \text{if } f_{\min}^j \leqslant f_i^j \leqslant f_{\max}^j \\ 0, & \text{if } f_i^j \geqslant f_{\max}^j \end{cases} \quad (14)$$

$$\mu_i(Normalized) = \frac{\sum_{j=1}^{N_{obj}} \mu_{ij}}{\sum_{i=1}^{M} \sum_{j=1}^{N_{obj}} \mu_{ij}} \quad (15)$$

where $M$ is the number of non-dominated solutions, $N_{obj}$ is the number of the objective function, and $f_{\max}^j$ and $f_{\min}^j$ are the maximum and minimum values of the respective objective function. The best-compromised value is the one with a high value of $\mu_i$.

### B. CONSTRAINT HANDLING APPROACH

The MOPGO algorithm, a static penalty approach, is employed as follows.

$$f_j(X) = f_j(X) + \sum_{i=1}^{p} P_i \max\{g_i(X), 0\} + \sum_{i=p}^{NC} P_i \max\{|h_i(X)| - \delta, 0\} \quad (16)$$

where $f_j(X), j = 1, 2 \ldots n$ is the objective function to be optimized (here minimized), $X = \{x_1, x_2, \ldots x_m\}$ are design variables, $g_i(X) \leqslant 0, i = 1, 2 \ldots p$ are inequality constraints, $h_i(X) = 0, i = p+1 \ldots NC$ are equality constraints, and $\delta$ is tolerance inequality constraints.

### C. COMPUTATION COMPLEXITY (CC) OF MOPGO ALGORITHM

The CC of the MOPGO algorithm is represented in terms of time complexity and space complexity. As per earlier discussion, the suggested MOPGO algorithm utilizes the NSGA-II operators [11]. Since the NDS and CD mechanisms are adopted from NSGA-II, the computational space complexity of MOPGO similar to MOPVS, MOSMA, MOALO, and MOSOS optimizers are $O(MN_p)^2$, where $N_p$ is the number of search agents/population size, and $M$ is the total number of objective functions. The computational time complexity of MOPGO is given for each iteration. The complexity is equal to $O(dim^*N_p + Cost(f_{obj})^*N_p)$ for the first iteration. The computational time complexity is equal to $O(dim^* N_p + Cost(f_{obj})^*N_p + (NDS + CD)^*dim)$ after the first iteration. The overall computational time complexity is given for the $Max_{it}$ to time $= O(M)|M = O(dim^*Max_{it}{}^*N_p + Cost(f_{obj})^*Max_{it}{}^*N_p + (NDS + CD)^*(Max_{it})^*dim + (NDS + CD)^*(Max_{it}) * Cost(f_{obj}))$. The cost of the objective function is denoted as $Cost(f_{obj})$, the objective function is denoted by $f_{obj}$, the current iteration is symbolized as t, and the maximum number of iterations is symbolized as $Max_{it}$, and the number of variables in the objective function is represented as *dim*.

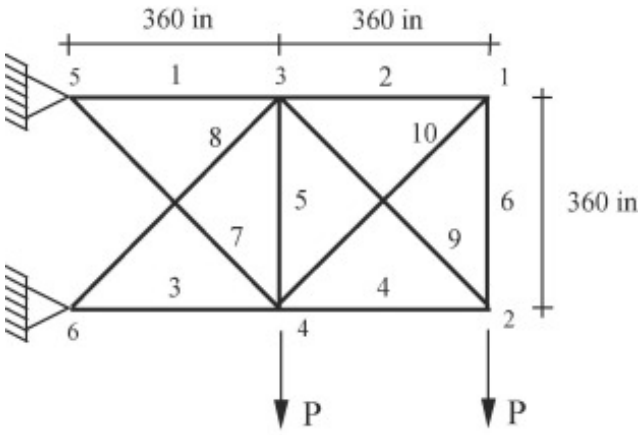


**FIGURE 3.** The 10-bar truss.

## IV. MATHEMATICAL FORMULATION OF MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

In this paper, structure weight is the first objective that needs to be minimized, whereas maximum nodal deflection is the second objective. The mathematical formulation for the MO truss optimization problem is as follows:

$$\left.\begin{aligned} &\text{Find}, A = \{A_1, A_2, .., A_m\} \\ &f_1(A) = \sum_{i=1}^{m} A_i \rho_i L_i \\ &f_2(A) = max\left(\left|\delta_j\right|\right) \end{aligned}\right\} \quad (17)$$

Subjects to:

Behavior constraints:

Stress constraints, $g(A) = |\sigma_i| - \sigma_i^{max} \leq 0$

Side constraints:

Cross $-$ sectional area constraints, $A_i^{min} \leq A_i \leq A_i^{max}$

where, $i = 1, 2, .., m; j = 1, 2, .., n$. Here, $A_i$ is a design variable vector; $\rho_i$ and $L_i$ are the mass density and length of the elements, respectively; $E_i$ and $\sigma_i$ Correspond to the '$i$' element Modulus of elasticity and stress, respectively. Moreover, the allowable upper and lower bounds are represented by superscripts '*max*' and '*min*,' respectively.

## V. EMPIRICAL EVALUATION

To examine the convergence, coverage, intensification, and diversification of the proposed MOPGO, numerous 2-D and 3-D structural tests were examined and contrasted with other state-of-the-art MO optimization strategies existing in the literature, viz. MOPVS [4], MOSMA [45], MOSOS [25], and MOALO [23]. The subsequent section elaborates on the eight truss problems, i.e., 2-D 10-bar, 3-D 25-bar, 3-D 60-bar ring, 3-D 72-bar, 3-D dome 120-bar, 3-D 200-bar, and tower 942-bar truss problems that were considered. Table 1 presents all design considerations and mechanical properties used to simulate eight truss MO optimization examples in one frame. Moreover, Figs. 3, 6, 9, 12, 15, 18, 21, 24 illustrates load directions, constraints, and truss dimensions.

**TABLE 1.** Design considerations of the truss problems [1].

| ***Truss bar problems*** | 10 *bar* | 25 *bar* | 37 *bar* | 60 *bar* | 72 *bar* | 120 *bar* | 200 *bar* | 942 *bar* |
|---|---|---|---|---|---|---|---|---|
| *Design variables* | $Zi, i = 10$ | $Zi, i = 8$ | $Zi, i = 15$ | $Zi, i = 25$ | $Zi, i = 16$ | $Zi, i = 7$ | $Zi, i = 29$ | $Zi, i = 59$ |
| Constraints ($Pa$) | $\sigma_{max} = 400e6$ | | | | | | | |
| Density ($kg/in^3$) | $\rho = 7850$ | | | | | | | |
| Young Modulus ($Pa$) | $E = 200e9$ | | | | | | | |
| Loading conditions ($N$) | $P_{y2}$=$P_{y4}$=−100e4 | $P_{x1}$=1e5, $P_{y1}$=$P_{z1}$=$P_{y2}$ = $P_{z2}$=−10e5, $P_{x3}$=5e4, $P_{x6}$=6e4 | - | Case 1: $P_{x1}$=−10e, $P_{x7}$=9e5 Case 2: $P_{x15}$=$P_{x18}$=−8e5, $P_{y15}$=$P_{y18}$=3e5 Case 3: $P_{x22}$=−20e5 $P_{y22}$=10e5 | Case 2: $P_{1x}$=$P_{1y}$=2e6 $P_{1z}$=−2e6 Case 2: $P_{1z}$=$P_{2z}$=$P_{3z}$= $P_{4z}$=−2e6 | - | - | At each node: Vertical loading: Section 1; $P_z$=−6e3 Section 2; $P_z$=−12e3 Section 3; $P_z$=−18e3 Lateral loading: Right-hand side; $P_x$=3e3 Left-hand side; $P_x$=2e3 Lateral Loading: $P_y$=2e3 |

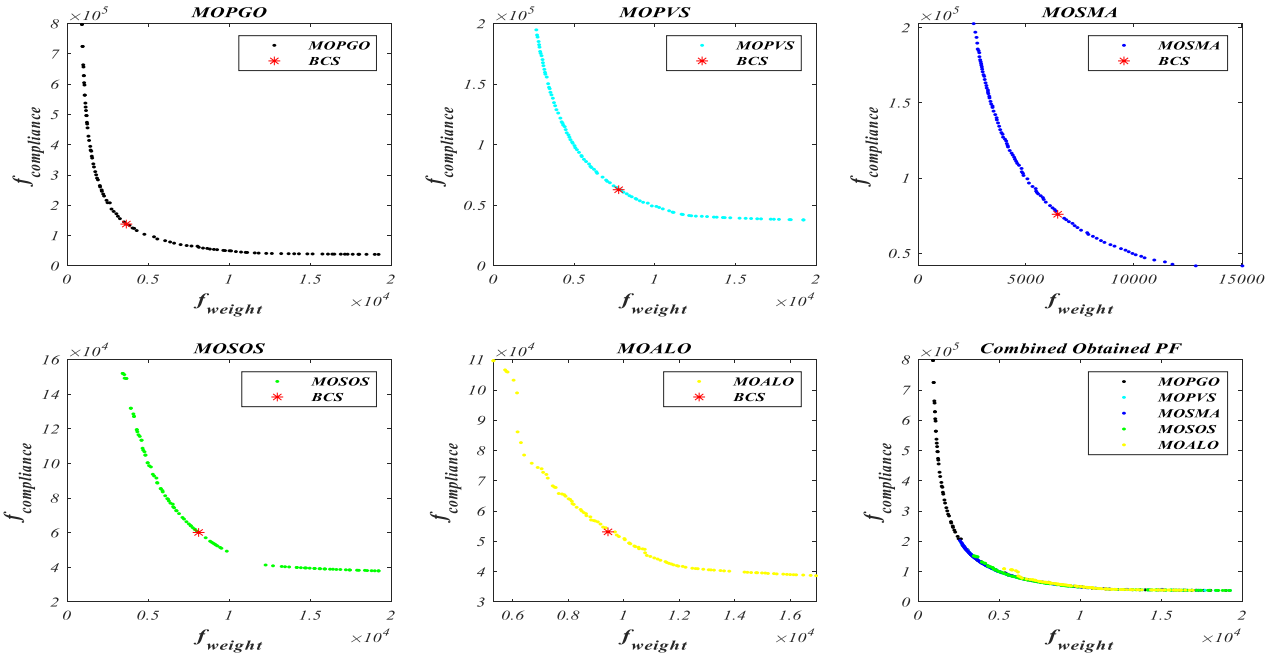


**FIGURE 4.** Best Pareto fronts of the 10-bar truss by all algorithms.

### *A. EVALUATION METHOD*

In this study, every algorithm is executed 30 times individually for all considered eight truss design problems with the population size of 40, the maximum number of iteration of 500, and the maximum number of function evaluations of 2000 [46].

- The Hypervolume (HV) and Inverted Generational Distance (IGD) metric are employed to concurrently examine the uniformity-convergence-spread of the non-dominated set of solutions procured from the computation experiments.
- To examine the search efficiency and reliability of considered algorithms in terms of faster convergence rate Generational Distance (GD), Spread (SD), Coverage (CVG), and Coverage over Pareto Front (CPF) metrics are used [46]–[48].
- To measure the computational complexity, Runtime (RT) metric and for combined diversity-spread,

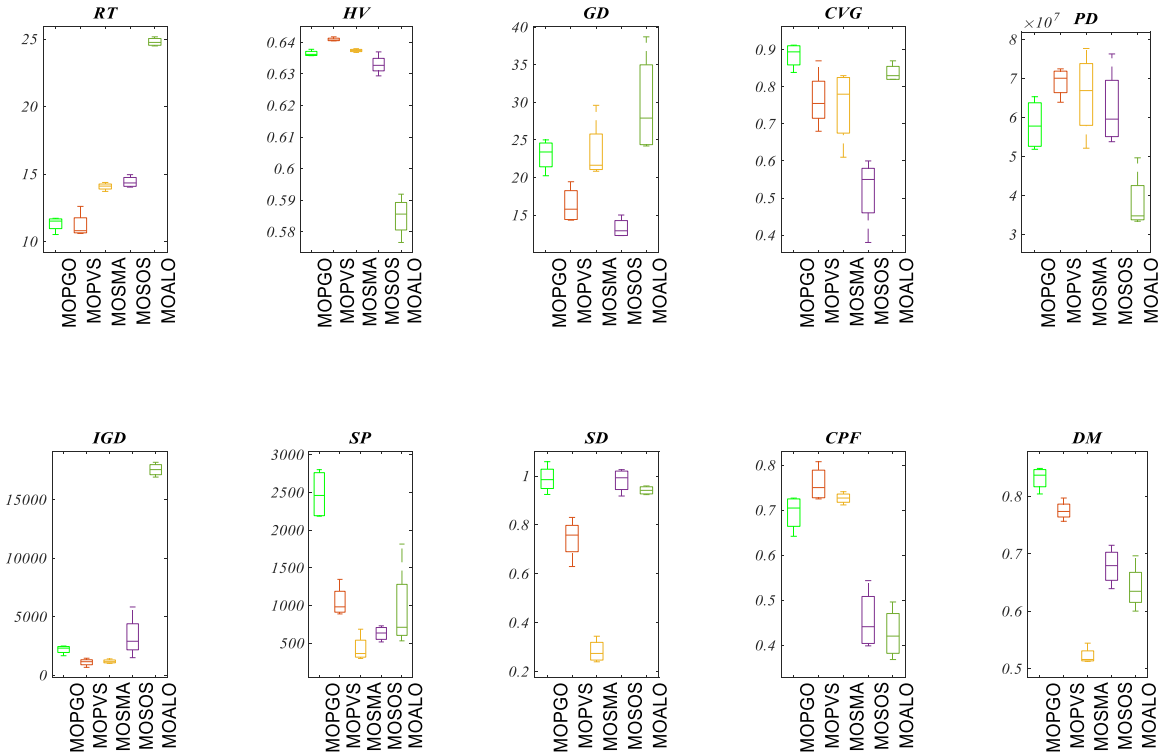


**FIGURE 5.** Boxplots of the 10-bar truss by all algorithm.

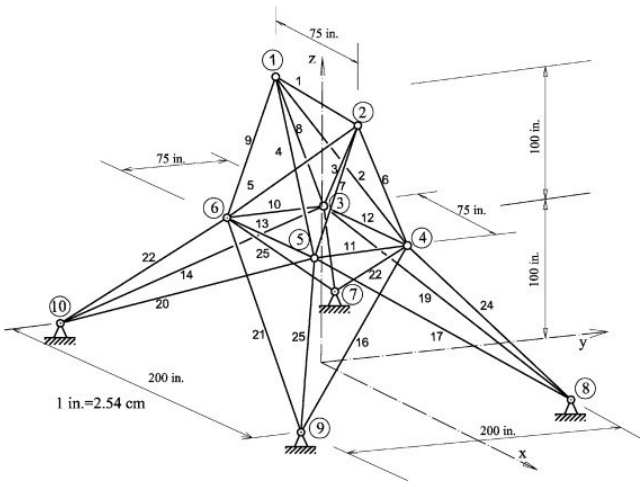


**FIGURE 6.** The 25-bar spatial truss.

spacing (SP), Diversity Maintenance (DM), and Pure Diversity (PD) metrics are calculated [46]–[48].

- The mean and standard deviation (STD) values of the metrics are regarded as the statistical performance measure [4], [15].
- Friedman's rank test (FNRT) is a statistical review of all the optimizers examined [20], [24].

$$GD = \frac{\sqrt{\sum_{i=1}^{no} d_i^2}}{n} \tag{18}$$

$$IGD = \frac{\sqrt{\sum_{i=1}^{nt} \left(d_i'\right)^2}}{n} \tag{19}$$

$$SP \triangleq \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2} \tag{20}$$

$$SD = \sqrt{\sum_{i=1}^{o} \max\left(d\left(a_i, b_i\right)\right)} \tag{21}$$

$$PD = \frac{\sum_{H(i,j,\ldots)\neq 0} m\left(h\left(i,j,\ldots\right)\right)}{\sum_{H(i,j,\ldots)\neq 0} m\left(H\left(i,j,\ldots\right)\right)} \tag{22}$$

$$HV = \Lambda\left(\bigcup_{s\in PF}\left\{s' \mid s \prec s' \prec s^{nadir}\right\}\right) \tag{23}$$

$$DM = \frac{d_f + d_l + \sum_{i=1}^{N-1}\left|d_i - \bar{d}\right|}{d_f + d_l + (N-1)\,\bar{d}} \tag{24}$$

$$CVG = \frac{\sum_{i=1}^{n}\psi_i}{N}, \psi_i = \begin{cases} 1, & \text{if } P_i \in PF \text{ and } \alpha_{i-1} \leq \tan\dfrac{f_1(x)}{f_2(x)} \leq \alpha_n \\ 0, & \text{Otherwise} \end{cases} \tag{25}$$

$$CPF = \frac{\sum_{i=1}^{n} PF_i}{N} \tag{26}$$

$$RT = \frac{\sum_{i=1}^{n} \mathrm{T}}{n} \tag{27}$$

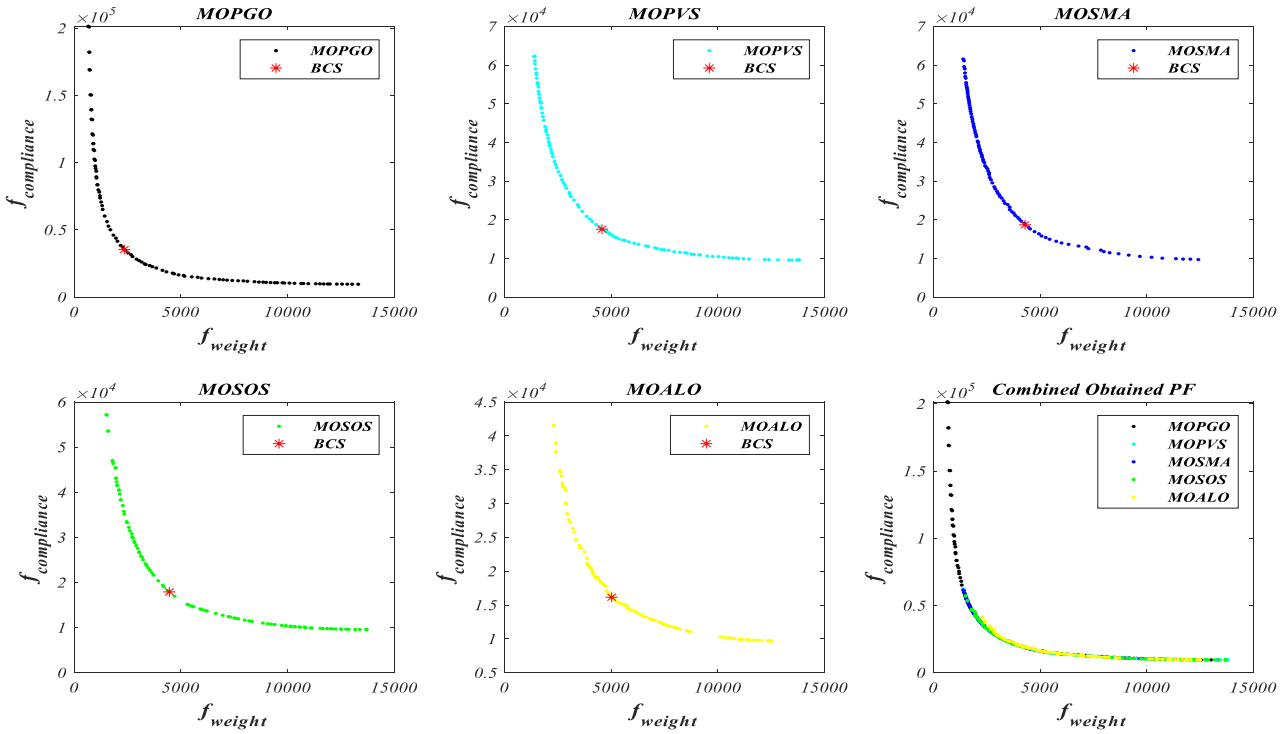


**FIGURE 7.** Best Pareto fronts of the 25-bar truss by all algorithms.

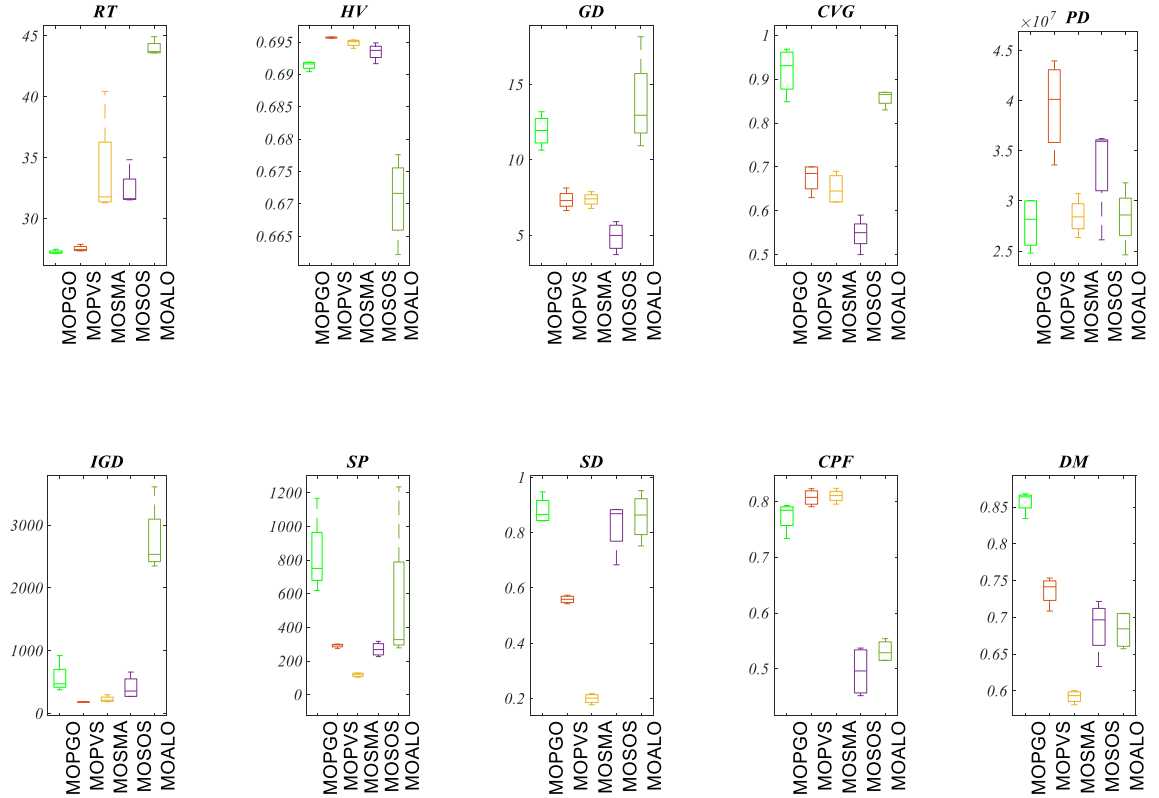


**FIGURE 8.** Boxplots of the 25-bar truss by all algorithms.

where *no* is the number of True Pareto solution (PS), *nt* is the number of true Pareto optimal solutions, *o* is the number of objectives, $\overline{d}$ is the average of all $d_i$, $d_i$, and $d_i'$ specifies the Euclidean distance, $n$ is the number of obtained PS,

