Archives of Computational Methods in Engineering (2021) 28:4031–4047
https://doi.org/10.1007/s11831-021-09531-8

ORIGINAL PAPER

A Comparative Study of Recent Multi-objective Metaheuristics for Solving Constrained Truss Optimisation Problems

Natee Panagant[1] · Nantiwat Pholdee[1] · Sujin Bureerat[1] · Ali Riza Yildiz[2] · Seyedali Mirjalili[3,4]

Received: 27 April 2020 / Accepted: 4 January 2021 / Published online: 16 January 2021

Abstract
Multi-objective truss optimisation is a research topic that has been less investigated in the literature compared to the single-objective cases. This paper investigates the comparative performance of fourteen new and established multi-objective metaheuristics when solving truss optimisation problems. The optimisers include multi-objective ant lion optimiser, multi-objective dragonfly algorithm, multi-objective grasshopper optimisation algorithm, multi-objective grey wolf optimiser, multi-objective multi-verse optimisation, multi-objective water cycle algorithm, multi-objective Salp swarm algorithm, success history-based adaptive multi-objective differential evolution, success history–based adaptive multi-objective differential evolution with whale optimisation, non-dominated sorting genetic algorithm II, hybridisation of real-code population-based incremental learning and differential evolution, differential evolution for multi-objective optimisation, multi-objective evolutionary algorithm based on decomposition, and unrestricted population size evolutionary multi-objective optimisation algorithm. The design problem is assigned to minimise structural mass and compliance subject to stress constraints. Eight classical trusses found in the literature are used for setting up the design test problems. Various optimisers are then implemented to tackle the problems. A comprehensive comparative study is given to critically analyse the performance of all algorithms in this problem area. The results provide new insights to the pros and cons of evolutionary multi-objective optimisation algorithms when addressing multiple, often conflicting objective in truss optimisation. The results and findings of this work assist with not only solving truss optimisation problem better but also designing customised algorithms for such problems.

1 Introduction

Truss optimisation has become a popular research topic for decades as there have been over hundreds of publications each year recently. The research in truss optimisation has several aspects of studies such as problem formulation and development of optimisation algorithms. The optimisation problem can be categorised as single-objective and multi-objective optimisation. Furthermore, the special cases of multi-objective truss optimisation having more than three objective functions being assigned are called many-objective truss design [1]. Traditional design objectives in literature are mass, displacement, compliance, natural frequencies, and frequency response function. Design variables, on the other hand, can be topological, shape and sizing optimisation. Topology design variables determine truss initial layout while shape design parameters specify nodal positions. Sizing variables will assign the truss elements' cross-sectional areas. Design constraints usually include stress, displacement, bifurcation buckling, and natural frequencies. Such constraints often lead to non-convex feasible regions, which means some gradient-based optimisers may struggle to find the optima, as a result, metaheuristics (MHs) are used as alternative optimisation solvers due to their high flexibility in coding and implementing.

The use of MHs is a more popular choice for truss optimisation because of their advantages in a derivative-free feature, robustness, simplicity to understand, and high flexibility in coding and implementation. One of the most

✉ Sujin Bureerat
sujbur@kku.ac.th

1 Sustainable Infrastructure Research and Development Centre, Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand

2 Department of Automotive Engineering, Bursa Uludağ University, Görükle, Bursa, Turkey

3 Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, 90 Bowen Terrace, Brisbane, QLD 4006, Australia

4 YFL (Yonsei Frontier Lab), Yonsei University, Seoul, Korea

outstanding features is that MHs can explore a Pareto front of a multi-objective optimisation within one optimisation run. Apart from that, they can deal with a wide range of design variables and functions although they may be less efficient in some cases [2–8]. Truss design with discrete shape and sizing variables, which is more practical, is possible with using MHs. They are capable of solving design problems with concurrent topology, shape and sizing design variables while some non-differentiable constraints can be imposed [9].