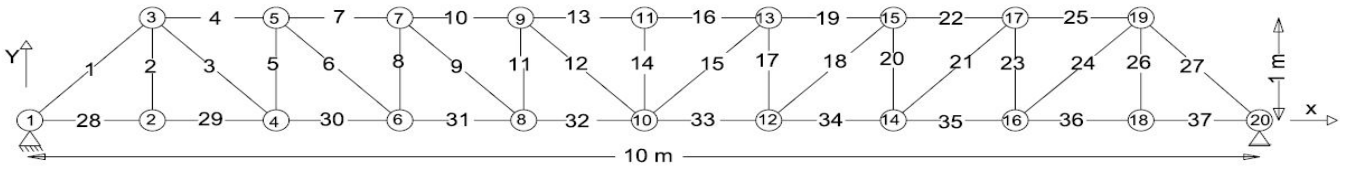


**FIGURE 9.** The 37-bar truss.

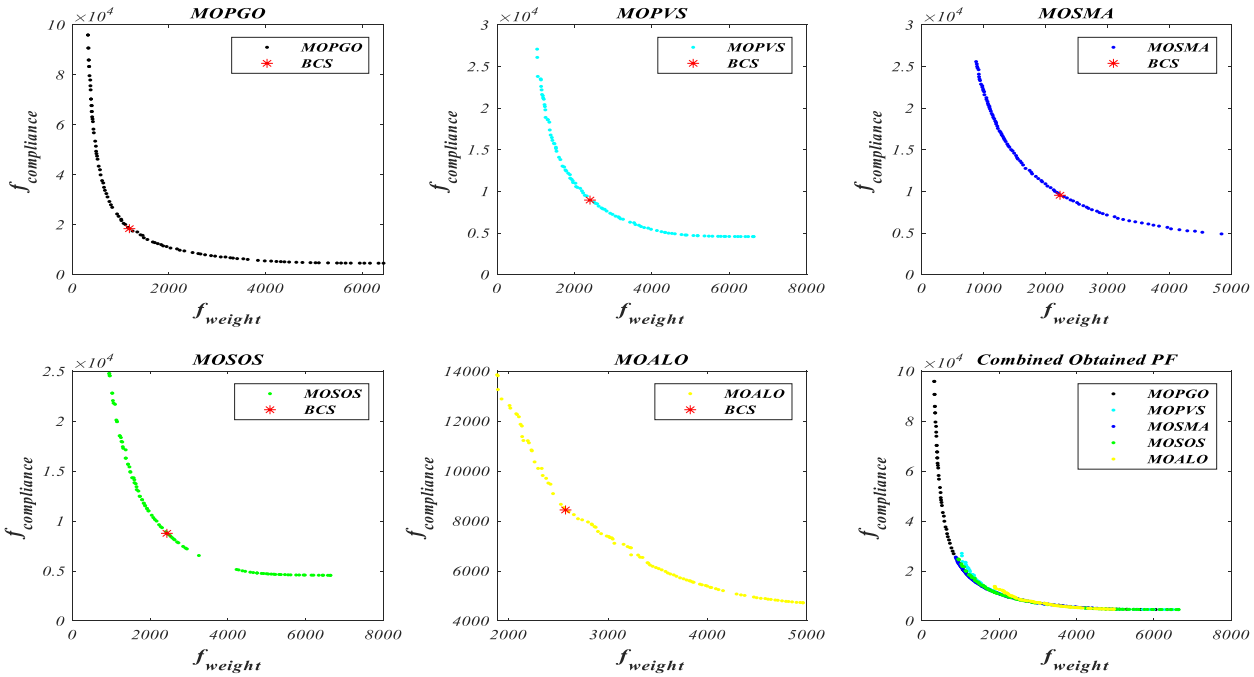


**FIGURE 10.** Best Pareto fronts of the 37-bar truss by all algorithms.

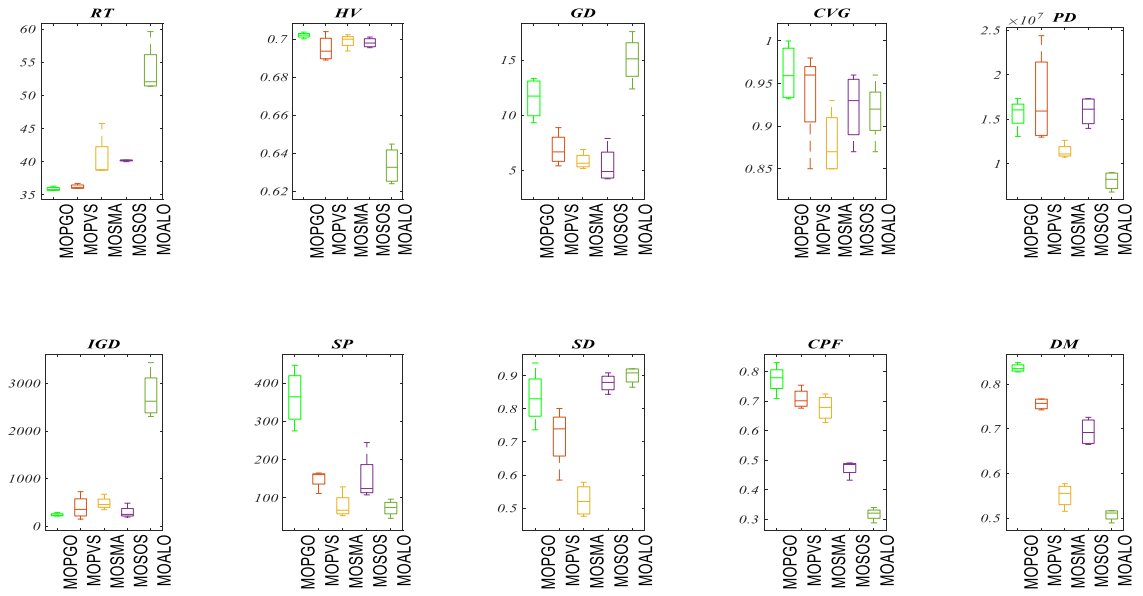


**FIGURE 11.** Boxplots of the 37-bar truss by all algorithms.

$d_i = \min_j \left(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})\right)$ for all $i,j = 1,2,\ldots,n.$, $a_i$ and $b_i$ is the maximum and minimum value in the $i^{\text{th}}$ objective.

### B. RESULTS AND DISCUSSIONS

The data obtained by all selected algorithms, such as MOPGO, MOPVS, MOSMA MOSOS, and MOALO for all

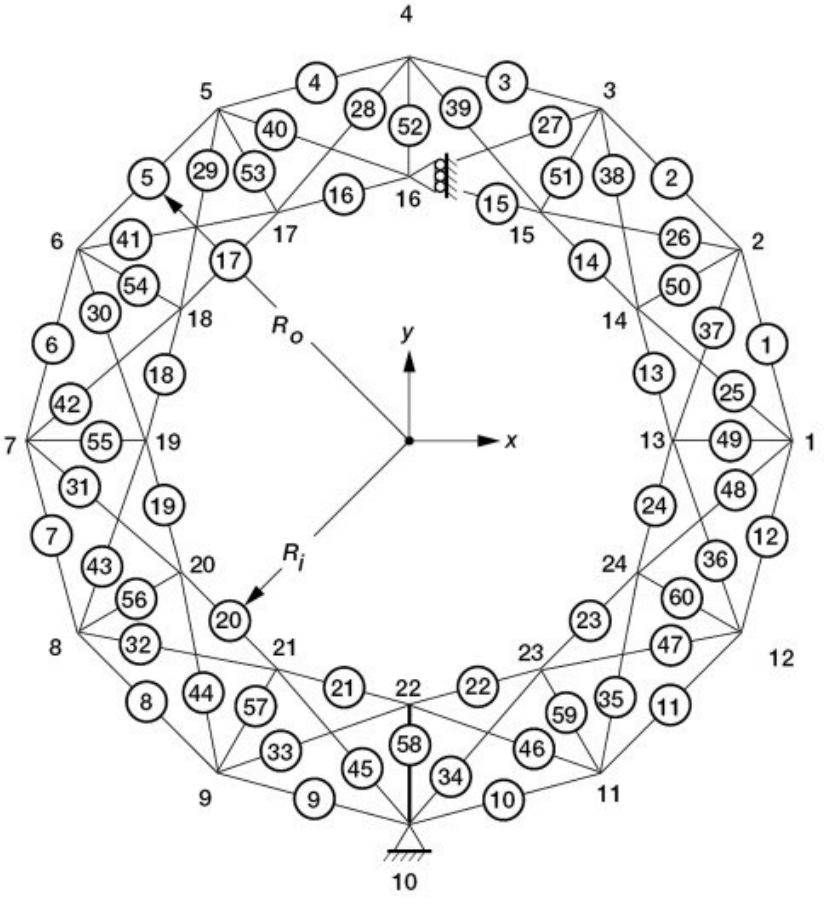


**FIGURE 12.** The 60-bar ring truss.

considered benchmarks, are illustrated in Tables 2–12 as per the sequence of all performance metrics.

### 1) 10-BAR PLANAR TRUSS

The HV metric results are illustrated in Table 2, from which it is evident that MOPGO achieved the best functional mean ($f_{\text{mean}}$) and standard deviation ($f_{\text{std}}$) values, relative to other considered algorithms. Moreover, the *FNRT* metric assigned the rank of 500, 300, 375, 200, 225, and 100 to the MOPGO, MOPVS, MOSMA, MOSOS, and MOALO, respectively. Hence at a 95% significance level, MOPGO outperforms by demonstrating its high solutions density in the proximity of the Pareto Front. For GD indicator, $f_{\text{mean}}$ results for MOPGO from Table 3 show a percentage decrease of 45.66%, 43.26%, 18.67%, and 42.23% with respect to MOPVS, MOSMA, MOSOS, and MOALO, respectively. Similarly, MOPGO obtain the least $f_{\text{std}}$ value of 1.293634 relative to others. Moreover, MOPGO found the least *FNRT* value, i.e., 100, followed by MOALO, MOSMA. Therefore, MOPGO has a better quality of convergence as per *FNRT* at 95% significance level. According to Table 4, the best CVG metric $f_{\text{mean}}$ was obtained by MOALO and stood first as per *FNRT* while the proposed MOPGO technique achieves the best $f_{\text{std}}$ value and demonstrate its enhanced coverage characteristic. In terms of CPF metric as reported in Table 5, the MOPVS, MOSMA, and MOPGO demonstrate their improved quality relatively and settled at 475, 400, and 325*FNRT* values, respectively. Table 6 depicts the DM metric results according to which MOPGO $f_{\text{mean}}$ value has a percentage increase of 29.63%, 22.61%, 59.23%, and 7.27% from MOALO, MOSOS, MOSMA, and MOPVS at least $f_{\text{std}}$ results. MOPGO, as per the *FNRT* obtained a maximum value of 500 relatively and at 95% significance level ranked first. These indicator outcomes exhibit the improved solution diversity of MOPGO concerning other contrasted methodology.

Similarly, for PD performance measure as illustrated in Table 7, the MOPGO algorithm demonstrates its superior pure diversity behavior through its best $f_{\text{mean}}$ and $f_{\text{std}}$ that was eventually proved by its highest 425 *FNRT* value in comparison to other selected optimization techniques. For SP measure, according to Table 8, MOPGO evidence a significant percentage decrease in its $f_{\text{mean}}$ value of 82.744%, 59.29%, 54.63%, and 32.10% relative to MOPVS, MOSMA, MOALO, and MOSOS, respectively. MOPGO also attain the best $f_{\text{std}}$ value of 176.6763, which is substantially less

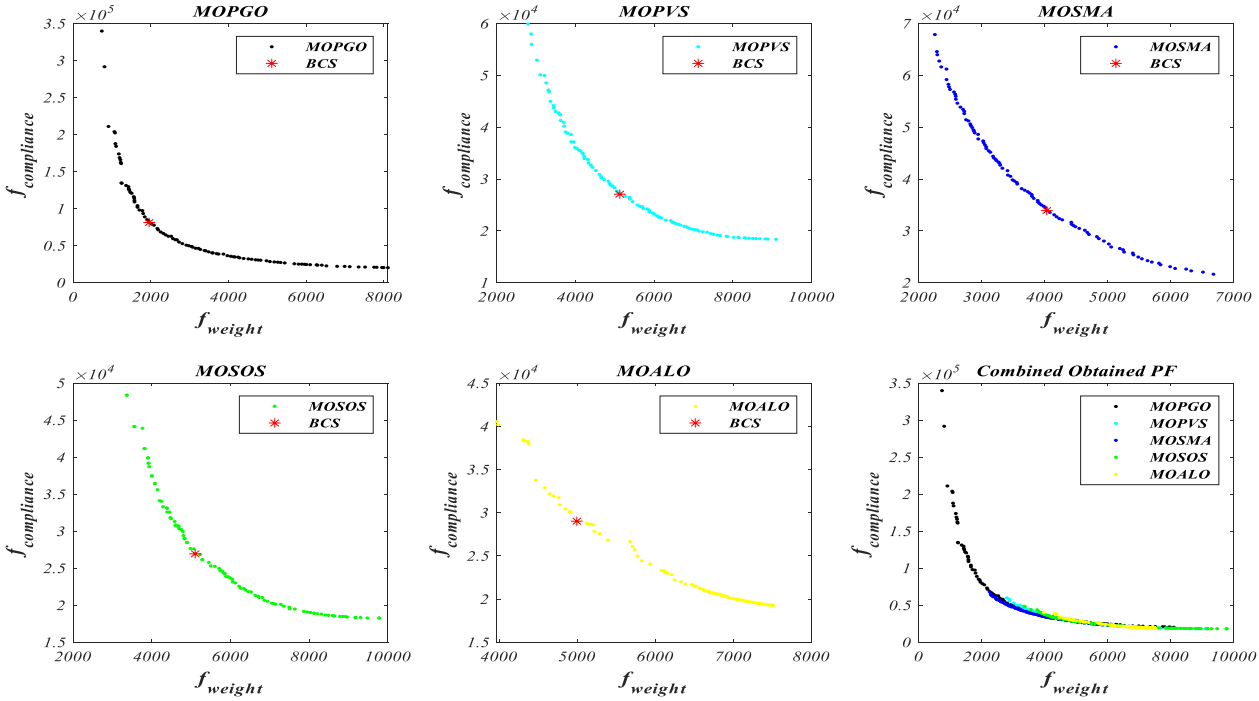


**FIGURE 13.** Best Pareto fronts of the 60-bar truss by all algorithms.

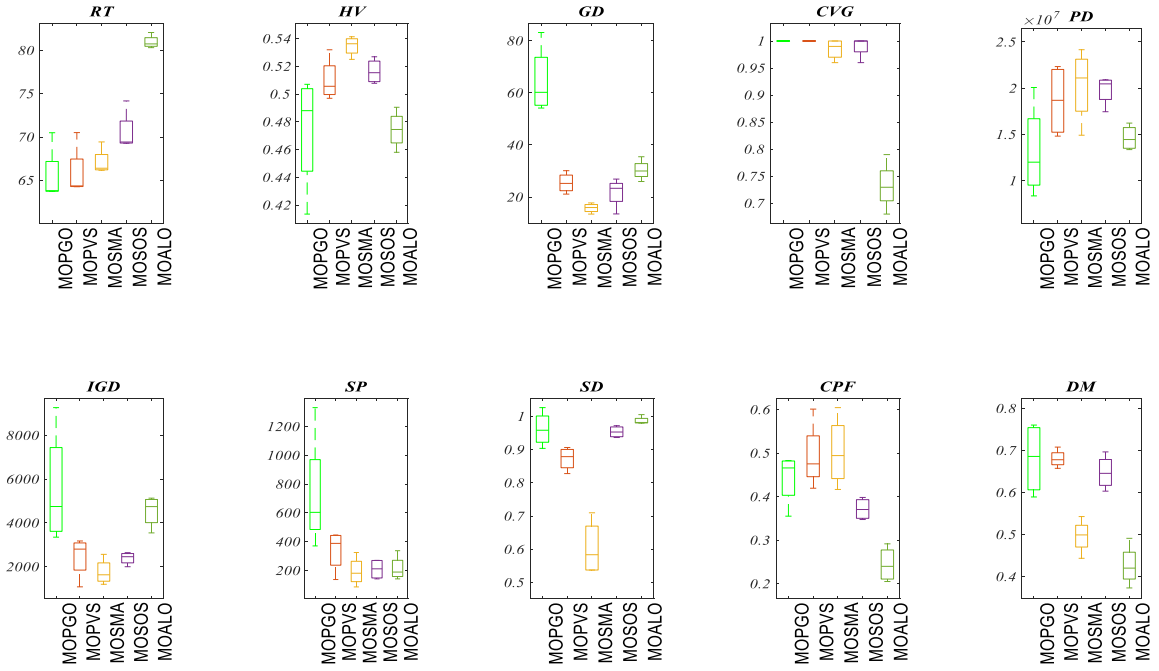


**FIGURE 14.** Boxplots of the 60-bar truss by all algorithms.

relatively. Furthermore, the *FNRT* metric indicates the best rank of 125 by MOPGO and thus, at 95% significance level, displays its enhanced spacing quality. From SD metric findings as shown in Table 9, MOPGO finds the superior $f_{\text{mean}}$ and $f_{\text{std}}$ value of 0.283055 and 0.046962 pertaining to other algorithms and achieve the best *FNRT* value of 100 followed by MOSMA. At a 95% significance level, MOPGO displays its well-distributed non-dominated solutions. In the IGD test, as indicated in Table 10, the MOPVS and MOSMA manifest superior $f_{\text{mean}}$, $f_{\text{std}}$ and *FNRT* results while MOPGO ranked third, showing its competing convergence-spread parity attribute. In terms of RT measure, the MOPVS realize the

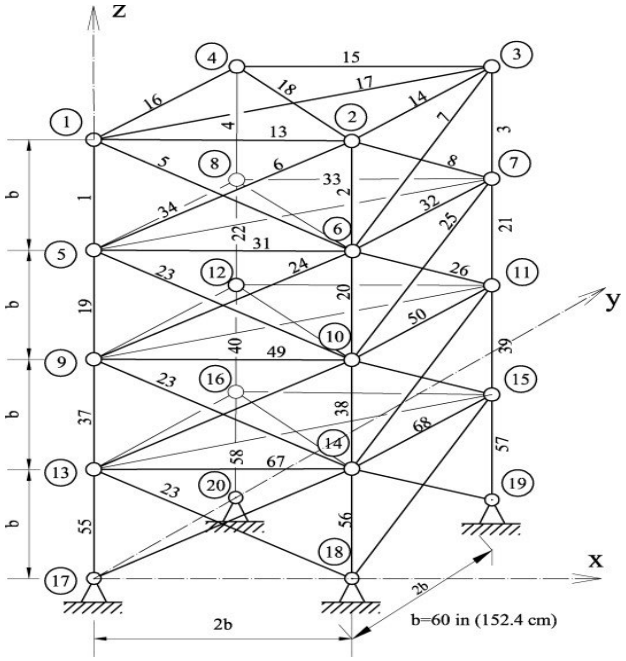


**FIGURE 15.** The 72-bar 3D truss.

best $f_{\text{mean}}$ value followed by MOPGO while the least $f_{\text{std}}$ value is an exhibit by MOSMA, as shown in Table 11. Moreover, the *FNRT* results portray the least computational run executed by MOPGO and MOPVS relatively to reach the optimal solution. Table 12 unveils the Best-Compromised Solution (BCS) that satisfied each objective ($f_{weight}$, $f_{compliance}$) relying on fuzzy decision technique. It is evident from all tables that for a 10-bar truss problem, the BCS, i.e. (3653.518, 138077.3) achieved by MOPGO, is superior among all selected algorithms.

Figure 4 depicts the best Pareto fronts for individual algorithms and their corresponding BCS results. The combined Pareto fronts illustration describes the diverse, continuous, and smooth qualitative behavior of MOPGO relatively. The dominance of the proposed MOPGO algorithm quantitatively over others is also illustrated in Figure 5 comprehensively in the form of all investigated performance metric outcomes boxplots.

### 2) 25-BAR SPATIAL TRUSS

HV metric results are depicted in Table 2, and it displays that the best $f_{\text{mean}}$ value of 0.695684 and least $f_{\text{std}}$ value of 9.34E-05 is achieved by MOPGO. The *FNRT* values of MOPGO, MOPVS, MOSMA, MOSOS, and MOALO are 500, 200, 375, 325, and 100, respectively. Thus, MOPGO ranked first among all algorithms at 95% significance level and, hence, better solutions density near Pareto Front. In terms of GD metric $f_{\text{mean}}$ value, MOPGO obtained the least value of 4.902076 with a substantial percentage decrease of 95.16%, 58.95%, 33.58%, and 33.29% from MOALO, MOPVS, MOSOS, and MOSMA, respectively, as per Table 3. Moreover, MOPGO attain the best *FNRT* value of 100 with minimum $f_{\text{std}}$ that describes its improved convergence behavior. Table 4 shows the better CVG values for MOALO relatively while the least value of $f_{\text{std}}$ is procured by MOPGO. In the CPF measure, the MOSMA and MOPVS attest to their satisfactory result and acquire *FNRT* values of 450 and 425, respectively, followed by MOPGO, as listed in Table 5. The MOPGO obtains $f_{\text{mean}}$ value for DM metric as depicted in Table 6 has a percentage increase of 25.58%, 24.82%, 44.83%, and 16.46% from MOALO, MOSOS, MOSMA, and MOPVS, respectively. The *FNRT* results for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO are 500, 400, 100, 275, and 225, respectively. Thus, at a 95% significance level, the proposed MOPGO algorithm outperforms others and exhibits better diversity in solutions. Similarly, from Table 7 for PD measure, the MOPGO has a percentage increase of 41.94%, 38.87%, 38.55%, and 17.57% in $f_{\text{mean}}$ value from MOPVS, MOALO, MOSMA and MOSOS. Also, MOPGO obtains the maximum *FNRT* value of 500, followed by MOSOS with 325. These outcomes describe the improved pure diversity nature of investigated MOPGO over other methodologies.

Table 8 reveals the SP metric where MOPGO $f_{\text{mean}}$ value reported a substantial percentage decrease of 78.32%, 56.44%, 59.68%, and 85.70% from MOALO, MOSOS, MOSMA, and MOPVS. Similarly, MOPGO realize a major percentage decrease of 97.33%, 94.83%, and 70.03%, corresponding to MOALO, MOPVS, and MOSOS in terms of $f_{\text{std}}$ values. Moreover, MOPGO manifests the best *FNRT* value that describes its optimal spacing feature relatively. For SD indicator as per Table 9, MOPGO $f_{\text{mean}}$ value reported a percentage decrease of 76.78%, 75.89%, 64.32%, and 77.37% against MOALO, MOSOS, MOSMA, and MOPVS. MOPGO realize a best *FNRT* value of 100 at minimum $f_{\text{std}}$ value of 0.018069 that manifests it well-distributed non-dominated solutions relative to others. For IGD performance measure the MOPVS realize the best $f_{\text{mean}}$ and $f_{\text{std}}$ value along with the least *FNRT* value, which is followed by MOSMA as presented in Table 10. Regarding RT metric, the proposed MOPGO realize a minimum $f_{\text{mean}}$ value of 27.26802 relatively. Also, its $f_{\text{std}}$ results show a substantial percentage decrease of 96.425, 90.22%, 75.16%, and 35.16% from MOSMA, MOSOS, MOALO, and MOPVS algorithms. MOPGO, MOPVS, MOSMA, MOSOS, and MOALO realize the *FNRT* value of 100, 200, 350, 350, and 500, respectively. At a 95% significance level, MOPGO ranked first and manifested its least computational time to reach the optimal solution. Table 12 presents the BCS results for 25-bar 3D truss that proves that the best result is obtained by MOPGO, i.e. (2346.643, 35377.12), against all other considered algorithms.

The best Pareto fronts achieved by individual algorithms are indicated in Figure 7 simultaneously with their BCS results. The combined Pareto fronts are contrasted for qualitative analysis in the last plot, reflecting the continuous and well-distributed nature of MOPGO solutions. Furthermore, all ten performance metrics results by all considered are plotted in boxplot as illustrated in Figure 8 that demonstrates

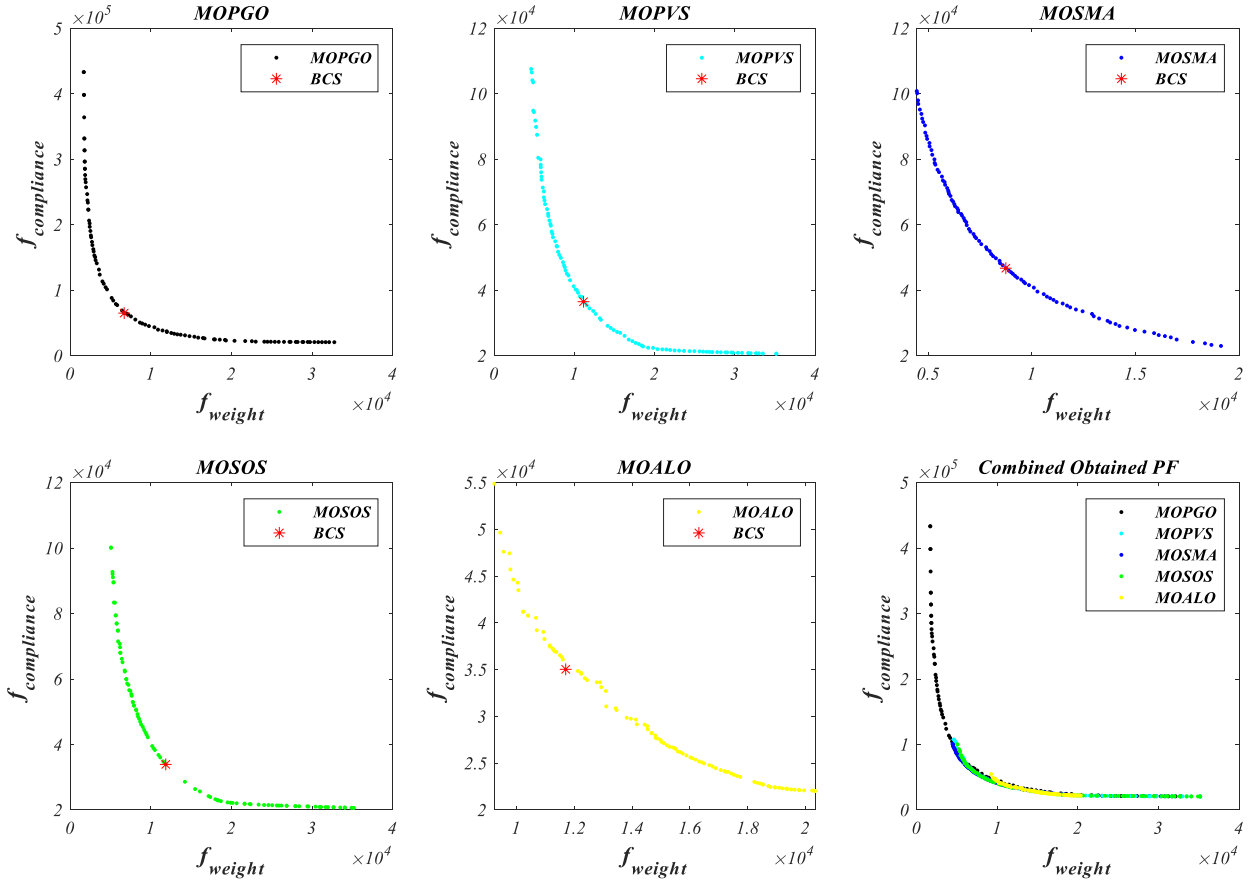


**FIGURE 16.** Best Pareto fronts of the 72-bar truss by all algorithms.

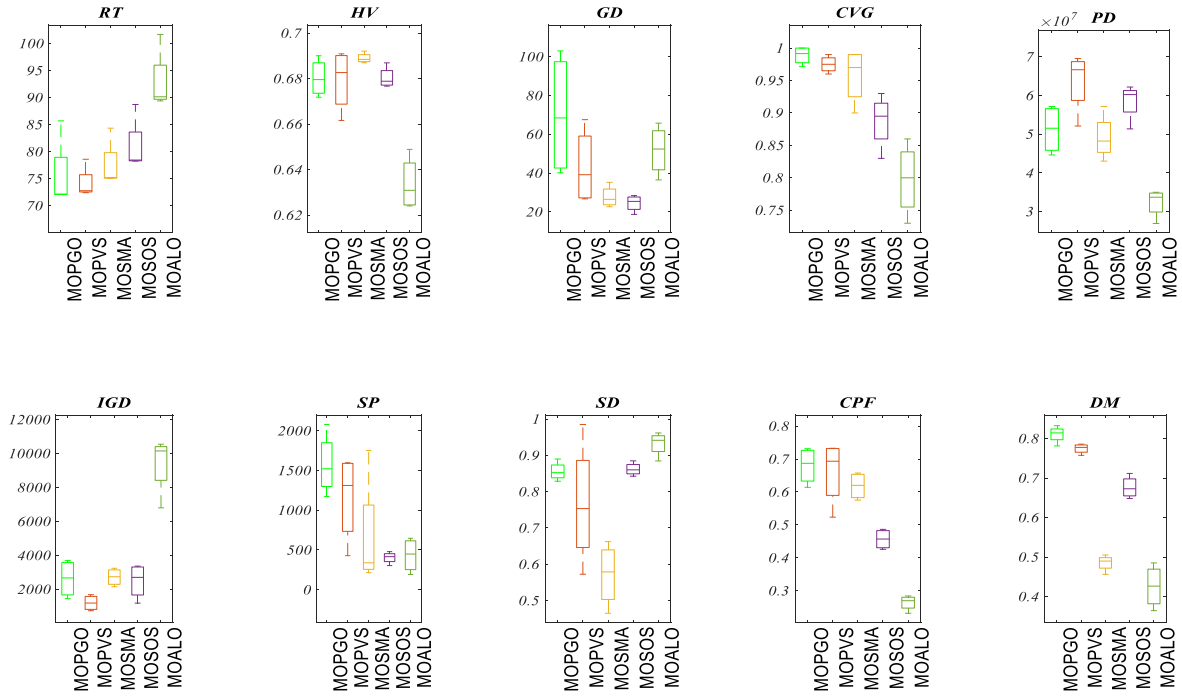


**FIGURE 17.** Boxplots of the 72-bar truss by all algorithms.

**TABLE 2.** Results of (*HVMetric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{\min}$ | 0.64049 | 0.635789 | 0.63682 | 0.629392 | 0.576638 | 0.661612 | 0.671809 | 0.686964 | 0.67668 | 0.624048 |
| $f_{\max}$ | 0.641752 | 0.637803 | 0.637995 | 0.636986 | 0.59193 | 0.690869 | 0.690029 | 0.69206 | 0.686912 | 0.648887 |
| $f_{\text{mean}}$ | 0.64093 | 0.636492 | 0.637423 | 0.632957 | 0.584954 | 0.679439 | 0.680238 | 0.688971 | 0.680314 | 0.633681 |
| $f_{\text{median}}$ | 0.640738 | 0.636187 | 0.637439 | 0.632724 | 0.585624 | 0.682637 | 0.679558 | 0.68843 | 0.678831 | 0.630894 |
| $f_{\text{std}}$ | 0.000574 | 0.000928 | 0.000489 | 0.003115 | 0.00635 | 0.013653 | 0.008253 | 0.002214 | 0.004633 | 0.011716 |
| *FNRT* | **500** | 300 | 375 | 225 | 100 | 325 | 325 | 450 | 300 | 100 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{\min}$ | 0.695559 | 0.69042 | 0.694026 | 0.691657 | 0.662183 | 0.551059 | 0.548183 | 0.551983 | 0.540452 | 0.4804 |
| $f_{\max}$ | 0.695756 | 0.691892 | 0.695325 | 0.694861 | 0.677605 | 0.55488 | 0.551334 | 0.554152 | 0.550227 | 0.509248 |
| $f_{\text{mean}}$ | 0.695684 | 0.691385 | 0.694866 | 0.693481 | 0.670763 | 0.553887 | 0.549824 | 0.552911 | 0.547638 | 0.496942 |
| $f_{\text{median}}$ | 0.695711 | 0.691614 | 0.695057 | 0.693704 | 0.671632 | 0.554805 | 0.54989 | 0.552756 | 0.549936 | 0.499059 |
| $f_{\text{std}}$ | 9.34E-05 | 0.000671 | 0.000582 | 0.001338 | 0.006561 | 0.001886 | 0.00129 | 0.000939 | 0.004793 | 0.014206 |
| *FNRT* | 500 | 200 | 375 | 325 | 100 | 450 | 250 | 425 | 275 | 100 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{\min}$ | 0.688941 | 0.700103 | 0.693727 | 0.695451 | 0.623971 | 0.749252 | 0.770153 | 0.728717 | 0.763731 | 0.682358 |
| $f_{\max}$ | 0.703895 | 0.70363 | 0.702253 | 0.701019 | 0.644789 | 0.780982 | 0.783622 | 0.755828 | 0.785712 | 0.746076 |
| $f_{\text{mean}}$ | 0.695012 | 0.701947 | 0.698944 | 0.698055 | 0.633491 | 0.767677 | 0.776494 | 0.747096 | 0.77449 | 0.715636 |
| $f_{\text{median}}$ | 0.693605 | 0.702027 | 0.699898 | 0.697875 | 0.632601 | 0.770237 | 0.776101 | 0.75192 | 0.774259 | 0.717055 |
| $f_{\text{std}}$ | 0.006854 | 0.001444 | 0.003663 | 0.002564 | 0.00987 | 0.013333 | 0.006613 | 0.012642 | 0.010383 | 0.02692 |
| *FNRT* | 275 | 450 | 375 | 300 | 100 | 400 | 425 | 200 | 375 | 100 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{\min}$ | 0.49697 | 0.413492 | 0.525034 | 0.507789 | 0.458044 | 0.692961 | 0.685825 | 0.679802 | 0.685604 | 0.680533 |
| $f_{\max}$ | 0.531959 | 0.507046 | 0.541363 | 0.526918 | 0.490493 | 0.729968 | 0.729637 | 0.68808 | 0.73427 | 0.702921 |
| $f_{\text{mean}}$ | 0.510049 | 0.474161 | 0.534752 | 0.516394 | 0.474378 | 0.711454 | 0.714961 | 0.684451 | 0.70832 | 0.694543 |
| $f_{\text{median}}$ | 0.505634 | 0.488053 | 0.536306 | 0.515435 | 0.474488 | 0.711443 | 0.722192 | 0.684961 | 0.706703 | 0.697358 |
| $f_{\text{std}}$ | 0.015396 | 0.042709 | 0.007124 | 0.008971 | 0.013462 | 0.015364 | 0.019739 | 0.003446 | 0.021693 | 0.009714 |
| *FNRT* | 325 | 150 | 500 | 375 | 150 | 425 | 375 | 125 | 325 | 250 |
| *Average FNRT* | 400 | 309.375 | 353.125 | 312.5 | 125 | 400 | 309.375 | 353.125 | 312.5 | 125 |

the dominance of the suggested MOPGO algorithm quantitatively over others.

### 3) 37-BAR PLANAR TRUSS EXAMPLE

The HV metric results are illustrated in Table 2, from which it is evident that the MOPVS achieved better $f_{\text{mean}}$ and $f_{\text{std}}$ value relative to other considered algorithms. Moreover, the *FNRT* metric assigned the rank of 275, 450, 375, 300, and 100 to the MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms. At a 95% significance level, MOPGO demonstrating its improved solution density. For GD indicator $f_{\text{mean}}$ results for MOPGO from Table 3 show a percentage decrease of 97.74%, 52.45%, 20.69%, and 6.16%% regarding MOALO, MOPVS, MOSMA, and MOSOS, respectively. However, the MOSOS obtain the least $f_{\text{std}}$ value followed by MOPGO that value shows a significant percentage decrease of 98.18% from MOALO. The proposed MOPGO also found a least *FNRT* value of 150, followed by MOSOS, MOSMA. Therefore, MOPGO has a better quality of convergence as per *FNRT* at 95% significance level. As per the CVG metrics listed in Table 4, the best $f_{\text{mean}}$ value was obtained by MOSOS and stood first as per *FNRT* while the proposed MOPGO technique achieves the second-best rank and demonstrates its enhanced coverage characteristic. In terms of CPF metric as reported in Table 5, MOPGO demonstrates its improved quality relatively and settled at the highest *FNRT* value of 475. Moreover, MOPGO realize a percentage increase of 143.98%, 63.54%, 14.34%, and 9.37% in $f_{\text{mean}}$ value relative to MOALO, MOSOS, MOSMA, and MOPVS, respectively. Table 6 depicts the DM metric results according to which MOPGO $f_{\text{mean}}$ value has a percentage increase of 64.86%, 20.58%, 51.82%, and 10.65% from MOALO, MOSOS, MOSMA, and MOPVS and the least $f_{\text{std}}$ values. MOPGO, as per the *FNRT* obtained a maximum 500 value relatively and at 95% significance level ranked first. These prospects exhibit the improved solution diversity of MOPGO concerning other contrasted methodology.

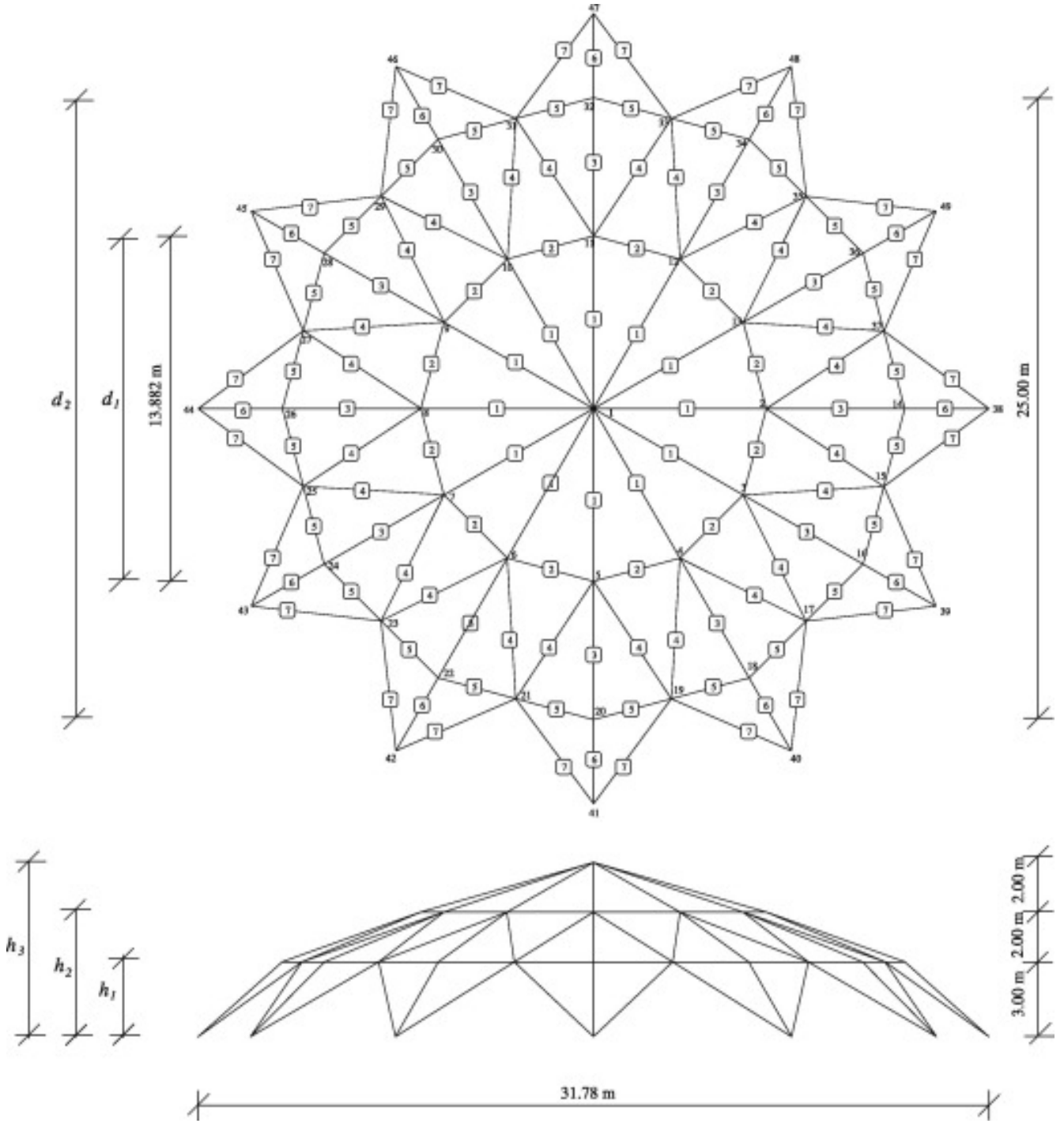


**FIGURE 18.** The 120-bar 3D truss.

For PD performance measure as illustrated in Table 7, the MOPGO, MOPVS, and MOSOS algorithm demonstrates their superior pure diversity behavior through its best $f_{\text{mean}}$ and $f_{\text{std}}$ values that were eventually proved by their highest 400 *FNRT* value. For SP measure, according to Table 8, MOPGO evidence a significant percentage decrease in its $f_{\text{mean}}$ value of 78.06%, 47.10%, and 46.84% relative to MOPVS, MOSOS, and MOSMA, respectively. MOPGO also attain the minimum $f_{\text{std}}$ value after MOALO and MOSMA. Furthermore, the *FNRT* metric indicates the best rank of 150 by MOPGO and, thus, at a 95% significance level, displays its enhanced spacing quality. From SD metric findings as shown in Table 9, MOPGO finds the superior $f_{\text{mean}}$ value of 0.523486 and minimum $f_{\text{std}}$ 0.049119 pertaining to other