The majority of research work on truss optimisation focused on single-objective design with most if not all assigning structural mass as a design objective while the constraints are set for safety requirements of the truss under several loading conditions. A large number of metaheuristics have been implemented on such a constrained design problem. In the early days, genetic algorithms (GA) [10, 11] and simulated annealing [12] were arguably the most popularly used MH. Then, it came to the era of particle swarm optimisation [13] and differential evolution (DE) [14]. Up to the present time, there has been a great number of algorithms being developed and used. Those include teaching learning based optimisation [15], artificial bee colony algorithm [16], charged system search [17], firefly algorithm [18], colliding body optimisation algorithm [19], bat algorithm [20], krill herd algorithm [21], ray optimisation [22], modified symbiotic organisms search [23], hybridised passing vehicle search and simulated annealing [24], and the list goes on. Meanwhile, Pholdee and Bureerat [25] bridged the gap between the fields of metaheuristics in computer science and truss optimisation by examining the performance of top self-adaptive metaheuristics mostly the winner of the annual completion at the congress on evolutionary computation (CEC competitions). It was found that those self-adaptive MHs are powerful for truss optimisation. The comparative performance of new algorithms should be studied as new MHs have been developed almost every month. That means new powerful algorithms could be noticeable.

For multi-objective truss optimisation, there has been much less research work compared to the single-objective cases. Nevertheless, muti-objective design is advantageous as more design objective can be added to the design problem. In real applications, it is common that designers are interested in optimising several design criteria so that a set of Pareto optimal solutions can be obtained for further decision making. Moreover, in cases of structural reliability optimisation, it is more effective to pose the design problem as multi-objective optimisation with a reliability index being one of the objective functions [26–28]. With the capability of solving multi-objective optimisation within one run and less time-consuming function evaluations of truss optimisation, MHs are good choices to be used for multi-objective optimisation of trusses.

Simultaneous topology, shape and sizing optimisation with many objective functions for static and dynamic applications of the structure was presented in [9]. The objective functions include mass, compliance, natural frequency, frequency response function and vibration force transmission. There have been a number of multi-objective metaheuristics (MOMHs) implemented on multi-objective truss optimisation. Those include: multi-objective evolutionary algorithms using an approximate gradient [29], hybridisation of real-code population-based incremental learning and differential evolution [30], hybrid real-code population-based incremental learning and approximate gradients [31], multi-objective adaptive symbiotic organisms search [32], multi-objective modified heat transfer search [33]. The comparative study of several MOMHs for truss design has been made e.g. in [27], however, as there have been a great number of new MOMHs recently, the comparative performance of them for truss design should be studied so as to update the knowledge in the field.

This paper presents comparative performance studies of MOMHs for truss optimisation. The design problem is posed to optimise structural mass and compliance subject to stress constraints. Eight traditional truss benchmark structures are used to create the test problems. MOMHs including: multi-objective ant lion optimiser (MALO) [34], multi-objective dragonfly algorithm (MODA) [35], multi-objective grasshopper optimisation algorithm (MOGOA) [36], multi-objective grey wolf optimiser (MOGWO) [37], multi-objective multi-verse optimisation (MOMVO) [38], multi-objective water cycle algorithm (MOWCA) [39], multi-objective Salp swarm algorithm (MSSA) [40], success history-based adaptive multi-objective differential evolution (SHAMODE) [28], success history-based adaptive multi-objective differential evolution with whale optimisation (SHAMODE-WO) [28], non-dominated sorting genetic algorithm II (NSGA-II) [41], hybridisation of real-code population-based incremental learning and differential evolution (RPBILDE) [30], differential evolution for multi-objective optimisation (DEMO) [42], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [43], and unrestricted population size evolutionary multi-objective optimisation algorithm (UPS-EMOA) [44] are used in this investigation while four performance indicators are employed. The rest of the paper is organised as follows: Sect. 2 presents the truss design problems. The optimisation algorithms used in this work are briefly presented in Sect. 3. Section 4 discusses and analyses the results. Finally, Sect. 5 concludes the work and suggest future directions.

2 Optimisation Problems

In this paper, eight truss optimisation problems are gathered from [23, 30]. All the test problems are converted to SI unit while some problems are modified, as a result, most of the truss optimisation problems evaluated in this study are different from their original problems. We employed eight multi-objective truss sizing optimisation problems, 10-bar, 25-bar, 37-bar, 60-bar, 72-bar, 120-bar, 200-bar, and 942-bar evaluated. Structural mass and compliance are assigned as objective functions subject to allowable stress constraints. The problems are described in Eq. 1 whereas the compliance is calculated using Eq. 2. Displacement and loading vectors in the equation are employed from finite element analysis.