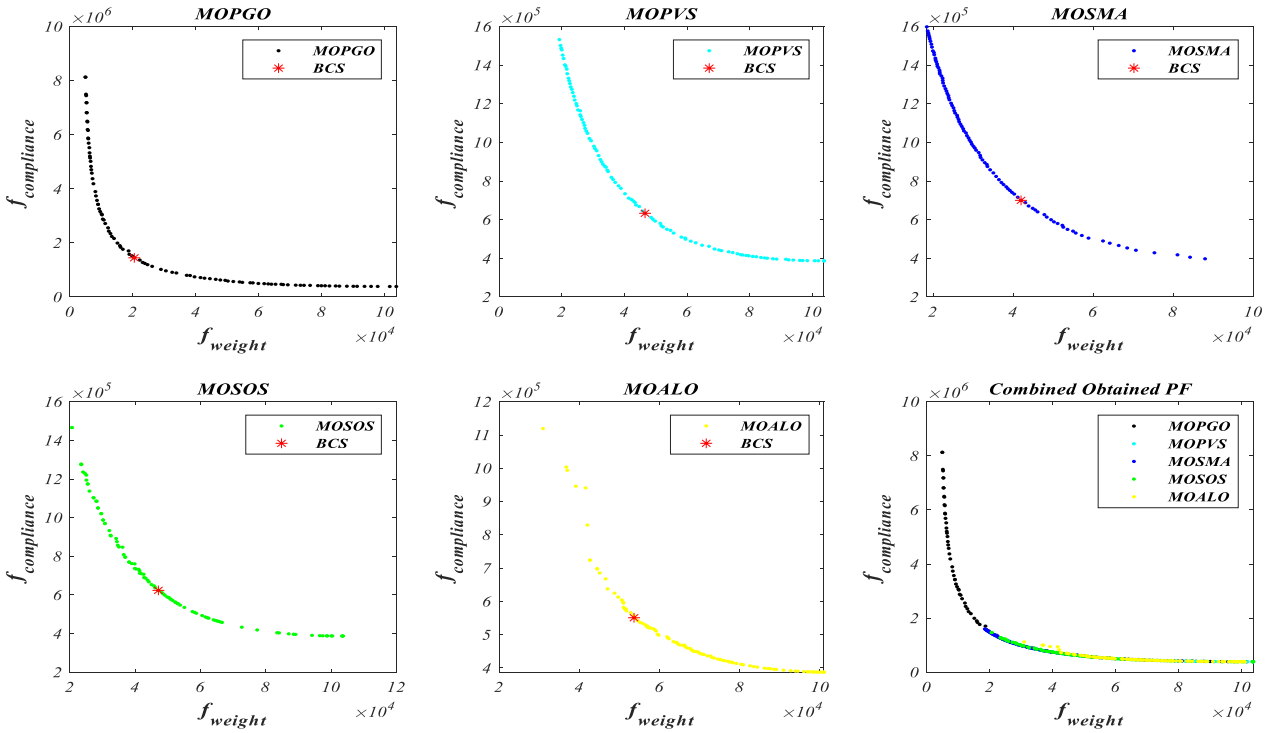


**FIGURE 19.** Best Pareto fronts of the 120-bar truss by all algorithms.

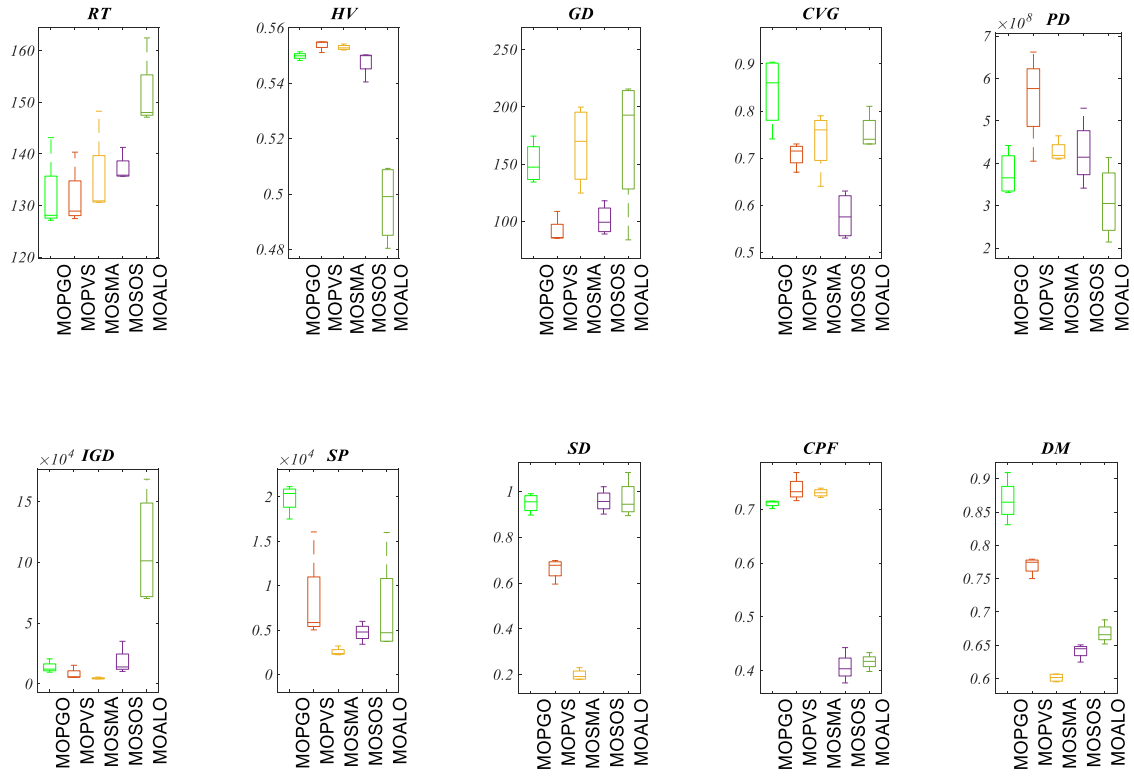


**FIGURE 20.** Boxplots of the 120-bar truss by all algorithms.

algorithms and achieve the best *FNRT* value of 100 followed by MOSMA. At a 95% significance level, MOPGO displays its well-distributed non-dominated solutions. In the IGD test, as indicated in Table 10, MOPGO manifests a percentage

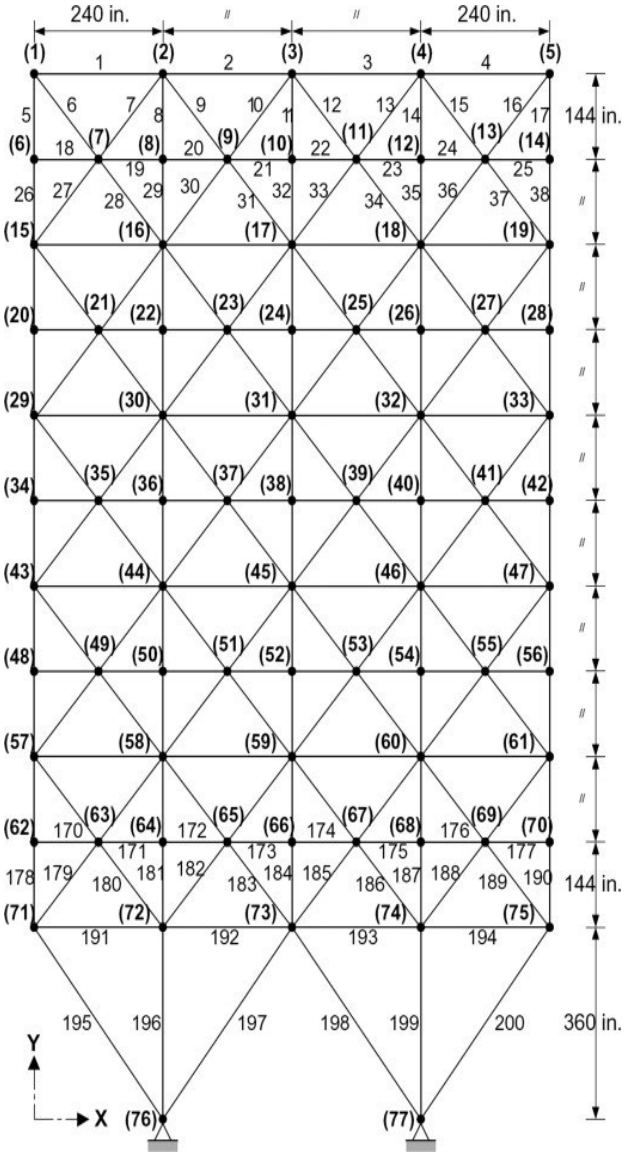


**FIGURE 21.** The 200-bar 3D truss.

decrease of 91.27%, 50.23%, 39.22%, and 17.91% regarding MOALO, MOSMA, MOPVS, and MOSOS, respectively. Also $f_{\text{mean}}$ value of MOPGO reported a significant percentage decrease of 93.25%, 86.28%, 75.03%, and 73.83% from MOALO, MOPVS, MOSMA, and MOSOS, respectively. MOPGO manifested a superior *FNRT* result of 175 and ranked first at a 95% significance level showing its competing convergence-spread parity attribute. In terms of RT measure, MOPGO realize the best $f_{\text{mean}}$ value while the least $f_{\text{std}}$ value is an exhibit by MOSOS followed by MOPGO, as shown in Table 11. Moreover, the *FNRT* results portray the least computational run executed by MOPGO relatively to reach the optimal solution. Table 12 unveils the superior BCS (1180.289, 18411.19) achieved by MOPGO among all executed algorithms.

Figure 10 depicts the best Pareto fronts for individual algorithms and their corresponding BCS results. The combined Pareto fronts illustration describes the diverse, continuous, and smooth qualitative behavior of MOPGO relatively. The dominance of the proposed MOPGO technique quantitatively over others is also illustrated in Figure 11 comprehensively in the form of all investigated performance metric outcomes boxplots.

#### 4) 60-BAR SPATIAL TRUSS

Table 2 displays HV measure results accordingly the best $f_{\text{mean}}$ and $f_{\text{std}}$ value is obtained by MOSMA. The *FNRT* values for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms are 325, 150, 500, 375, and 150, respectively. Thus, MOPGO at 95% significance level exhibit an acceptable solution density near Pareto Front. In terms of GD metric, MOPGO obtained $f_{\text{mean}}$ value of 21.79952, which is 99.97% and 66.17% less from MOALO, and MOPVS respectively, as per Table 3. Moreover, MOPGO attain the second-best *FNRT* values of 200 after MOSOS with minimum $f_{\text{std}}$ result that describes its improved convergence behavior relatively. Table 4 shows the best CVG $f_{\text{mean}}$ and *FNRT* values of 0.732292 and 100, respectively, for MOPGO that governs its improved coverage quality relatively. In the CPF measure, the MOPVS and MOSMA attest to their satisfactory result and acquire *FNRT* values of 450 and 400, respectively, accompanied by MOPGO, as illustrated in Table 5. MOPGO $f_{\text{mean}}$ value for DM metric as depicted in Table 6 has a percentage increase of 59.70% and 37.13% from MOALO, and MOSMA respectively. The *FNRT* results for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO are 400, 425, 175, 375, and 125, respectively. Thus, at a 95% significance level, the proposed MOPGO algorithm exhibit better diversity in solutions.

Table 7 describes the PD measure results according to which the MOSMA finds the best $f_{\text{mean}}$ value while MOSOS realize the better $f_{\text{std}}$ result out of all considered optimization techniques.

As per *FNRT* results, the maximum 400 value is achieved by MOSOS, succeeded by MOPGO and MOSMA, both of which attain the same 375 value. At a 95% significance level, the outcomes illustrate the improved pure diversity nature of investigated MOPGO over other methodologies. Table 8 reveals the SP metric results where MOPGO $f_{\text{mean}}$ value reported a percentage decrease of 9.87%, 7.46%, 43.16%, and 73.42% from MOALO, MOSOS, MOSMA, and MOPVS. Similarly, MOPGO realize a major percentage decrease of 75.99% and 30.60% corresponding to MOPVS, MOSMA in terms of $f_{\text{std}}$ results. Moreover, MOPGO, MOSOS, MOALO manifests the best *FNRT* result that describes its optimal spacing feature relatively. For SD indicator as per Table 9, MOPGO $f_{\text{mean}}$ value reported a percentage decrease of 38.83%, 36.68%, 30.84%, and 37.21% against MOALO, MOSOS, MOSMA, and MOPVS. MOPGO realize a best *FNRT* value of 100 at minimum $f_{\text{std}}$ value that manifests it well-distributed non-dominated solutions relative to others. From IGD performance measure as depicted in Table 10, the MOSMA realize the

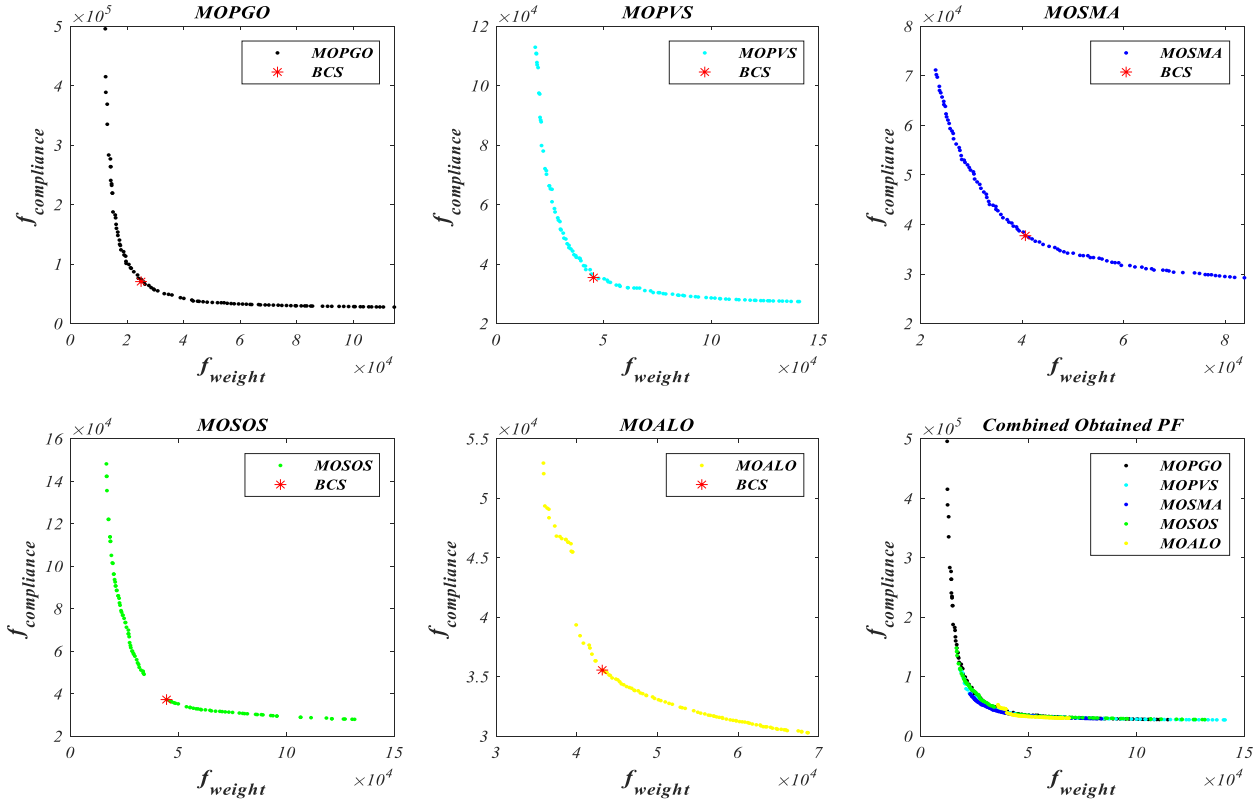


**FIGURE 22.** Best Pareto fronts of the 200-bar truss by all algorithms.

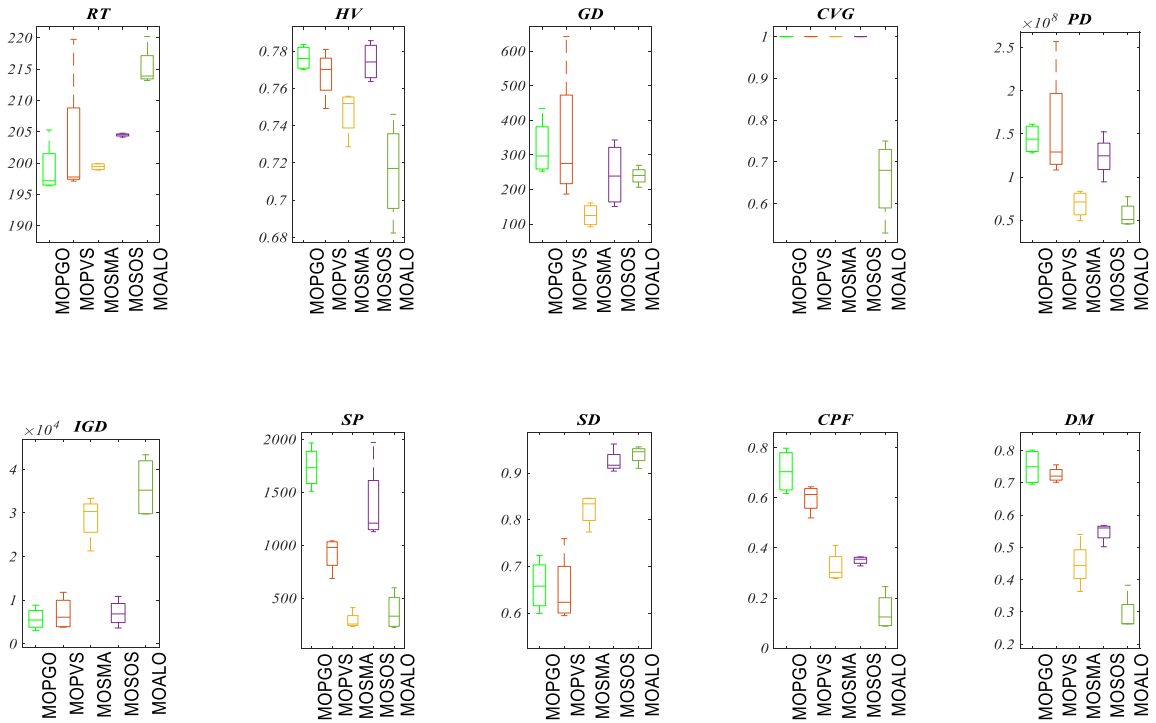


**FIGURE 23.** Boxplots of the 200-bar truss by all algorithms.

best $f_{\text{mean}}$ value of 1757.774 while the least $f_{\text{std}}$ value is obtained by MOSOS. The least *FNRT* value is manifested by MOSMA, which is followed by MOSOS. Regarding RT metric, the proposed MOPGO realize a minimum $f_{\text{mean}}$

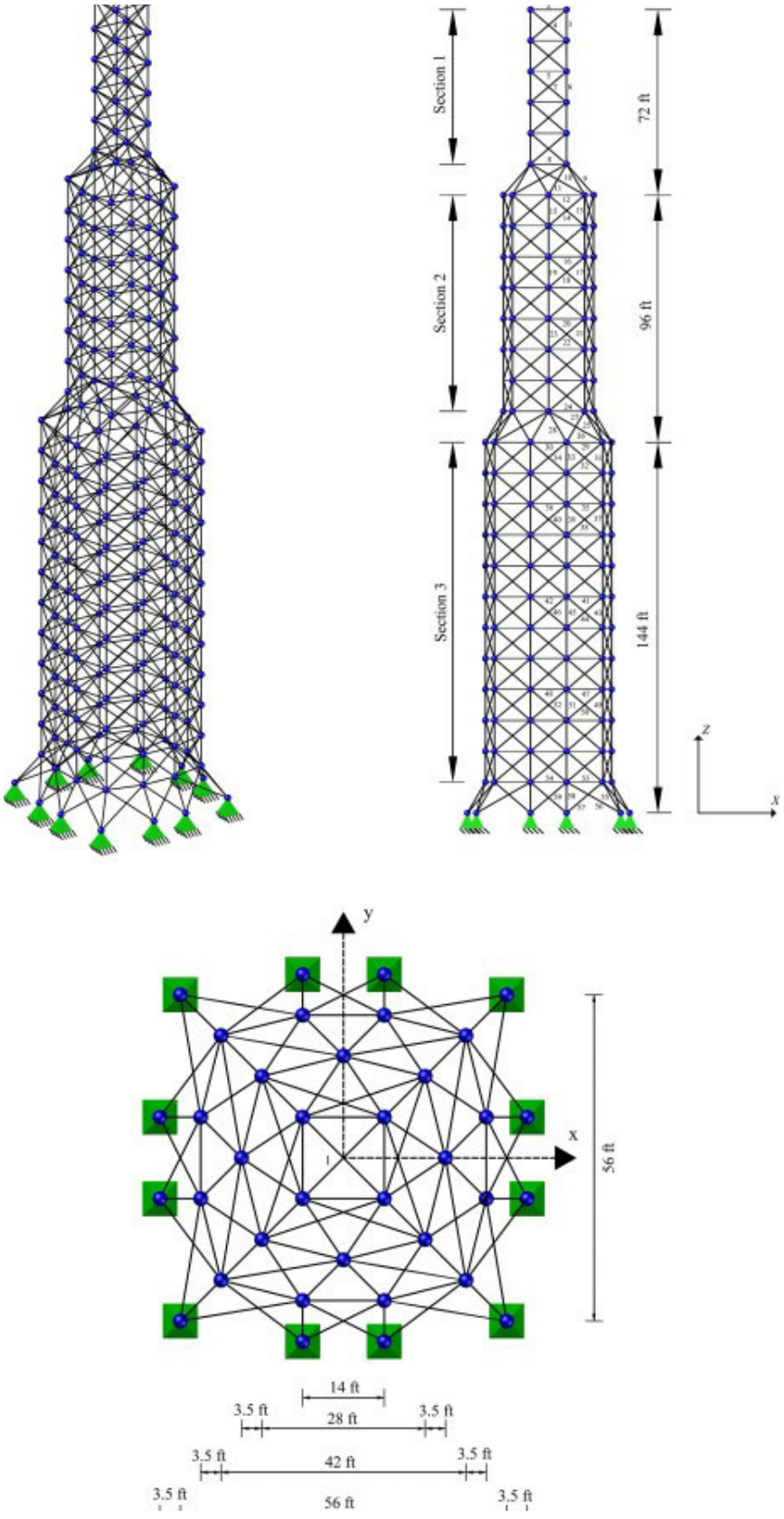


**FIGURE 24.** The 942-bar tower truss.

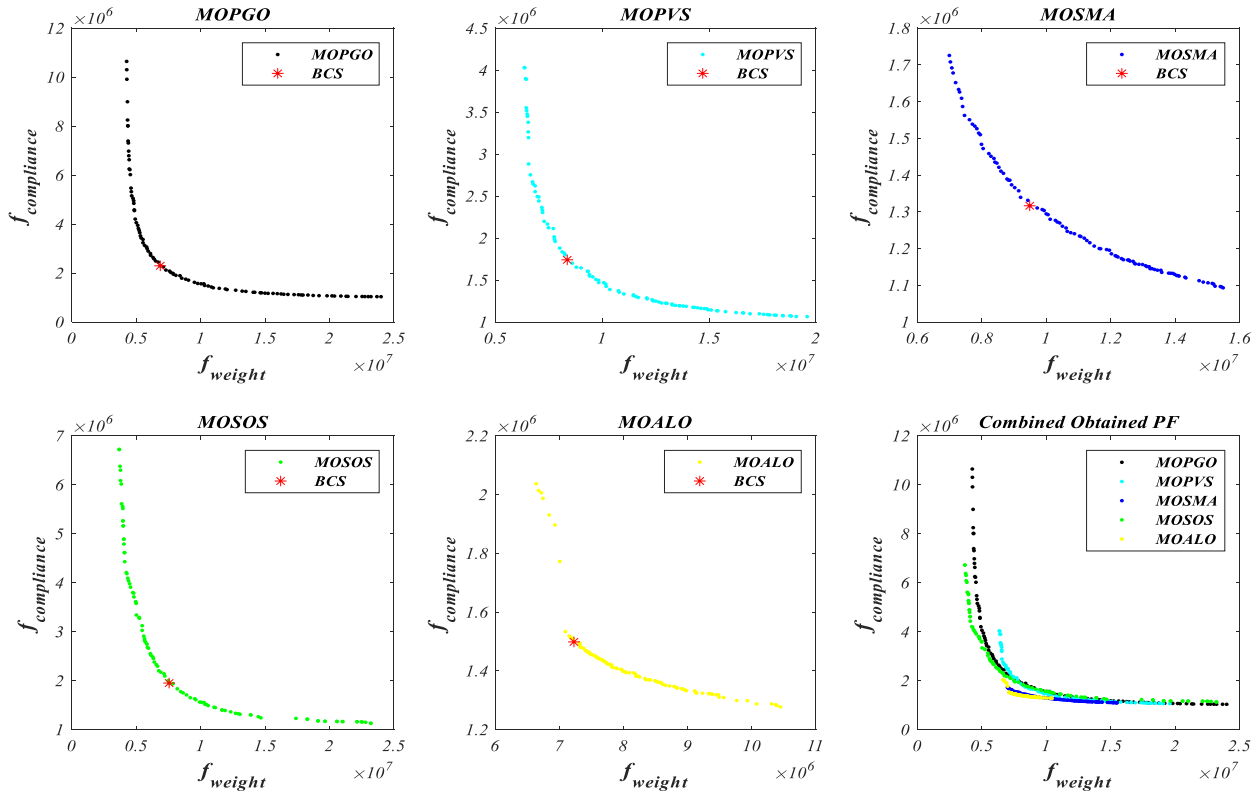


**FIGURE 25.** Best Pareto fronts of the 942-bar truss by all algorithms.

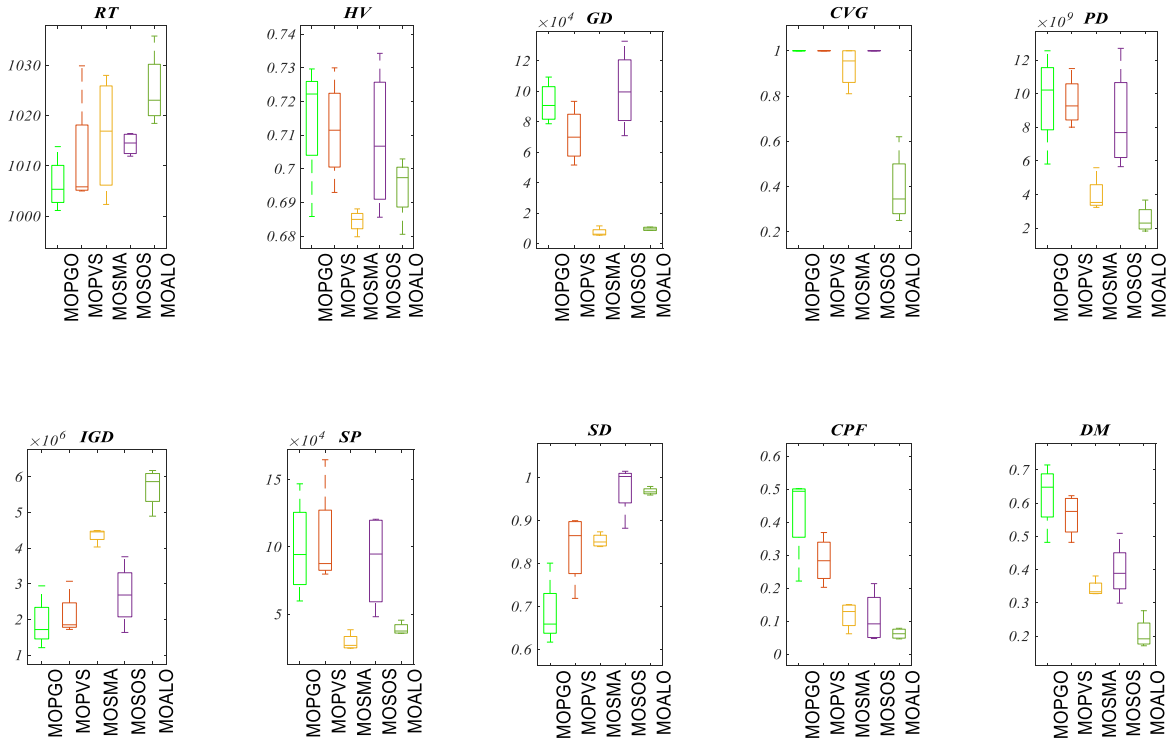


**FIGURE 26.** Boxplots of the 942-bar truss by all algorithms.

value of 65.45753 relatively. MOPGO, MOPVS, MOSMA, MOSOS, and MOALO realize the *FNRT* value of 125, 225, 250, 400, and 500, respectively. At a 95% significance level, MOPGO ranked first and manifested its least computational

**TABLE 3.** Results of (*GD-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{\min}$ | 12.29115 | 20.24102 | 14.34143 | 20.83741 | 18.68845 | 40.14802 | 26.70989 | 22.66492 | 36.47709 | 18.68845 |
| $f_{\max}$ | 15.02235 | 25.00747 | 19.44657 | 29.60009 | 28.3801 | 103.0239 | 67.5477 | 35.20639 | 65.7407 | 28.3801 |
| $f_{\text{mean}}$ | 13.29241 | 23.01133 | 16.34385 | 23.42731 | 24.46313 | 70.01614 | 43.1448 | 27.68917 | 51.75758 | 24.46313 |
| $f_{\text{median}}$ | 12.92806 | 23.39843 | 15.7937 | 21.63587 | 25.39199 | 68.44634 | 39.1608 | 26.44267 | 52.40626 | 25.39199 |
| $f_{\text{std}}$ | 1.293634 | 2.095063 | 2.417821 | 4.141751 | 4.28684 | 32.04997 | 19.6668 | 5.524895 | 12.78459 | 4.28684 |
| *FNRT* | 100 | 350 | 200 | 350 | 125 | 400 | 300 | 250 | 425 | 125 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{\min}$ | 3.721396 | 10.65518 | 6.646613 | 6.784546 | 88.99381 | 134.3691 | 85.22438 | 124.7395 | 83.89036 | 88.99381 |
| $f_{\max}$ | 5.903043 | 13.19963 | 8.140516 | 7.895379 | 117.9918 | 174.4681 | 108.7311 | 199.8003 | 215.5778 | 117.9918 |
| $f_{\text{mean}}$ | 4.902076 | 11.9419 | 7.348914 | 7.380763 | 101.3713 | 150.8986 | 91.44735 | 166.0922 | 171.2445 | 101.3713 |
| $f_{\text{median}}$ | 4.991932 | 11.9564 | 7.304264 | 7.421564 | 99.2498 | 147.3787 | 85.91695 | 169.9146 | 192.755 | 99.2498 |
| $f_{\text{std}}$ | 0.967095 | 1.077262 | 0.616918 | 0.459146 | 13.0704 | 18.26553 | 11.52876 | 35.39223 | 61.44483 | 13.0704 |
| *FNRT* | 100 | 425 | 250 | 250 | 200 | 375 | 150 | 375 | 400 | 200 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{\min}$ | 4.236549 | 9.31844 | 5.409434 | 5.172688 | 150.5926 | 253.0394 | 186.5421 | 91.84767 | 206.397 | 150.5926 |
| $f_{\max}$ | 7.889606 | 13.31837 | 8.874736 | 6.890034 | 343.1443 | 434.2124 | 642.099 | 160.936 | 269.7945 | 343.1443 |
| $f_{\text{mean}}$ | 5.479135 | 11.52478 | 6.908521 | 5.83916 | 242.7585 | 320.1545 | 344.7113 | 125.5405 | 239.2806 | 242.7585 |
| $f_{\text{median}}$ | 4.895192 | 11.73116 | 6.674958 | 5.64696 | 238.6486 | 296.6831 | 275.1021 | 124.6891 | 240.4655 | 238.6486 |
| $f_{\text{std}}$ | 1.690573 | 1.898367 | 1.486709 | 0.748842 | 93.38359 | 82.81278 | 203.9372 | 32.66148 | 26.10161 | 93.38359 |
| *FNRT* | 150 | 400 | 275 | 175 | 300 | 425 | 375 | 125 | 275 | 300 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{\min}$ | 13.53725 | 54.18284 | 21.142 | 13.47743 | 70880.97 | 78697.99 | 51587.67 | 5588.081 | 8653.43 | 70880.97 |
| $f_{\max}$ | 26.91212 | 83.17845 | 30.1375 | 17.75696 | 132887.2 | 109347.6 | 93407.78 | 11631.69 | 10874.74 | 132887.2 |
| $f_{\text{mean}}$ | 21.79952 | 64.4468 | 25.44594 | 15.80612 | 100741.2 | 92359.86 | 71189.21 | 7388.747 | 9691.529 | 100741.2 |
| $f_{\text{median}}$ | 23.37436 | 60.21295 | 25.25213 | 15.99505 | 99598.32 | 90696.91 | 69880.7 | 6167.609 | 9618.973 | 99598.32 |
| $f_{\text{std}}$ | 5.75814 | 13.20134 | 3.876235 | 1.816843 | 26355.12 | 13588.55 | 17961.2 | 2860.339 | 1204.221 | 26355.12 |
| *FNRT* | 200 | 500 | 275 | 125 | 475 | 425 | 300 | 125 | 175 | 475 |
| *Average FNRT* | 206.25 | 412.5 | 265.625 | 221.875 | 393.75 | 206.25 | 412.5 | 265.625 | 221.875 | 393.75 |

time to reach the optimal solution. Table 12 presents the BCS results for a 60-bar 3D truss that proves that the best result is obtained by MOPGO, i.e. (1955.053, 81300.43) against all other considered algorithms. The best Pareto fronts achieved by individual algorithms are indicated in Figure 13 simultaneously with their BCS results. The combined Pareto fronts are contrasted for qualitative analysis in the last plot, reflecting the continuous and well-distributed nature of MOPGO solutions. Furthermore, all ten performance metrics results by all considered are plotted in boxplot as illustrated in Figure 14 that demonstrates the dominance of the suggested MOPGO algorithm quantitatively over others.

### 5) 72-BAR SPATIAL TRUSS

The HV metric results are illustrated in Table 2, from which it is evident that the MOSMA achieved the best $f_{\text{mean}}$ and $f_{\text{std}}$ value relative to other considered algorithms. Moreover, the *FNRT* metric assigned the rank of 325, 325, 450, 300, and 100 to the MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms. Hence, at*FNRT* 95% significance level, MOSMA and MOPGO expressed their high solution density in the proximity of the Pareto Front. For GD indicator, MOALO exhibit its dominance by obtaining the best $f_{\text{mean}}$ and $f_{\text{std}}$ value along with the best*FNRT* value of 125 as depicted in Table 3. As per the CVG metric listed in Table 4, MOPGO obtained the best $f_{\text{mean}}$ value shows a percentage decrease of 10.14%, 16.71%, 18.20%, and 19.34% from MOALO, MOSOS, MOSMA, and MOPVS. The *FNRT* results obtained by MOPGO, MOPVS, MOSMA, MOSOS, and MOALO are 125, 425, 400, 350, and 200, respectively. At a 95% significance level, MOPGO ranked first and illustrated its enhanced coverage characteristic among all. The $f_{\text{mean}}$ results of CPF metric for MOPGO algorithm exhibits a percentage increase of 157.76%, 48.89%, 9.97%, and 2.89% from MOALO, MOSOS, MOSMA, and MOPVS, respectively, as illustrated in Table 5. Moreover, the *FNRT* values of MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms are equal to 425, 425, 350, 200, and 100, respectively.

**TABLE 4.** Results of (*CVG-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{min}$ | 0.82 | 0.83871 | 0.68 | 0.61 | 0.38 | 0.73 | 0.971831 | 0.96 | 0.9 | 0.83 |
| $f_{max}$ | 0.87 | 0.912281 | 0.87 | 0.83 | 0.6 | 0.86 | 1 | 0.99 | 0.99 | 0.93 |
| $f_{mean}$ | 0.8375 | 0.884848 | 0.765 | 0.75 | 0.52 | 0.7975 | 0.98872 | 0.975 | 0.9575 | 0.8875 |
| $f_{median}$ | 0.83 | 0.894201 | 0.755 | 0.78 | 0.55 | 0.8 | 0.991525 | 0.975 | 0.97 | 0.895 |
| $f_{std}$ | 0.023629 | 0.034155 | 0.078528 | 0.101653 | 0.096609 | 0.055603 | 0.013806 | 0.01291 | 0.04272 | 0.041932 |
| *FNRT* | 375 | 475 | 300 | 250 | 100 | 125 | 425 | 400 | 350 | 200 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{min}$ | 0.83 | 0.848485 | 0.63 | 0.62 | 0.5 | 0.73 | 0.740741 | 0.67 | 0.64 | 0.53 |
| $f_{max}$ | 0.87 | 0.968254 | 0.7 | 0.69 | 0.59 | 0.81 | 0.903846 | 0.73 | 0.79 | 0.63 |
| $f_{mean}$ | 0.8575 | 0.919553 | 0.675 | 0.65 | 0.5475 | 0.755 | 0.841147 | 0.7075 | 0.7375 | 0.5775 |
| $f_{median}$ | 0.865 | 0.930737 | 0.685 | 0.645 | 0.55 | 0.74 | 0.86 | 0.715 | 0.76 | 0.575 |
| $f_{std}$ | 0.01893 | 0.054381 | 0.033166 | 0.03559 | 0.036856 | 0.037859 | 0.077295 | 0.0263 | 0.06702 | 0.049917 |
| *FNRT* | 425 | 475 | 287.5 | 212.5 | 100 | 337.5 | 475 | 237.5 | 350 | 100 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{min}$ | 0.87 | 0.932203 | 0.85 | 0.85 | 0.87 | 0.53 | 1 | 1 | 1 | 1 |
| $f_{max}$ | 0.96 | 1 | 0.98 | 0.93 | 0.96 | 0.75 | 1 | 1 | 1 | 1 |
| $f_{mean}$ | 0.9175 | 0.962685 | 0.9375 | 0.88 | 0.9225 | 0.66 | 1 | 1 | 1 | 1 |
| $f_{median}$ | 0.92 | 0.959267 | 0.96 | 0.87 | 0.93 | 0.68 | 1 | 1 | 1 | 1 |
| $f_{std}$ | 0.036856 | 0.03404 | 0.05909 | 0.038297 | 0.04113 | 0.095917 | 0 | 0 | 0 | 0 |
| *FNRT* | 250 | 450 | 350 | 162.5 | 287.5 | 100 | 350 | 350 | 350 | 350 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{min}$ | 0.68 | 1 | 1 | 0.96 | 0.96 | 0.25 | 1 | 1 | 0.81 | 1 |
| $f_{max}$ | 0.79 | 1 | 1 | 1 | 1 | 0.62 | 1 | 1 | 1 | 1 |
| $f_{mean}$ | 0.732292 | 1 | 1 | 0.985 | 0.99 | 0.39 | 1 | 1 | 0.93 | 1 |
| $f_{median}$ | 0.729583 | 1 | 1 | 0.99 | 1 | 0.345 | 1 | 1 | 0.955 | 1 |
| $f_{std}$ | 0.045017 | 0 | 0 | 0.019149 | 0.02 | 0.162275 | 0 | 0 | 0.090554 | 0 |
| *FNRT* | 100 | 387.5 | 387.5 | 300 | 325 | 100 | 375 | 375 | 275 | 375 |
| *Average FNRT* | 226.5625 | 426.5625 | 335.9375 | 281.25 | 229.6875 | 226.5625 | 426.5625 | 335.9375 | 281.25 | 229.6875 |

These outcomes manifest MOPGO enhanced coverage over Pareto Front characteristics concerning other investigated algorithms. Table 6 depicts the DM metric results according to which MOPGO $f_{mean}$ value has a percentage increase of 90.45%, 19.85%, 67.06%, and 4.65% from MOALO, MOSOS, MOSMA, and MOPVS at minimum $f_{std}$ value. As per the *FNRT*, MOPGO obtained a maximum value of 500 relatively, and at 95% significance level ranked first. These outcomes exhibit the improved solution diversity of MOPGO concerning other contrasted methodology.

Similarly, for PD performance measure as illustrated in Table 7, the MOPGO algorithm reported a substantial percentage increase of 97.14%, 29.65%, and 24,45% in $f_{mean}$ value with respect to MOALO, MOSMA, and MOPVS. MOPGO algorithm demonstrates its superior pure diversity behavior through its best $f_{mean}$ and $f_{std}$ that was eventually proved by its highest 450 *FNRT* value in comparison to other selected optimization techniques. For SP measure according to Table 8, MOPGO and MOALO evidence an acceptable value of $f_{mean}$ and $f_{std}$ among all contrasted methodologies. Furthermore, the *FNRT* metric indicates the best rank of 225 by MOPGO and thus, at 95% significance level, displays its enhanced spacing quality. From SD metric findings as shown in Table 9, MOPGO finds the superior $f_{mean}$ value that shows a percentage decrease of 38.71%, 33.71%, 25.42%, and 33.24% from MOALO, MOSOS, MOSMA, and MOPVS. Also, MOPGO finds the better $f_{std}$ value pertaining to other algorithms and achieve the best *FNRT* value of 100 followed by MOSMA at 95% significance level, MOPGO displays its well-distributed non-dominated solutions. In the IGD test, as indicated in Table 10, the MOPVS and MOPGO manifest superior $f_{mean}$, $f_{std}$ and *FNRT* values were showing its competing convergence-spread parity attribute comparatively. In terms of RT measure, the MOPVS realize the best $f_{mean}$ and $f_{std}$ value followed by MOPGO as shown in Table 11. Moreover, the *FNRT* results portray the least computational time executed by MOPGO and MOPVS relatively to reach the optimal solution. Table 12 unveils the

**TABLE 5.** Results of (*CPF-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{min}$ | 0.641969 | 0.72494 | 0.71165 | 0.398605 | 0.368187 | 0.613982 | 0.523426 | 0.575082 | 0.425867 | 0.231503 |
| $f_{max}$ | 0.726772 | 0.807655 | 0.740975 | 0.543499 | 0.496064 | 0.731293 | 0.732639 | 0.657412 | 0.486357 | 0.284085 |
| $f_{mean}$ | 0.694527 | 0.758201 | 0.726616 | 0.456103 | 0.426191 | 0.679746 | 0.660644 | 0.618115 | 0.456525 | 0.263712 |
| $f_{median}$ | 0.704683 | 0.750106 | 0.726919 | 0.441153 | 0.420256 | 0.686854 | 0.693257 | 0.619982 | 0.456937 | 0.26963 |
| $f_{std}$ | 0.039562 | 0.038649 | 0.012341 | 0.066901 | 0.056313 | 0.056285 | 0.098276 | 0.041519 | 0.030417 | 0.023187 |
| *FNRT* | 325 | 475 | 400 | 175 | 125 | 425 | 425 | 350 | 200 | 100 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{min}$ | 0.733846 | 0.790876 | 0.795398 | 0.451823 | 0.515136 | 0.701273 | 0.716137 | 0.722061 | 0.376829 | 0.398076 |
| $f_{max}$ | 0.792987 | 0.822968 | 0.823768 | 0.537116 | 0.554071 | 0.715219 | 0.768422 | 0.739315 | 0.44257 | 0.433238 |
| $f_{mean}$ | 0.773838 | 0.807236 | 0.810165 | 0.495303 | 0.531736 | 0.710896 | 0.737488 | 0.730748 | 0.40645 | 0.416306 |
| $f_{median}$ | 0.784259 | 0.807549 | 0.810747 | 0.496137 | 0.528869 | 0.713547 | 0.732697 | 0.730808 | 0.403201 | 0.416955 |
| $f_{std}$ | 0.02717 | 0.014629 | 0.011702 | 0.044885 | 0.019715 | 0.006523 | 0.022164 | 0.007431 | 0.027103 | 0.014383 |
| *FNRT* | 325 | 425 | 450 | 150 | 150 | 300 | 475 | 425 | 125 | 175 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{min}$ | 0.709244 | 0.676191 | 0.627548 | 0.432487 | 0.288071 | 0.617241 | 0.520146 | 0.278424 | 0.328441 | 0.087688 |
| $f_{max}$ | 0.83038 | 0.753857 | 0.724174 | 0.490879 | 0.339872 | 0.798382 | 0.644281 | 0.410283 | 0.364408 | 0.245535 |
| $f_{mean}$ | 0.774506 | 0.708115 | 0.677357 | 0.473575 | 0.317438 | 0.706224 | 0.597845 | 0.323517 | 0.350783 | 0.145552 |
| $f_{median}$ | 0.7792 | 0.701206 | 0.678853 | 0.485467 | 0.320906 | 0.704636 | 0.613477 | 0.302681 | 0.355142 | 0.124493 |
| $f_{std}$ | 0.049801 | 0.034179 | 0.043273 | 0.027528 | 0.021584 | 0.087986 | 0.055477 | 0.060853 | 0.016311 | 0.073828 |
| *FNRT* | 475 | 400 | 325 | 200 | 100 | 500 | 400 | 225 | 275 | 100 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{min}$ | 0.355249 | 0.419451 | 0.416612 | 0.347589 | 0.205133 | 0.221358 | 0.202351 | 0.06168 | 0.047292 | 0.04614 |
| $f_{max}$ | 0.483049 | 0.601221 | 0.604649 | 0.398281 | 0.291691 | 0.5012 | 0.368385 | 0.150211 | 0.213665 | 0.078325 |
| $f_{mean}$ | 0.442609 | 0.492879 | 0.502539 | 0.371738 | 0.244191 | 0.427331 | 0.284179 | 0.117496 | 0.111105 | 0.062012 |
| $f_{median}$ | 0.466068 | 0.475422 | 0.494447 | 0.370541 | 0.23997 | 0.493383 | 0.282991 | 0.129046 | 0.091732 | 0.061791 |
| $f_{std}$ | 0.060037 | 0.076934 | 0.080582 | 0.025207 | 0.040528 | 0.137467 | 0.071258 | 0.041123 | 0.078014 | 0.015913 |
| *FNRT* | 325 | 450 | 400 | 225 | 100 | 475 | 400 | 250 | 225 | 150 |
| *Average FNRT* | 393.75 | 431.25 | 353.125 | 196.875 | 125 | 393.75 | 431.25 | 353.125 | 196.875 | 125 |

best-compromised solution (BCS) that satisfied each objective ($f_{weight}, f_{compliance}$) relying on fuzzy decision technique. It is evident from the table that for a 72-bar truss, the superior BCS, i.e. (6687.743, 65199.19), is achieved by MOPGO among all executed algorithms.