$$\min \left(f_1, f_2\right) \quad s.t. \sigma_{max} \sigma_a \tag{1}$$

$$compliance = \boldsymbol{u}^T \boldsymbol{F} \tag{2}$$

where f_1 and f_2 re structural mass and compliance respectively. σ_{max} and σ_a are maximum stress occurs on the structure and allowable stress respectively. Material properties and allowable stress of all test problems are equally specified. Density, modulus of elasticity, and allowable stress are set as 7850 kg/m^3, 200 GPa, and 400 MPa respectively. In practice, the size of each member of the structures are usually discrete design variables due to beam standard sizing, therefore, the sizing variables are assigned as being discrete in this study. The ground structure of the 10-bar, 25-bar, 37-bar, 60-bar, 72-bar, 120-bar, 200-bar, and 942-bar structures are displayed in Figs. 1, 2, 3, 4, 5, 6, 7 and 8 respectively. The 10-bar, 37-bar, 60-bar and 200-bar test problems in Fig. 1, 3, 4, and 7 are 2D or plane trusses while the 25-bar, 72-bar, 120-bar, and 942-bar in Figs. 2, 5, 6, and 8 are 3D or space trusses.

There are grouped design variables in some test problems, thus, the number of design variables may not equal to the number of truss members. The numbers of design variables of the 10-bar, 25-bar, 37-bar, 60-bar, 72-bar, 120-bar, 200-bar, and 942-bar problems are 10, 8, 15, 25, 16, 7, 29, and 59 respectively.

Four performance metrics, hypervolume (HV), Generational Distance (GD) [45], Inverted Generational Distance (IGD) [46], and Spacing-to-Extent (STE) [23] employed to measure the performance of the optimisation algorithms. HV is used to measure the spread of a Pareto front while STE is ratio between spacing and extent of a front. GD and IGD, on the other hand, are used to measure distances between an obtained Pareto front and a reference front. The reference front can be a true Pareto front of the optimisation problem being solved or a non-dominated front where its members are not dominated by any non-dominated solution obtained from the optimisers being compared.

HV is an area (or hypervolume in cases of having more than two objective functions) between a reference point and a Pareto front as illustrated in Fig. 2. From Fig. 2, the reference point is on the top-right corner of the figure. Points with the x-marker are solutions in the Pareto front (sometimes referred to as non-dominated solutions) while the grey-highlighted area between the reference point and the Pareto front is the hypervolume of the front. Reference points of the test problems used in this study are maximum objective functions found from all algorithms and all optimisation runs. The more superior front has higher HV or a larger area.

For GD and IGD, reference fronts are needed. In fact, the original definitions of GD and IGD require the true Pareto front of an optimisation problem as a reference front. However, such a front is not available in this study. The reference front of a particular test problem herein is obtained in such a way that, having run each algorithm M_1 times for all M_2 algorithms, all non-dominated solutions from the obtained M_1M_2 fronts are put together. Then, the non-dominated solutions of those combined M_1M_2 fronts are sorted and selected to be the members of the reference front. The indicators GD and IGD can be computed using Eqs. 3 and 4 respectively.

$$GD = \frac{\sqrt{\sum_{i=1}^{|P|} (d_i)^2}}{|P|} \tag{3}$$

where $|P|$ is the number of solutions in the obtained Pareto front and d_i is the Euclidian distance of the objective functions vector of the ith solution in the obtained front to its nearest solution from the reference front.

$$IGD = \frac{\sqrt{\sum_{i=1}^{|P'|} (d'_i)^2}}{|P'|} \tag{4}$$

where $|P'|$s the number of solutions in the reference front and d'_is Euclidian distance of objective functions vector of the ith solution in the reference front to its nearest solution from the obtained front. IGD, by its definition, can measure both front advancement and extension.

STE is the ratio between front spacing and extent. Calculation of STE of a Pareto front is given in Eqs. 5–7.

$$Spacing = \frac{1}{|P| - 1} \sum_{i=1}^{|P|} (d_i - \bar{d})^2 \tag{5}$$

$$Extent = \sum_{i=1}^{M} \left| f_i^{max} - f_i^{min} \right| \tag{6}$$

$$STE = Spacing/Extent \tag{7}$$

where $|P|$ is number of solutions in the obtained Pareto front, d_i is the Euclidian distance of objective functions vector of the ith solution to its nearest neighbour, $\bar{d}$ is the mean value of all di M is the number of objective functions, f_i^{max} and f_i^{min} are respectively the maximum and minimum values of