Figure 16 depicts the best Pareto fronts for individual algorithms and their corresponding BCS results. The combined Pareto fronts illustration describes the diverse, continuous, and smooth qualitative behavior of MOPGO relatively. The dominance of the proposed MOPGO technique quantitatively over others is also illustrated in Figure 17 comprehensively in the form of all investigated performance metric outcomes box-plots.

### 6) 120-BAR SPATIAL TRUSS

HV measure results are depicted in Table 2, and it displays that MOPGO obtain the best $f_{mean}$ value of 0.553887 while the least $f_{std}$ MOSMA realizes a value of 0.000939 in comparison to other algorithms. The *FNRT* values for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO techniques are 450, 250, 425, 275, and 100, respectively. Thus, MOPGO ranked first, followed by MOSMA among all algorithms at 95% significance level, and hence have better solutions density near the Pareto front. In terms of GD metric $f_{mean}$ value, MOPVO obtained the least value of 91.44735, followed by MOPGO as per Table 3. Moreover, MOPVO attain the best *FNRT* values of 150 with minimum $f_{std}$ result that describes its improved convergence behavior. Table 4 shows the better CVG values for MOALO relatively while the least value of $f_{std}$ is procured by MOSMA, and thus MOALO demonstrates its improved coverage attribute relative to other considered algorithms. In CPF measure, as illustrated in Table 5, the MOPVS obtain the best $f_{mean}$ result of 0.737488 and MOPGO obtains the best $f_{std}$ result of 0.006523, relatively.

The *FNRT* values for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms are 300, 475, 425, 125, and 175, respectively. Thus, at a 95% significance level, MOPGO shows its improved coverage over Pareto front

**TABLE 6.** Results of (*DM-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{\min}$ | 0.804182 | 0.7566 | 0.5127 | 0.6393 | 0.6002 | 0.781268 | 0.7574 | 0.4561 | 0.6481 | 0.3643 |
| $f_{\max}$ | 0.848621 | 0.7971 | 0.5444 | 0.7148 | 0.6965 | 0.832545 | 0.7857 | 0.5052 | 0.7116 | 0.4852 |
| $f_{\text{mean}}$ | 0.831639 | 0.775275 | 0.522275 | 0.678225 | 0.641525 | 0.810576 | 0.7745 | 0.4852 | 0.676325 | 0.4256 |
| $f_{\text{median}}$ | 0.836877 | 0.7737 | 0.516 | 0.6794 | 0.6347 | 0.814245 | 0.77745 | 0.48975 | 0.6728 | 0.42645 |
| $f_{\text{std}}$ | 0.020127 | 0.01673 | 0.014889 | 0.032156 | 0.040236 | 0.021394 | 0.012513 | 0.02079 | 0.027789 | 0.054116 |
| *FNRT* | 500 | 400 | 100 | 275 | 225 | 500 | 400 | 200 | 300 | 100 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{\min}$ | 0.834328 | 0.7085 | 0.581 | 0.633 | 0.6572 | 0.831 | 0.7501 | 0.5951 | 0.6248 | 0.652 |
| $f_{\max}$ | 0.867969 | 0.7534 | 0.6003 | 0.7216 | 0.7054 | 0.909231 | 0.7792 | 0.6069 | 0.6506 | 0.6881 |
| $f_{\text{mean}}$ | 0.857506 | 0.73625 | 0.592075 | 0.68695 | 0.682825 | 0.867421 | 0.769625 | 0.6013 | 0.641275 | 0.66795 |
| $f_{\text{median}}$ | 0.863864 | 0.74155 | 0.5935 | 0.6966 | 0.68435 | 0.864726 | 0.7746 | 0.6016 | 0.64485 | 0.66585 |
| $f_{\text{std}}$ | 0.015579 | 0.01962 | 0.008514 | 0.038154 | 0.025815 | 0.032174 | 0.013297 | 0.006019 | 0.011329 | 0.014965 |
| *FNRT* | 500 | 400 | 100 | 275 | 225 | 500 | 400 | 100 | 200 | 300 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{\min}$ | 0.827458 | 0.7425 | 0.5154 | 0.6649 | 0.4895 | 0.695833 | 0.7017 | 0.3634 | 0.502 | 0.2634 |
| $f_{\max}$ | 0.848305 | 0.7667 | 0.5772 | 0.7258 | 0.5168 | 0.801316 | 0.7558 | 0.5402 | 0.5681 | 0.3827 |
| $f_{\text{mean}}$ | 0.836355 | 0.755825 | 0.55085 | 0.6936 | 0.5073 | 0.749226 | 0.724775 | 0.44805 | 0.5476 | 0.293225 |
| $f_{\text{median}}$ | 0.834828 | 0.75705 | 0.5554 | 0.69185 | 0.51145 | 0.749877 | 0.7208 | 0.4443 | 0.56015 | 0.2634 |
| $f_{\text{std}}$ | 0.009164 | 0.012615 | 0.027033 | 0.030715 | 0.012712 | 0.055675 | 0.023067 | 0.072312 | 0.030725 | 0.05965 |
| *FNRT* | 500 | 400 | 200 | 300 | 100 | 450 | 450 | 225 | 275 | 100 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{\min}$ | 0.589333 | 0.6574 | 0.4434 | 0.6031 | 0.3729 | 0.481919 | 0.481919 | 0.32755 | 0.299018 | 0.170778 |
| $f_{\max}$ | 0.760303 | 0.7079 | 0.5427 | 0.6965 | 0.4909 | 0.71474 | 0.622283 | 0.380748 | 0.508988 | 0.276233 |
| $f_{\text{mean}}$ | 0.680282 | 0.680275 | 0.49605 | 0.64765 | 0.425974 | 0.62308 | 0.563493 | 0.343698 | 0.396347 | 0.207593 |
| $f_{\text{median}}$ | 0.685746 | 0.6779 | 0.49905 | 0.6455 | 0.420048 | 0.64783 | 0.574885 | 0.333246 | 0.388691 | 0.191681 |
| $f_{\text{std}}$ | 0.086651 | 0.020962 | 0.040721 | 0.040101 | 0.048818 | 0.099852 | 0.06394 | 0.025147 | 0.086221 | 0.047581 |
| *FNRT* | 400 | 425 | 175 | 375 | 125 | 475 | 400 | 225 | 300 | 100 |
| *Average FNRT* | 478.125 | 409.375 | 165.625 | 287.5 | 159.375 | 478.125 | 409.375 | 165.625 | 287.5 | 159.375 |

quality. MOPGO $f_{\text{mean}}$ value for DM metric as listed in Table 6 has a percentage increase of 29.86%, 35.26%, 44.25%, and 12.70% from MOALO, MOSOS, MOSMA, and MOPVS, respectively. The *FNRT* values for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO are 500, 400, 100, 200, and 100, respectively.

Thus, at a 95% significance level, the proposed MOPGO algorithm outperforms others and exhibits better diversity in solutions. Similarly, from Table 7 for PD measure, the MOPGO has a percentage increase of 79.61%, 30.58%, 29.67%, and 47.60% in $f_{\text{mean}}$ value from MOALO, MOSOS, MOSMA, and MOPVS, respectively. Also, MOPGO obtains the maximum *FNRT* value of 450. These outcomes illustrate the improved pure diversity nature of investigated MOPGO over other methodologies. Table 8 reveals the SP metric results where MOPGO $f_{\text{mean}}$ value reported a substantial percentage decrease of 65.03%, 46.30%, 68.93%, and 87.17% from MOALO, MOSOS, MOSMA, and MOPVS. Similarly, MOPGO realize a major percentage decrease of 92.33%, 57.34%, 91.43%, and 72.12% corresponding to MOALO, MOSOS, MOSMA, and MOPVS in terms of $f_{\text{std}}$ results. Moreover, MOPGO manifests the best *FNRT* result that describes its optimal spacing feature. For SD indicator as per Table 9, $f_{\text{mean}}$ value reported by the MOPGO is a large percentage decrease of 79.52%, 79.36%, 70.12%, and 79.15% against MOALO, MOSOS, MOSMA, and MOPVS. MOPGO realize a best *FNRT* value of 100 at minimum $f_{\text{std}}$ value of 0.023643 that manifests its well-distributed non-dominated solutions relative to others. For IGD performance measure, the MOSMA realize the best $f_{\text{mean}}$ and $f_{\text{std}}$ value along with the least *FNRT* value, followed by MOPVS and MOPGO, as presented in Table 10. Regarding RT metric, the proposed MOPGO realize a minimum $f_{\text{mean}}$ value of 131.6244 relatively. MOPGO, MOPVS, MOSMA, MOSOS, and MOALO realize the *FNRT* value of 150, 175, 325, 350, 350, and 500. At a 95% significance level, MOPGO ranked first and manifested its least computational time to reach the optimal solution. Table 12 presents the BCS results

**TABLE 7.** Results of (*PD-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{\min}$ | 63896267 | 51838045 | 52114664 | 53783885 | 33395812 | 52094520 | 44620434 | 43023633 | 51354240 | 26916981 |
| $f_{\max}$ | 72464000 | 65318476 | 77688008 | 76290111 | 49650278 | 69519307 | 57137540 | 57143645 | 62163809 | 34973925 |
| $f_{\text{mean}}$ | 69119969 | 58182377 | 65896975 | 62302844 | 38144989 | 63722203 | 51200123 | 49146439 | 58488576 | 32322964 |
| $f_{\text{median}}$ | 70059805 | 57786494 | 66892614 | 59568689 | 34766934 | 66637492 | 51521259 | 48209238 | 60218127 | 33700474 |
| $f_{\text{std}}$ | 3796213 | 6599317 | 10795370 | 10055437 | 7715646 | 7942384 | 6306248 | 5890566 | 4845287 | 3721075 |
| *FNRT* | 425 | 275 | 375 | 325 | 100 | 450 | 275 | 250 | 425 | 100 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{\min}$ | 33598182 | 24801785 | 26353988 | 26127797 | 24626669 | 4.05E+08 | 3.31E+08 | 4.1E+08 | 3.41E+08 | 2.14E+08 |
| $f_{\max}$ | 43972086 | 30047562 | 30727768 | 36219296 | 31815915 | 6.62E+08 | 4.42E+08 | 4.65E+08 | 5.3E+08 | 4.13E+08 |
| $f_{\text{mean}}$ | 39459659 | 27799928 | 28479814 | 33561794 | 28413328 | 5.55E+08 | 3.76E+08 | 4.28E+08 | 4.25E+08 | 3.09E+08 |
| $f_{\text{median}}$ | 40134184 | 28175182 | 28418751 | 35950041 | 28605365 | 5.76E+08 | 3.66E+08 | 4.18E+08 | 4.14E+08 | 3.05E+08 |
| $f_{\text{std}}$ | 4625217 | 2635905 | 1804084 | 4957675 | 2945634 | 1.08E+08 | 51971669 | 25173050 | 78321456 | 86596859 |
| *FNRT* | 500 | 225 | 225 | 325 | 225 | 450 | 200 | 350 | 325 | 175 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{\min}$ | 13003561 | 13079813 | 10768528 | 13985101 | 6835959 | 1.08E+08 | 1.28E+08 | 49333508 | 94404323 | 45336403 |
| $f_{\max}$ | 24395936 | 17351166 | 12648096 | 17328534 | 9036429 | 2.57E+08 | 1.61E+08 | 83349790 | 1.52E+08 | 77149363 |
| $f_{\text{mean}}$ | 17315293 | 15637868 | 11411361 | 15898856 | 8099462 | 1.56E+08 | 1.44E+08 | 68720635 | 1.24E+08 | 56050768 |
| $f_{\text{median}}$ | 15930836 | 16060246 | 11114410 | 16140895 | 8262730 | 1.29E+08 | 1.44E+08 | 71099621 | 1.25E+08 | 50858653 |
| $f_{\text{std}}$ | 5339034 | 1810701 | 845748.2 | 1655964 | 1050231 | 68458456 | 17102375 | 15565291 | 23713875 | 14771458 |
| *FNRT* | 400 | 400 | 200 | 400 | 100 | 400 | 425 | 175 | 375 | 125 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{\min}$ | 14820683 | 8360067 | 14916235 | 17458746 | 13365426 | 8.01E+09 | 5.81E+09 | 3.24E+09 | 5.66E+09 | 1.83E+09 |
| $f_{\max}$ | 22339929 | 20077248 | 24163371 | 20913061 | 16228165 | 1.15E+10 | 1.26E+10 | 5.6E+09 | 1.27E+10 | 3.67E+09 |
| $f_{\text{mean}}$ | 18636119 | 13111841 | 20323843 | 19827690 | 14625593 | 9.51E+09 | 9.7E+09 | 3.97E+09 | 8.43E+09 | 2.53E+09 |
| $f_{\text{median}}$ | 18691931 | 12005024 | 21107883 | 20469475 | 14454390 | 9.27E+09 | 1.02E+10 | 3.53E+09 | 7.69E+09 | 2.3E+09 |
| $f_{\text{std}}$ | 3939996 | 5067413 | 3966399 | 1619731 | 1357086 | 1.49E+09 | 2.83E+09 | 1.09E+09 | 3.1E+09 | 8.18E+08 |
| *FNRT* | 375 | 125 | 375 | 400 | 225 | 400 | 425 | 200 | 375 | 100 |
| *Average FNRT* | 425 | 293.75 | 268.75 | 368.75 | 143.75 | 425 | 293.75 | 268.75 | 368.75 | 143.75 |

for 120-bar 3D truss that proves that MOPGO, i.e., least obtain the best result $f_{weight}$ value of 20453.26 with maximum $f_{compliance}$ of 1447942 against all other considered algorithms.

The best Pareto fronts achieved by individual algorithms are indicated in Figure 19 simultaneously with their BCS results. The combined Pareto fronts are contrasted for qualitative analysis in the last plot that reflects the continuous and well-distributed nature of MOPGO solutions. Furthermore, all ten performance metrics results are plotted in boxplot as illustrated in Figure 20, demonstrating the dominance of the suggested MOPGO algorithm quantitatively over others.

### 7) 200-BAR SPATIAL TRUSS

The HV metric results are illustrated in Table 2, from which it is evident that MOPVS and MOPGO achieved the superior $f_{\text{mean}}$ and $f_{\text{std}}$ value relative to other considered algorithms. Moreover, the *FNRT* metric assigned the rank of 400, 425, 200, 375, and 100 to the MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms, respectively. Hence at*FNRT* 95% significance level, MOPGO expressed its high solutions density in the proximity of the Pareto front. For GD indicator, MOSMA exhibit its dominance by obtaining the best $f_{\text{mean}}$ and $f_{\text{std}}$ value along with the best *FNRT* value of 125 as depicted in Table 3. According to Table 4, the best CVG metric $f_{\text{mean}}$ the result was obtained by MOPGO shows a percentage decrease of 34% from all other optimization techniques. The *FNRT* results obtained by MOPGO, MOPVS, MOSMA, MOSOS, and MOALO are 100, 350, 350, 350, and 350 each. At a 95% significance level, MOPGO ranked first and illustrated its enhanced coverage characteristic among all. The $f_{\text{mean}}$ results of CPF metric for the MOPGO algorithm exhibit a substantial percentage increase of 385.20%, 101.32%, 118.29%, and 18.12% from MOALO, MOSOS, MOSMA, and MOPVS, respectively, as illustrated in Table 5. Moreover, the *FNRT* results attain by MOPGO, MOPVS, MOSMA, MOSOS, and MOALO techniques are 500, 400, 225, 275, and 100 each. These prospects manifest MOPGO enhanced coverage over Pareto front characteristics

**TABLE 8.** Results of (*SP-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{min}$ | 297.6046 | 2184.726 | 888.3706 | 516.8352 | 531.2223 | 214.5549 | 1173.424 | 428.2149 | 303.7444 | 191.6801 |
| $f_{max}$ | 685.4001 | 2801.511 | 1346.646 | 730.3162 | 1814.697 | 1751.82 | 2079.446 | 1597.901 | 479.3324 | 647.1748 |
| $f_{mean}$ | 427.3006 | 2476.311 | 1049.798 | 629.3917 | 941.9936 | 660.9634 | 1574.278 | 1162.472 | 403.7876 | 434.2584 |
| $f_{median}$ | 363.0989 | 2459.504 | 982.0871 | 635.2077 | 711.0277 | 338.7393 | 1522.122 | 1311.886 | 416.0367 | 449.0893 |
| $f_{std}$ | 176.6763 | 331.0173 | 206.5312 | 96.65837 | 588.6415 | 730.3312 | 383.1655 | 553.8538 | 73.65402 | 216.6911 |
| *FNRT* | 125 | 500 | 375 | 200 | 300 | 225 | 475 | 325 | 250 | 225 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{min}$ | 104.2963 | 619.3793 | 273.8689 | 226.2322 | 278.551 | 2228.979 | 17510.02 | 5022.761 | 3407.838 | 3717.766 |
| $f_{max}$ | 128.3055 | 1167.577 | 300.7606 | 317.179 | 1235.8 | 3212.08 | 21146.66 | 16053.08 | 5983.912 | 15990.16 |
| $f_{mean}$ | 117.5281 | 822.0693 | 291.4915 | 269.8598 | 542.2806 | 2545.739 | 19853.94 | 8195.298 | 4741.079 | 7280.604 |
| $f_{median}$ | 118.7552 | 750.6605 | 295.6683 | 268.0141 | 327.3856 | 2370.948 | 20379.53 | 5852.677 | 4786.283 | 4707.248 |
| $f_{std}$ | 12.33472 | 238.6874 | 12.04796 | 41.16262 | 463.0835 | 450.0255 | 1614.615 | 5253.406 | 1055.037 | 5873.13 |
| *FNRT* | 100 | 475 | 300 | 250 | 375 | 100 | 500 | 350 | 250 | 300 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{min}$ | 53.62469 | 275.143 | 111.6223 | 107.9337 | 46.72249 | 233.1611 | 1510.139 | 686.2533 | 1128.689 | 224.7026 |
| $f_{max}$ | 128.8711 | 446.2343 | 165.0344 | 244.5844 | 96.63768 | 411.9699 | 1966.236 | 1042.236 | 1971.306 | 598.0939 |
| $f_{mean}$ | 79.51933 | 362.4493 | 149.5973 | 150.3453 | 73.37848 | 289.6515 | 1735.383 | 921.8234 | 1379.492 | 370.9954 |
| $f_{median}$ | 67.79075 | 364.2099 | 160.8663 | 124.4316 | 75.07688 | 256.7375 | 1732.579 | 979.4022 | 1208.986 | 330.5926 |
| $f_{std}$ | 33.76103 | 73.5329 | 25.39327 | 63.44049 | 20.86319 | 82.34433 | 196.0732 | 163.945 | 397.8876 | 174.3569 |
| *FNRT* | 150 | 500 | 350 | 325 | 175 | 150 | 475 | 300 | 425 | 150 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{min}$ | 86.12169 | 370.9343 | 137.425 | 141.9594 | 141.9138 | 24747.67 | 59830.61 | 79785.65 | 48140 | 35830 |
| $f_{max}$ | 325.133 | 1331.521 | 446.5709 | 270.3608 | 337.7114 | 38474.18 | 146823.9 | 164705 | 120490.3 | 45553.44 |
| $f_{mean}$ | 193.363 | 727.5934 | 340.2282 | 208.9667 | 214.5523 | 29223.46 | 98820.21 | 104926.7 | 89495.66 | 39101.09 |
| $f_{median}$ | 181.0986 | 603.9593 | 388.4584 | 211.7733 | 189.292 | 26835.99 | 94313.15 | 87608.08 | 94676.15 | 37510.45 |
| $f_{std}$ | 100.1716 | 417.3559 | 144.3591 | 70.37141 | 85.91492 | 6377.023 | 36832.92 | 40063.3 | 36117.96 | 4519.288 |
| *FNRT* | 225 | 475 | 350 | 225 | 225 | 100 | 425 | 375 | 400 | 200 |
| *Average FNRT* | 146.875 | 478.125 | 340.625 | 290.625 | 243.75 | 146.875 | 478.125 | 340.625 | 290.625 | 243.75 |

in relation to other investigated algorithms. Table 6 depicts the DM metric results according to which MOPGO $f_{mean}$ value has a percentage increase of 155.51%, 36.81%, 67.21%, and 3.37% from MOALO, MOSOS, MOSMA, and MOPVS, respectively, at minimum $f_{std}$ value. MOPGO, as per the *FNRT* obtained a maximum 450 value relatively and at 95% significance level ranked first. These outcomes exhibit the improved solution diversity of MOPGO concerning other contrasted methodology.

Similarly, for PD performance measure as illustrated in Table 7, the MOPGO algorithm reported a substantial percentage increase of 178.31%, 25.80%, and 127% in $f_{mean}$ value in respect to MOALO, MOSOS, and MOSMA. Here, the MOPGO algorithm demonstrates its superior pure diversity behavior through its best $f_{mean}$ and $f_{std}$ that was eventually proved by its high *FNRT* value of 400 in comparison to other selected optimization techniques. For SP measure, according to Table 8, MOPGO evidence an acceptable value of $f_{mean}$ i.e., 289.6515 that is significantly less than MOPVS, MOSOS, and MOSMA methodologies. Moreover, $f_{std}$ value reported by MOPGO is a percentage decrease of 79.30%, 58%, 52.77%, and 49.77% from MOSOS, MOPVS, MOALO, and MOSMA. Furthermore, the *FNRT* metric indicates the best rank of 150 by MOPGO and, thus, at 95% significance level, displays its enhanced spacing quality against other algorithms. From SD metric findings as shown in Table 9, MOPVS and MOSMA finds the superior $f_{mean}$ and $f_{std}$ value over others. The *FNRT* results attain by MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms are 300, 150, 150, 425, and 475 each. At a 95% significance level, MOPGO displays its well-distributed non-dominated solutions. In the IGD test, as indicated in Table 10, the MOPGO algorithm exhibits a substantial percentage decrease of 84.10%, 19.08%, 80.19%, and 17.86% from MOALO, MOSOS, MOSMA, and MOPVS, respectively. The *FNRT* results attain by MOPGO, MOPVS, MOSMA, MOSOS, and MOALO techniques are 175, 225, 425, 200, and 475 each. These results are showing its

**TABLE 9.** Results of (*SD-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{\min}$ | 0.239788 | 0.924493 | 0.629585 | 0.918107 | 0.923815 | 0.464974 | 0.828689 | 0.572478 | 0.842599 | 0.88462 |
| $f_{\max}$ | 0.344418 | 1.058567 | 0.830546 | 1.025942 | 0.959251 | 0.66236 | 0.889789 | 0.985028 | 0.884771 | 0.961603 |
| $f_{\text{mean}}$ | 0.283055 | 0.988318 | 0.744116 | 0.982385 | 0.94139 | 0.571281 | 0.855764 | 0.766081 | 0.861879 | 0.932239 |
| $f_{\text{median}}$ | 0.274007 | 0.985106 | 0.758167 | 0.992746 | 0.941247 | 0.578895 | 0.852288 | 0.753409 | 0.860073 | 0.941367 |
| $f_{\text{std}}$ | 0.046962 | 0.055716 | 0.083838 | 0.048874 | 0.017233 | 0.086756 | 0.025545 | 0.171282 | 0.017739 | 0.033402 |
| *FNRT* | 100 | 425 | 200 | 425 | 350 | 100 | 325 | 275 | 325 | 475 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{\min}$ | 0.177001 | 0.842797 | 0.542787 | 0.683891 | 0.752214 | 0.178727 | 0.896872 | 0.595314 | 0.901197 | 0.894332 |
| $f_{\max}$ | 0.217493 | 0.948056 | 0.573612 | 0.88437 | 0.952348 | 0.229966 | 0.990101 | 0.69815 | 1.020614 | 1.082661 |
| $f_{\text{mean}}$ | 0.199255 | 0.880547 | 0.55856 | 0.826468 | 0.858166 | 0.197805 | 0.949112 | 0.662095 | 0.958623 | 0.966314 |
| $f_{\text{median}}$ | 0.201262 | 0.865668 | 0.558921 | 0.868805 | 0.864051 | 0.191263 | 0.954737 | 0.677458 | 0.95634 | 0.944131 |
| $f_{\text{std}}$ | 0.018069 | 0.049028 | 0.01397 | 0.096023 | 0.085545 | 0.023643 | 0.041454 | 0.046297 | 0.049373 | 0.082048 |
| *FNRT* | 100 | 400 | 200 | 400 | 400 | 100 | 375 | 200 | 425 | 400 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{\min}$ | 0.475421 | 0.736658 | 0.584838 | 0.844236 | 0.865627 | 0.773623 | 0.599074 | 0.594473 | 0.904169 | 0.909956 |
| $f_{\max}$ | 0.578204 | 0.938775 | 0.801246 | 0.90908 | 0.921314 | 0.845811 | 0.723365 | 0.759342 | 0.962052 | 0.95599 |
| $f_{\text{mean}}$ | 0.523486 | 0.834174 | 0.71639 | 0.878386 | 0.901188 | 0.82185 | 0.659388 | 0.649975 | 0.92488 | 0.939279 |
| $f_{\text{median}}$ | 0.520159 | 0.830632 | 0.739739 | 0.880114 | 0.908906 | 0.833984 | 0.657556 | 0.623042 | 0.91665 | 0.945585 |
| $f_{\text{std}}$ | 0.049119 | 0.083162 | 0.092691 | 0.027418 | 0.026223 | 0.033901 | 0.054871 | 0.075505 | 0.025472 | 0.020249 |
| *FNRT* | 100 | 325 | 225 | 400 | 450 | 300 | 150 | 150 | 425 | 475 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{\min}$ | 0.537625 | 0.903756 | 0.827936 | 0.936454 | 0.979852 | 0.83945 | 0.617021 | 0.718818 | 0.882062 | 0.959359 |
| $f_{\max}$ | 0.709925 | 1.026421 | 0.906392 | 0.972011 | 1.005357 | 0.873767 | 0.800822 | 0.899675 | 1.0147 | 0.979317 |
| $f_{\text{mean}}$ | 0.603767 | 0.961691 | 0.873029 | 0.953616 | 0.987145 | 0.853372 | 0.683959 | 0.837065 | 0.975461 | 0.968067 |
| $f_{\text{median}}$ | 0.583759 | 0.958294 | 0.878895 | 0.953 | 0.981685 | 0.850135 | 0.658996 | 0.864883 | 1.00254 | 0.966797 |
| $f_{\text{std}}$ | 0.083026 | 0.052262 | 0.035189 | 0.01736 | 0.012193 | 0.015802 | 0.080384 | 0.084296 | 0.062559 | 0.00839 |
| *FNRT* | 100 | 375 | 200 | 350 | 475 | 250 | 100 | 275 | 450 | 425 |
| *Average FNRT* | 143.75 | 309.375 | 215.625 | 400 | 431.25 | 143.75 | 309.375 | 215.625 | 400 | 431.25 |

competing convergence-spread parity attribute comparatively. In terms of RT measure, MOPGO realize the best $f_{\text{mean}}$ and $f_{\text{std}}$ value followed by MOPVS as shown in Table 11. Moreover, the least *FNRT* value of 175 portrays the least computational time executed by MOPGO relatively to reach the optimal solution. Table 12 unveils the BCS that satisfied each objective, i.e., minimum $f_{weight}$ and maximum $f_{compliance}$ and it is evident that the suggested MOPGO accomplishes a superior value of (6687.743, 65199.19) for a 200-bar 3D truss among all executed algorithms.

Figure 22 depicts the best Pareto fronts for individual algorithms and their corresponding BCS results. The combined Pareto fronts illustration describes the well-diverse, continuous, and smooth qualitative behavior of MOPGO relatively. The dominance of the proposed MOPGO technique quantitatively over others is also illustrated in Figure 23 comprehensively in the form of all investigated performance metric outcomes box-plots.

#### 8) 942-BAR SPATIAL TOWER TRUSS

According to Table 2 that displays HV measure results, the best $f_{\text{mean}}$ and $f_{\text{std}}$ value is obtained by MOPGO followed by MOPVS. The *FNRT* values for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms are 425, 375, 125, 325, and 250, respectively. Thus, MOPGO ranked first and at 95% significance level exhibit a superior solution density near Pareto front. In terms of GD metric $f_{\text{mean}}$ value, MOSMA, and MOSOS obtained the best value among all as per Table 3. MOPGO attains the *FNRT* value of 425 that describes its improved convergence behavior at a 95% significance level relatively. Table 4 shows the best CVG $f_{\text{mean}}$ value, according to which MOPGO manifest a 61%, 58.06%, 61%, and 61% from MOALO, MOSOS, MOSMA, and MOPVS, respectively. The MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms realize a *FNRT* values of 100, 375, 375, 275, and 375 each. MOPGO ranked at the first position at 95% significance level and governed its improved

**TABLE 10.** Results of (*IGD-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{min}$ | 1663.004 | 669.3506 | 1010.344 | 1496.253 | 16935.28 | 1447.864 | 735.9324 | 2151.355 | 1183.869 | 6799.245 |
| $f_{max}$ | 2491.803 | 1448.771 | 1391.227 | 5831.317 | 18199.38 | 3692.308 | 1681.942 | 3236.961 | 3365.199 | 10546.19 |
| $f_{mean}$ | 2195.228 | 1105.985 | 1177.911 | 3281.61 | 17571.57 | 2619.698 | 1198.052 | 2716.422 | 2487.961 | 9412.384 |
| $f_{median}$ | 2313.052 | 1152.909 | 1155.036 | 2899.435 | 17575.82 | 2669.31 | 1187.167 | 2738.687 | 2701.389 | 10152.05 |
| $f_{std}$ | 372.613 | 324.3478 | 164.507 | 1824.671 | 549.2313 | 1118.101 | 454.5308 | 510.1474 | 1030.674 | 1754.399 |
| *FNRT* | 325 | 150 | 150 | 375 | 500 | 275 | 125 | 300 | 300 | 500 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{min}$ | 376.755 | 174.2427 | 186.7903 | 271.4774 | 2350.4 | 9616.403 | 5351.738 | 4139.374 | 10269.31 | 70203.28 |
| $f_{max}$ | 921.6493 | 188.3175 | 297.684 | 659.4901 | 3611.171 | 20565.82 | 15325.65 | 5581.666 | 35020.58 | 168283.4 |
| $f_{mean}$ | 560.3473 | 180.3499 | 226.2102 | 411.3597 | 2757.912 | 13619.08 | 8097.637 | 4603.163 | 18295.17 | 110218.4 |
| $f_{median}$ | 471.4924 | 179.4197 | 210.1832 | 357.2357 | 2535.038 | 12147.04 | 5856.579 | 4345.806 | 13945.39 | 101193.5 |
| $f_{std}$ | 245.1732 | 6.642211 | 49.68994 | 183.8964 | 576.7051 | 4783.979 | 4834.275 | 671.9493 | 11285.14 | 47221.25 |
| *FNRT* | 400 | 100 | 225 | 275 | 500 | 325 | 200 | 125 | 350 | 500 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{min}$ | 213.3998 | 149.1124 | 347.9143 | 192.0994 | 2305.397 | 3075.448 | 3804.123 | 21304.65 | 3626.427 | 29784.44 |
| $f_{max}$ | 289.3297 | 724.9337 | 670.4392 | 481.891 | 3437.252 | 8863.133 | 11799.4 | 33331.05 | 10875.23 | 43353.69 |
| $f_{mean}$ | 239.7479 | 394.5028 | 481.7558 | 292.0681 | 2748.639 | 5707.932 | 6949.556 | 28826.64 | 7054.015 | 35899.25 |
| $f_{median}$ | 228.1311 | 351.9826 | 454.3349 | 247.141 | 2625.954 | 5446.573 | 6097.352 | 30335.44 | 6857.203 | 35229.43 |
| $f_{std}$ | 33.90431 | 247.2318 | 135.8339 | 129.5593 | 502.4452 | 2502.579 | 3799.514 | 5225.252 | 3031.048 | 7098.44 |
| *FNRT* | 175 | 250 | 350 | 225 | 500 | 175 | 225 | 425 | 200 | 475 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{min}$ | 3350.781 | 1082.766 | 1199.562 | 2001.666 | 3546.307 | 1208912 | 1720320 | 4030769 | 1635280 | 4894355 |
| $f_{max}$ | 9267.69 | 3177.081 | 2565.982 | 2643.415 | 5132.376 | 2941170 | 3069593 | 4492126 | 3757705 | 6174558 |
| $f_{mean}$ | 5530.265 | 2467.42 | 1757.774 | 2390.604 | 4543.579 | 1895785 | 2122510 | 4357333 | 2690449 | 5698860 |
| $f_{median}$ | 4751.295 | 2804.917 | 1632.777 | 2458.668 | 4747.817 | 1716529 | 1850064 | 4453218 | 2684405 | 5863262 |
| $f_{std}$ | 2673.076 | 952.8802 | 589.8991 | 286.5413 | 722.0399 | 737012.1 | 634401.2 | 218480.7 | 878142 | 567831 |
| *FNRT* | 450 | 250 | 125 | 225 | 450 | 175 | 200 | 400 | 225 | 500 |
| *Average FNRT* | 287.5 | 187.5 | 262.5 | 271.875 | 490.625 | 287.5 | 187.5 | 262.5 | 271.875 | 490.625 |

coverage quality. In CPF metric, as illustrated in Table 5, MOPGO reported a 589.11%, 284.61%, 263.69%, and 50.37% increase in $f_{mean}$ value relative to MOALO, MOSOS, MOSMA, and MOPVS, respectively. The *FNRT* values for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO algorithms are 475, 400, 250, 225, and 150, respectively. At a 95% significance level, the investigated MOPGO algorithm outperforms others in terms of coverage over Pareto front quality. MOPGO $f_{mean}$ value for DM metric as depicted in Table 6 has a percentage increase of 200.14%, 57.20%, 81.28%, and 10.57% from MOALO, MOSOS, MOSMA, and MOPVS, respectively. The *FNRT* results for MOPGO, MOPVS, MOSMA, MOSOS, and MOALO are 475, 400, 225, 300, and100, respectively. Thus, at a 95% significance level, the proposed MOPGO techniques exhibit better diversity in solutions.

Table 7 describes the PD measure results according to which the MOPGO finds the best $f_{mean}$ value that shows a substantial percentage increase of 275.88% and 139.54% from MOALO and MOSMA, respectively. MOSMA and MOPGO also obtain the minimum value of $f_{std}$ whereas the best *FNRT* results are realized by MOPVS and MOPGO. Hence at a 95% significance level, MOPGO illustrates the improved pure diversity nature over other methodologies. Table 8 reveals the SP metric results where MOPGO $f_{mean}$ value reported a percentage decrease of 25.26%, 67.34%, 72.14%, and 70.42% from MOALO, MOSOS, MOSMA, and MOPVS. Moreover, MOPGO, MOPVS, MOSMA, MOSOS, and MOALO obtain the *FNRT* value of 100, 425, 375, 400, and 200, respectively, describing the optimal spacing feature MOPGO over other considered methodologies. For SD indicator as per Table 9, MOPVS realize the best $f_{mean}$ value while the best $f_{std}$ result is procured by MOPGO. The MOPVO realizes a best *FNRT* value of 100, followed by MOPGO. At a 95% significance level, the suggested MOPGO algorithm manifests its well-distributed non-dominated solutions relative to others. From IGD performance measure, as depicted in Table 10, MOPGO realize the best $f_{mean}$ value with

**TABLE 11.** Results of (*RUNTIME – RT-Metric*) on truss bar problems.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|---|---|---|---|---|
| | **10 *bar*** | | | | | **72 *bar*** | | | | |
| $f_{\min}$ | 10.52134 | 10.60112 | 13.71625 | 14.032 | 24.48882 | 72.06388 | 72.40407 | 75.03435 | 78.25583 | 89.37959 |
| $f_{\max}$ | 11.72248 | 12.60622 | 14.37789 | 14.95669 | 25.16257 | 85.69324 | 78.58249 | 84.33703 | 88.71498 | 101.677 |
| $f_{\text{mean}}$ | 11.31989 | 11.20518 | 14.07731 | 14.41821 | 24.78684 | 75.50898 | 74.13212 | 77.44584 | 80.96273 | 92.83377 |
| $f_{\text{median}}$ | 11.51787 | 10.8067 | 14.10755 | 14.34209 | 24.74799 | 72.1394 | 72.77097 | 75.206 | 78.44006 | 90.13923 |
| $f_{\text{std}}$ | 0.549554 | 0.943371 | 0.274431 | 0.418797 | 0.309626 | 6.789729 | 2.973097 | 4.595131 | 5.169407 | 5.908018 |
| *FNRT* | 175 | 125 | 325 | 375 | 500 | 150 | 175 | 275 | 400 | 500 |
| | **25 *bar*** | | | | | **120 *bar*** | | | | |
| $f_{\min}$ | 27.15025 | 27.37074 | 31.30694 | 31.55037 | 43.60418 | 127.1529 | 127.4661 | 130.6324 | 135.676 | 147.1061 |
| $f_{\max}$ | 27.5003 | 27.89938 | 40.44693 | 34.84346 | 44.93789 | 143.1505 | 140.3373 | 148.2668 | 141.2581 | 162.4699 |
| $f_{\text{mean}}$ | 27.26802 | 27.5558 | 33.83647 | 32.42191 | 43.99345 | 131.6244 | 131.409 | 135.1752 | 137.1738 | 151.4025 |
| $f_{\text{median}}$ | 27.21076 | 27.47654 | 31.796 | 31.64691 | 43.71587 | 128.0971 | 128.9163 | 130.9007 | 135.8806 | 148.0169 |
| $f_{\text{std}}$ | 0.157958 | 0.243625 | 4.421022 | 1.615126 | 0.636058 | 7.697764 | 5.996037 | 8.730048 | 2.727692 | 7.391627 |
| *FNRT* | 100 | 200 | 350 | 350 | 500 | 150 | 175 | 325 | 350 | 500 |
| | **37 *bar*** | | | | | **200 *bar*** | | | | |
| $f_{\min}$ | 35.68907 | 36.00401 | 38.73837 | 40.01812 | 51.38625 | 196.3653 | 197.1134 | 198.8859 | 204.141 | 213.1838 |
| $f_{\max}$ | 36.32434 | 36.73899 | 45.76548 | 40.2507 | 59.62096 | 205.2674 | 219.7489 | 199.9099 | 204.7652 | 220.1894 |
| $f_{\text{mean}}$ | 35.9093 | 36.25632 | 40.51906 | 40.19059 | 53.78413 | 198.9954 | 203.1025 | 199.4259 | 204.484 | 215.2863 |
| $f_{\text{median}}$ | 35.81189 | 36.14113 | 38.78619 | 40.24677 | 52.06466 | 197.1744 | 197.7739 | 199.4539 | 204.5148 | 213.886 |
| $f_{\text{std}}$ | 0.298144 | 0.346248 | 3.49769 | 0.114997 | 3.935561 | 4.223112 | 11.10212 | 0.525912 | 0.258 | 3.289835 |
| *FNRT* | 100 | 200 | 325 | 375 | 500 | 175 | 225 | 250 | 350 | 500 |
| | **60 *bar*** | | | | | **942 *bar*** | | | | |
| $f_{\min}$ | 63.72904 | 64.26448 | 66.1612 | 69.28469 | 80.3214 | 1001.134 | 1005.016 | 1002.336 | 1011.926 | 1018.45 |
| $f_{\max}$ | 70.49556 | 70.51224 | 69.43394 | 74.16968 | 82.06947 | 1013.831 | 1029.895 | 1027.993 | 1016.438 | 1035.815 |
| $f_{\text{mean}}$ | 65.45753 | 65.87182 | 67.09768 | 70.57497 | 80.97416 | 1006.416 | 1011.646 | 1016.036 | 1014.367 | 1025.095 |
| $f_{\text{median}}$ | 63.80276 | 64.35527 | 66.39778 | 69.42276 | 80.75289 | 1005.349 | 1005.836 | 1016.908 | 1014.551 | 1023.056 |
| $f_{\text{std}}$ | 3.359196 | 3.094192 | 1.565267 | 2.39865 | 0.764819 | 5.39259 | 12.17995 | 11.93667 | 2.252695 | 7.573137 |
| *FNRT* | 125 | 225 | 250 | 400 | 500 | 125 | 275 | 325 | 325 | 450 |
| *Average FNRT* | 137.5 | 200 | 303.125 | 365.625 | 493.75 | 137.5 | 200 | 303.125 | 365.625 | 493.75 |

a percentage decrease of 66.73%, 29.53%, 56.49%, and 10.68% from MOALO, MOSOS, MOSMA, and MOPVS. The least *FNRT* value of 175 is manifest by MOPGO, which is followed by MOPVS.