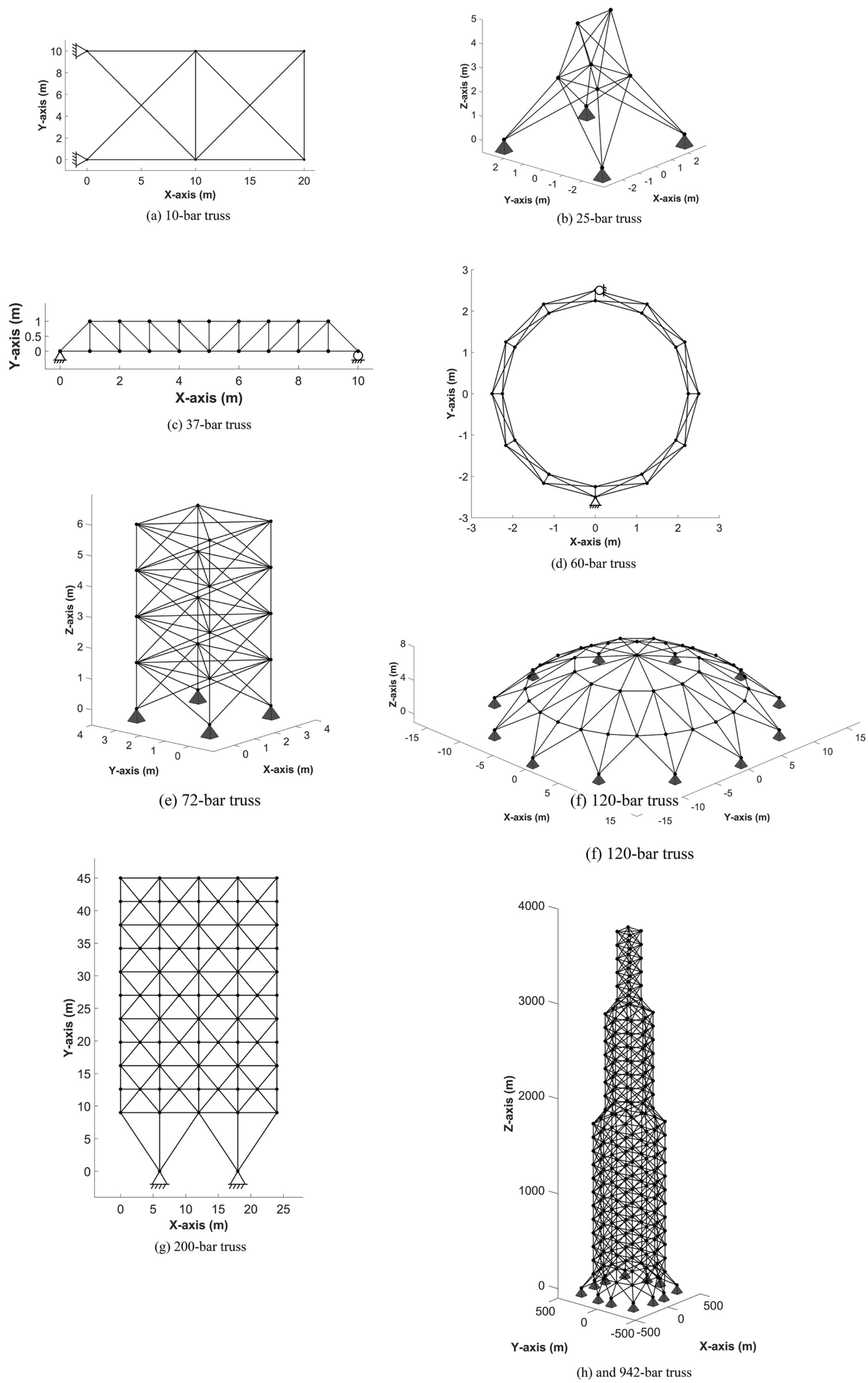


Fig. 1 Ground structures of 10-bar truss, 25-bar truss, 37-bar truss, 60-bar truss, 72-bar truss, 120-bar truss, 200-bar truss, and 942-bar truss