Therefore, at a 95% significance level, MOPGO proves its improved uniformity-convergence-spread attribute relatively. Regarding RT metric, the proposed MOPGO realize a minimum $f_{\text{mean}}$ value of 1006.416 relatively at minimum $f_{\text{std}}$ result. MOPGO, MOPVS, MOSMA, MOSOS, and MOALO realize the *FNRT* value of 125, 275, 325, 325, and 450. At a 95% significance level, MOPGO ranked first and manifested its least computational time to reach the optimal solution. Table 12 presents the BCS results for the 942-bar tower truss that proves that the best result is obtained by MOPGO, i.e. (6867423, 2296479) against all other considered algorithms. The best Pareto fronts achieved by individual algorithms are indicated in Figure 25 simultaneously with their BCS results. The combined Pareto fronts are contrasted for qualitative analysis in the last plot, reflecting the continuous and well-distributed nature of MOPGO solutions. Furthermore, all ten performance metrics results by all considered are plotted in boxplot as illustrated in Figure 26 that demonstrates the dominance of the suggested MOPGO algorithm quantitatively over others.

To make an outlook of the proposed MOPGO efficiency, each performance metric *FNRT* average value is computed and analyzed. For SD, GD, CVG, CPF, DM, PD, SP, IGD, and RT measure, MOPGO realizes a best 400, 206.25, 226.5625, 393.75, 478.125, 425, 146.875, 143.75, 287.5, and 137.5 value with respect to other algorithms. Moreover, MOPGO finds the best-compromised solution for all the eight considered planar and spatial benchmarks. Thus, all these prospects demonstrate the superiority of the proposed MOPGO algorithm in solving multi-objective large and complex structural optimization problems and can create harmony between the local intensification and global diversification of search.

**TABLE 12.** The best compromise solution *(BCS)* results of all algorithms.

| Algorithms | MOPGO | MOPVS | MOSMA | MOSOS | MOALO |
|---|---|---|---|---|---|
| **10 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 3653.518 | 7764.568 | 6470.473 | 8113.873 | 9433.749 |
| $f_{compliance}$ | 138077.3 | 62904.17 | 75972.26 | 60179.1 | 53160.87 |
| **25 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 2346.643 | 4569.145 | 4299.293 | 4458.512 | 5020.84 |
| $f_{compliance}$ | 35377.12 | 17574.4 | 18694.87 | 17916.99 | 16161.94 |
| **37 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 1180.289 | 2401.176 | 2232.516 | 2429.324 | 2572.644 |
| $f_{compliance}$ | 18411.19 | 8957.591 | 9534.427 | 8776.521 | 8447.349 |
| **60 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 1955.053 | 5127.511 | 4032.198 | 5100.266 | 4994.388 |
| $f_{compliance}$ | 81300.43 | 27080.56 | 33917.8 | 26945.6 | 29013.76 |
| **72 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 6687.743 | 11126.34 | 8744.188 | 11845.38 | 11705.13 |
| $f_{compliance}$ | 65199.19 | 36520.59 | 46750.3 | 33917.8 | 35036.05 |
| **120 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 20453.26 | 46565.56 | 41994.43 | 47102.66 | 53635.9 |
| $f_{compliance}$ | 1447942 | 633309.9 | 700154.4 | 623544.1 | 550986.3 |
| **200 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 24956.83 | 45246.9 | 40631.44 | 44493.47 | 43150.95 |
| $f_{compliance}$ | 70681.12 | 35508.79 | 37742.5 | 37242.01 | 35558.17 |
| **942 – *bar truss problem*** | | | | | |
| $f_{weight}$ | 6867423 | 8362877 | 9486430 | 7554179 | 7225323 |
| $f_{compliance}$ | 2296479 | 1740965 | 1316513 | 1951677 | 1498586 |

## VI. CONCLUSION

The framework and development of a new MOPGO algorithm for multi-objective truss-bar design problems are discussed in this paper. The MOPGO algorithm combines the three primary phases of PGO, namely excitation, de-excitation, and ionization, with plasma generation to support the search for the global best solution. For performance measure, eight challenging multi-objective structure layout optimization problems (i.e., 10-bar, 25-bar, 37-bar, 60-bar, 72-bar, 120-bar, 200-bar, and 942-bar) are tested related to various constraints with distinct design variables to adjust for the practicality of examination. The results obtained by the proposed MOPGO are compared with four well-known algorithms under the same input parameters. For all eight design problems, ten performance metrics (HV, GD, CVG, CPF, DM, PD, SP, SD, IGD, and RT) are used to assess the enhancement and diversifying of a non-dominated solution1 set. The analysis and discussions show that the MOPGO algorithm has a significant advantage over MOPVS, MOSMA, MOSOS, and MOALO in terms of coverage, convergence, and solution diversity. Furthermore, the proposed algorithm is rated first for all design problems based on the average *FNRT* values. The analysis shows that MOPGO is capable of successfully solving large real-world optimization problems. Therefore, it is concluded that the MOPGO algorithm can solve problems involving a higher-dimensional optimization process.

In future, the researchers in various fields can utilize the proposed MOPGO algorithm to solve multi-modal and non-linear functional demanding technical challenges with several competing goals and assess the results. Furthermore, multiple comparative analyses with other well-known optimizers may be carried out to find the best optimizer for a specific design problem.

## ACKNOWLEDGMENT

The authors would like to thank the editor and blind reviewer for their effort in shaping the article in good form.

## APPENDIX

The "sorting by fitness" concept in MOPGO has been explained with one simple multi-objective benchmark function. The main aim of this appendix is to help the researchers in other fields to use in their field of research. The

**TABLE 13.** Initial population solution.

| Initial population, $P_t$ | | | | | Offspring population (PGO), $Q_t$ | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Solution | $x_1$ | $x_2$ | $f_1$ | $f_2$ | Solution | $x_1$ | $x_2$ | $f_1$ | $f_2$ |
| 1 | 0.31 | 0.89 | 0.31 | 6.10 | a | 0.21 | 0.24 | 0.21 | 5.90 |
| 2 | 0.43 | 1.92 | 0.43 | 6.79 | b | 0.79 | 2.14 | 0.79 | 3.97 |
| 3 | 0.22 | 0.56 | 0.22 | 7.09 | c | 0.51 | 2.32 | 0.51 | 6.51 |
| 4 | 0.59 | 3.63 | 0.59 | 7.85 | d | 0.27 | 0.87 | 0.27 | 6.93 |
| 5 | 0.66 | 1.41 | 0.66 | 3.65 | e | 0.58 | 1.62 | 0.58 | 4.52 |
| 6 | 0.83 | 2.51 | 0.83 | 4.23 | f | 0.24 | 1.05 | 0.24 | 8.54 |

**TABLE 14.** Sorting by first objective fitness.

| Front 2 | | | | | Sorting in | | CD |
|---|---|---|---|---|---|---|---|
| Solution | $x_1$ | $x_2$ | $f_1$ | $f_2$ | $f_1$ | $f_2$ | |
| **1** | 0.31 | 0.89 | 0.31 | 6.10 | **Third** | Second | 0.63 |
| 3 | 0.22 | 0.56 | 0.22 | 7.09 | **First** | Fourth | Infinite |
| b | 0.79 | 2.14 | 0.79 | 3.97 | **Fourth** | First | Infinite |
| d | 0.27 | 0.87 | 0.27 | 6.93 | **Second** | Third | 0.12 |

multi-objective benchmark function is as follows.

$$\left.\begin{aligned} &\textbf{Minimize, } \mathbf{f_1}(\mathbf{x}) = \mathbf{x_1} \\ &\textbf{Minimize, } \mathbf{f_2}(\mathbf{x}) = \frac{\mathbf{1}+\mathbf{x_2}}{\mathbf{x_1}} \\ &\textbf{Subjected to: } \mathbf{0.1} \le \mathbf{x_1} \le \mathbf{1} \\ &\mathbf{0} \le \mathbf{x_2} \le \mathbf{5} \end{aligned}\right\}$$

Table 13 lists the initial solutions generated by the population.

The following are steps in generating the solutions based on non-dominated sorting.

**Step 1:** First combine the populations $P_t$ and $Q_t$ and form $R_t = \{1,2, 3, 4, 5, 6, a, b, c, d, e, f\}$. Next, perform a non-dominated sorting on $R_t$ and obtain the following non-dominated fronts: $F_1\{5, a, e\}$, $F_2\{1,3, b, d\}$, $F_3\{2, 6, c, f\}$, $F_4\{4\}$.

**Step 2:** Set $P_t + 1 = 0$ and $i = 1$. Next, observe that $!P_t + 1! + !F_1! = 0 + 3 = 3$. Since this is less than the population size N (= 6), include this front in $P_t + 1$ and set $P_t + 1 = \{5, a, e\}$. With these three solutions, we now need three more solutions to fill up the new parent population. Now, with the inclusion of the second front, the size of $!Pt + 2! + !F2!$ is $(3 + 4)$ or 7. Since this is greater than 6, we stop including any more fronts into the population.

**Step 3:** Next, consider solutions of the second front only and observe that 3 of the 4 solutions must be chosen to fill up 3 remaining slots in the new population. This requires that first sort this sub-population (solutions 1, 3, a, and d) using the CD operator.

For the first objective fitness, the sorting of these solutions is shown in Table 14 and is as follows: $I_1 = \{3, d, 1, b\}$.

Now, we turn to the second objective fitness and update the above distances. First, the sorting on this objective yields

**TABLE 15.** Sorting by second objective fitness.

| Front 2 | | | | | Sorting in | | CD |
|---|---|---|---|---|---|---|---|
| Solution | $x_1$ | $x_2$ | $f_1$ | $f_2$ | $f_1$ | $f_2$ | |
| **1** | 0.31 | 0.89 | 0.31 | 6.10 | Third | **Second** | 0.63 |
| 3 | 0.22 | 0.56 | 0.22 | 7.09 | First | **Fourth** | Infinite |
| b | 0.79 | 2.14 | 0.79 | 3.97 | Fourth | **First** | Infinite |
| d | 0.27 | 0.87 | 0.27 | 6.93 | Second | **Third** | 0.12 |

$I_2 = \{b, 1, d, 3\}$. Thus, Table 15 lists the solution sorting of second objective fitness.

**Step 4:** A sorting according to the descending order of these CD values yields the sorted set $\{3, b, 1, d\}$. Then, choose the first three solutions. The new population is $P_t + 1 = \{5, a, e, 3, b, 1\}$.

**Step 5:** The offspring population $Q_t + 1$ must be created next by using this parent population $P_t + 1 = \{5, a, e, 3, b, 1\}$. This is a complete one generation of MOPGO.

## REFERENCES

[1] S. Kumar, G. G. Tejani, N. Pholdee, S. Bureerat, and P. Mehta, "Hybrid heat transfer search and passing vehicle search optimizer for multi-objective structural optimization," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106556.
[2] R. Moghdani, K. Salimifard, E. Demir, and A. Benyettou, "Multi-objective volleyball premier league algorithm," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105781.
[3] G. Dhiman and V. Kumar, "Multi-objective spotted hyena optimizer: A multi-objective optimization algorithm for engineering problems," *Knowl.-Based Syst.*, vol. 150, pp. 175–197, Jun. 2018.
[4] S. Kumar, G. G. Tejani, N. Pholdee, and S. Bureerat, "Multi-objective passing vehicle search algorithm for structure optimization," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114511.
[5] S. Kumar, G. G. Tejani, N. Pholdee, and S. Bureerat, "Multi-objective modified heat transfer search for truss optimization," *Eng. Comput.*, vol. 37, pp. 641–662, 2021.
[6] H. Li, L. Zhang, B. Huang, and X. Zhou, "Sequential three-way decision and granulation for cost-sensitive face recognition," *Knowl.-Based Syst.*, vol. 91, pp. 241–251, Jan. 2016.
[7] K. Deb and S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms," *Finite Elements Anal. Des.*, vol. 37, no. 5, pp. 447–465, May 2001.
[8] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Nov. 1995, pp. 1942–1948.
[10] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2, Jul. 1999, pp. 1470–1477.
[11] 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, pp. 182–197, Apr. 2002.
[12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Institut für Technische Informatik und Kommunikationsnetze, Switzerland, vol. 103.
[13] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. 3rd Annu. Conf. Genet. Evol. Comput.*, Jul. 2001, pp. 283–290.
[14] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 1, Jul. 1999, pp. 98–105.
[15] G. G. Tejani, S. Kumar, and A. H. Gandomi, "Multi-objective heat transfer search algorithm for truss optimization," *Eng. Comput.*, vol. 37, pp. 1–22, Aug. 2019.
[16] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1616–1624, Jun. 2013.

[17] C. A. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2, May 2002, pp. 1051–1056.
[18] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[19] T. Murata and H. Ishibuchi, "MOGA: Multi-objective genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, vol. 1, Nov. 1995, pp. 289–294.
[20] S. Kumar, G. G. Tejani, and S. Mirjalili, "Modified symbiotic organisms search for structural optimization," *Eng. Comput.*, vol. 35, no. 4, pp. 1269–1296, 2019.
[21] 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," *Knowl.-Based Syst.*, vol. 134, pp. 50–71, Oct. 2017.
[22] S. Kumar, G. G. Tejani, N. Pholdee, and S. Bureerat, "Improved metaheuristics through migration-based search and an acceptance probability for truss optimization," *Asian J. Civil Eng.*, vol. 21, no. 7, pp. 1217–1237, Nov. 2020.
[23] S. Mirjalili, P. Jangir, and S. Saremi, "Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems," *Appl. Intell.*, vol. 46, no. 1, pp. 79–95, Jan. 2017.
[24] S. Kumar, G. G. Tejani, N. Pholdee, and S. Bureerat, "Multiobjecitve structural optimization using improved heat transfer search," *Knowl.-Based Syst.*, vol. 219, May 2021, Art. no. 106811.
[25] G. G. Tejani, N. Pholdee, S. Bureerat, and D. Prayogo, "Multiobjective adaptive symbiotic organisms search for truss optimization problems," *Knowl.-Based Syst.*, vol. 161, pp. 398–414, Dec. 2018.
[26] P. Jangir and N. Jangir, "A new non-dominated sorting grey wolf optimizer (NS-GWO) algorithm: Development and application to solve engineering designs and economic constrained emission dispatch problem with integration of wind power," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 449–467, Jun. 2018.
[27] M. Premkumar, P. Jangir, R. Sowmya, R. M. Elavarasan, and B. S. Kumar, "Enhanced chaotic JAYA algorithm for parameter estimation of photovoltaic cell/modules," *ISA Trans.*, pp. 1–28, Jan. 2021, doi: 10.1016/j.isatra.2021.01.045.
[28] P. Jangir, S. A. Parmar, I. N. Trivedi, and R. H. Bhesdadiya, "A novel hybrid particle swarm optimizer with multi verse optimizer for global numerical optimization and optimal reactive power dispatch problem," *Eng. Sci. Technol., Int. J.*, vol. 20, no. 2, pp. 570–586, Apr. 2017.
[29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[30] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
[31] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.
[32] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *Int. J. Ind. Eng. Comput.*, vol. 7, no. 1, pp. 19–34, 2016.
[33] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[34] M. Premkumar, R. Sowmya, P. Jangir, K. S. Nisar, and M. Aldhaifallah, "A new metaheuristic optimization algorithms for brushless direct current wheel motor design problem," *Comput., Mater. Continua*, vol. 67, no. 2, pp. 2227–2242, 2021.
[35] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
[36] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
[37] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
[38] M. Premkumar, P. Jangir, C. Ramakrishnan, G. Nalinipriya, H. H. Alhelou, and B. S. Kumar, "Identification of solar photovoltaic model parameters using an improved gradient-based optimization algorithm with chaotic drifts," *IEEE Access*, vol. 9, pp. 62347–62379, 2021.
[39] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[40] A. Kaveh, H. Akbari, and S. M. Hosseini, "Plasma generation optimization: A new physically-based metaheuristic algorithm for solving constrained optimization problems," *Eng. Comput.*, Oct. 2020.
[41] K. Deb and T. Goel, "Controlled elitist non-dominated sorting genetic algorithms for better convergence," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, Mar. 2001, pp. 67–81.
[42] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Proc. Genet. Evol. Comput. Conf.* Berlin, Germany: Springer, Jul. 2003, pp. 37–48.
[43] H. A. Abbass, R. Sarker, and C. Newton, "PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems," in *Proc. Congr. Evol. Comput.*, vol. 2, May 2001, pp. 971–978.
[44] S. Vinodh, S. Sarangan, and S. Chandra Vinoth, "Application of fuzzy compromise solution method for fit concept selection," *Appl. Math. Model.*, vol. 38, no. 3, pp. 1052–1063, Feb. 2014.
[45] M. Premkumar, P. Jangir, R. Sowmya, H. H. Alhelou, A. A. Heidari, and H. Chen, "MOSMA: Multi-objective slime mould algorithm based on elitist non-dominated sorting," *IEEE Access*, vol. 9, pp. 3229–3248, 2021, doi: 10.1109/ACCESS.2020.3047936.
[46] M. Premkumar, P. Jangir, and R. Sowmya, "MOGBO: A new multiobjective gradient-based optimizer for real-world structural optimization problems," *Knowl.-Based Syst.*, vol. 218, Apr. 2021, Art. no. 106856.
[47] M. Premkumar, P. Jangir, B. S. Kumar, R. Sowmya, H. H. Alhelou, L. Abualigah, A. R. Yildiz, and S. Mirjalili, "A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: Diversity analysis and validations," *IEEE Access*, early access, Jun. 2, 2021, doi: 10.1109/ACCESS.2021.3085529.
[48] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

**SUMIT KUMAR** received the B.E. degree (Hons.) in mechanical engineering from Dr. A.P.J. Abdul Kalam Technical University, Lucknow, India, in 2012, and the M.E. degree (Hons.) in design engineering from the Malaviya National Institute of Technology (NIT), Jaipur, India, in 2015. He is currently a Ph.D. Research Scholar with the College of Sciences and Engineering, Australian Maritime College, University of Tasmania, Launceston, Australia. He has more than six years of teaching experience. He has published technical articles in various national/international peer-reviewed journals such as *Knowledge-Based Systems* (Elsevier), *Expert Systems With Applications* (Elsevier), *Engineering With Computers* (Springer), and *Archives of Computational Methods in Engineering* (Springer). His major research interests include metaheuristics techniques, multi-objective optimization, evolutionary algorithm, and renewable energy systems. He is a member of many professional bodies like ISHRAE. He has received numerous awards and grants. He is serving as a Reviewer for leading journals such as Elsevier and Springer.

**PRADEEP JANGIR** is currently the Co-Director of the Zero Lab Optimization. He is a Power Engineer at RVPN, Jaipur. He is internationally recognized for his advances in swarm intelligence and optimization. He has published over 76 publications with over 1 000 citations and an H-index of 15. His research interests include many-objective, robust optimization, power system engineering optimization, multi-objective optimization, swarm intelligence, evolutionary algorithms, and artificial neural networks. He is working on the application of multi-objective, many-objective, and robust meta-heuristic optimization techniques.

**GHANSHYAM G. TEJANI** received the bachelor's and master's degrees in mechanical engineering (specialization machine design) from GTU, in 2003 and 2012, respectively, and the Ph.D. degree from PDPU, in 2017. He is currently an Assistant Professor with the Mechanical Engineering Department, School of Technology, GSFC University. He has published more than 22 research articles in high-quality peer-reviewed journals and international conferences. He received the Outstanding Reviewer Award from *Knowledge-Based Systems* (IF: 5.921; Elsevier; ScienceDirect), in 2018. He received the Double Gold Medal from GTU for his bachelor's and master's degree. He is serving as an editorial board member in five leading journals. He has performed more than 110 reviews for high prestigious journals, including IEEE Access, *Knowledge-Based Systems*, *Applied Mathematics and Computation*, *Neural Computing and Applications*, *Materials and Design*, and *Engineering Structures*. His research article is the second most cited in *Journal of Computational Design and Engineering* (IF: 3.408; Elsevier; ScienceDirect) for the period 2018–2021.

**MANOHARAN PREMKUMAR** (Member, IEEE) was born in Coimbatore, India. He received the B.E. degree in electrical and electronics engineering from the Sri Ramakrishna Institute of Technology, Coimbatore, in 2004, the M.E. degree in applied electronics from the Anna University of Technology, Coimbatore, in 2010, and the Ph.D. degree from Anna University, Chennai, India, in 2019. He is currently working as an Associate Professor with the Dayananda Sagar College of Engineering, Bengaluru, India. He has more than 12 years of teaching experience. He has published more than 70 technical articles in various national/international peer-reviewed journals such as IEEE, Elsevier, and Springer. He has over 350 citations and an H-index of 11. He has published/granted four patents accepted and approved by IPR, India, and IPR, Australia. His current research interests include optimization techniques, including single-, multi-, and many-objectives, solar PV microinverter, solar PVparameter extraction, modern solar PVMPPTs (optimization technique based), PV array faults, and non-isolated/isolated dc–dc converters for PV systems. He is a member of various professional bodies such as IEEE, ISTE, and IAENG. He is serving as an Editor/a Reviewer for leading journals such as IEEE, IET, Wiley, Taylor & Francis, and Springer.

**HASSAN HAES ALHELOU** (Senior Member, IEEE) is currently a Faculty Member at Tishreen University, Lattakia, Syria. He is included in the 2018 Publons list of the top 1% best reviewer and researchers in the field of engineering in the world. He has published more than 130 research articles in high-quality peer-reviewed journals and international conferences. He has performed more than 600 reviews for highly prestigious journals, including IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics, *Energy Conversion and Management*, *Applied Energy*, and *International Journal of Electrical Power and Energy Systems*. He has participated in more than 15 industrial projects. His major research interests include power systems, power system dynamics, power system operation and control, dynamic state estimation, frequency control, smart grids, micro-grids, demand response, and load shedding. He was a recipient of the Outstanding Reviewer Award from many journals such as *Energy Conversion and Management (ECM)*, *ISA Transactions*, and *Applied Energy*. He was a recipient of the Best Young Researcher in the Arab Student Forum Creative among 61 researchers from 16 countries at Alexandria University, Egypt, in 2011.