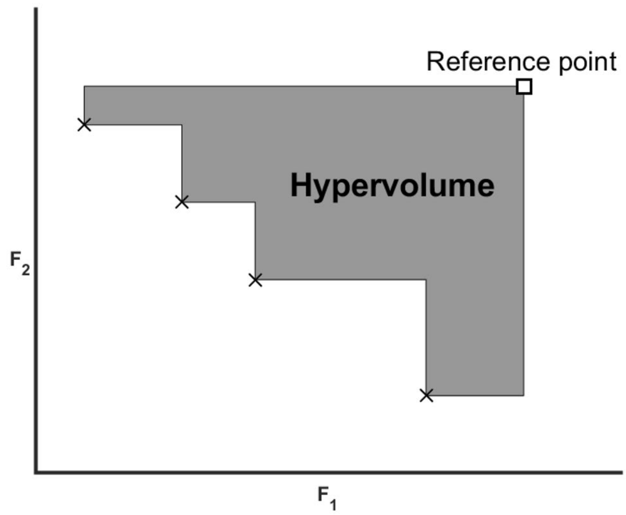


Fig. 2 The Hyper Volume performance indicator for quantifying the coverage of multi-objective algorithms

the *i*th objective function of the front. A superior front can be indicated with higher HV whereas fronts lower value of GD, IGD, and STE means the better fronts.

3 Optimisation algorithms

Multio-bjective metaheuristics work based upon the use of a set of solutions (individuals) which is traditionally called a population. The population is then iteratively improved by means of improving mechanisms (e.g. reproduction and selection) until a termination condition is met. For multi-objective optimisation, metaheuristics using a population of design solutions have some advantage as one can sort and find non-dominated solutions from a population. This idea leads to the use of MHs for exploring a Pareto front of multi-objective optimisation within one run. There have been several ideas employed for dealing with multi-objective metaheuristic search. Those include a Pareto dominance concept [30], elitism [41], and decomposition based concept [43]. Most of the algorithms, however, exploit the non-dominated sorting scheme for identifying non-dominated solutions. For minimisation, solution $\mathbf{x}^1$ is said to dominate solution $\mathbf{x}^2$ if its objective vector $\mathbf{f}^1$ dominates $\mathbf{f}^2$. This is true if at least one element in $\mathbf{f}^1$ is strictly lower than the corresponding element in $\mathbf{f}^2$ and all elements in $\mathbf{f}^1$ are not higher than those corresponding elements in $\mathbf{f}^2$. For a set of solutions or a population any individual that is not dominated by the others is defined as a non-dominated solution in the set. With these definitions, a set of non-dominated solutions can be set and improve iteratively to reach the true Pareto front.

To estimate the Pareto optimal solutions for the truss design problems discussed above, we employ the following recently developed and well-known algorithms:

- Multi-objective ant lion optimiser (MOALO) [34]
  MOALO uses the relationship principals of ants and antlions in nature. The multi-objective mechanisms of MOLAO are similar to those in MOPSO: an archive to store non-dominated solutions during the optimisation process and a leader selection mechanism to select ‘best’ non-dominated solutions for position updating of other solutions. A niching mechanism is used to help with choosing the leaders.
- Multi-objective dragonfly algorithm (MODA) [35]
  The DA algorithm is inspired by the rare swarming behaviours of dragonflies in nature. In the multi-objective version (MODA), non-dominated solutions are stored in a repository. Such solutions are then selected based on a grid selection mechanism to be used in the main equations of the DA algorithm.
- Multi-objective grasshopper optimisation algorithm (MOGOA) [36]
  The GOA algorithm was proposed with inspiration from the swarming behaviour of grasshoppers in nature. The original version of this algorithm requires a set of solution to update their positions in an n-dimensional space, which is defined by the problem. To solve multi-objective problems using this algorithm, it was equipped with an archive and leader selection mechanisms.
- Multi-objective grey wolf optimiser (MOGWO) [37]
  As a recent optimisation algorithm, GWO mimics the social hierarchy and hunting mechanisms of grey wolves in nature. This algorithm requires the three best solutions (alpha, beta, and delta) to update the position of other solutions. To solve multi-objective problems, these three solutions are selected from non-dominated solutions during the optimisation process.
- Multi-objective multi-verse optimisation (MOMVO) [38]
  The inspiration of the MVO algorithm is the theory of multi-verse in physics. The multi-objective version of this algorithm is developed using similar mechanisms of MOGWO: archive and selector. There is also a method to maintain the diversity of solutions in the archive.
- Multi-objective water cycle algorithm (MOWCA) [39]
  As its name suggests, the WCA algorithm simulates the water cycle in nature. A crowding distance is used in this algorithm to choose non-dominated solutions.
- Multi-objective Salp swarm algorithm (MSSA) [40]
  The SSA algorithm is inspired by the swarming patterns of salps in ocean, in which such creatures create a chain to navigate and forage. There is a leader that plays a critical role in this algorithm, which is one solution when solving single-objective problems. The MSSA algorithm

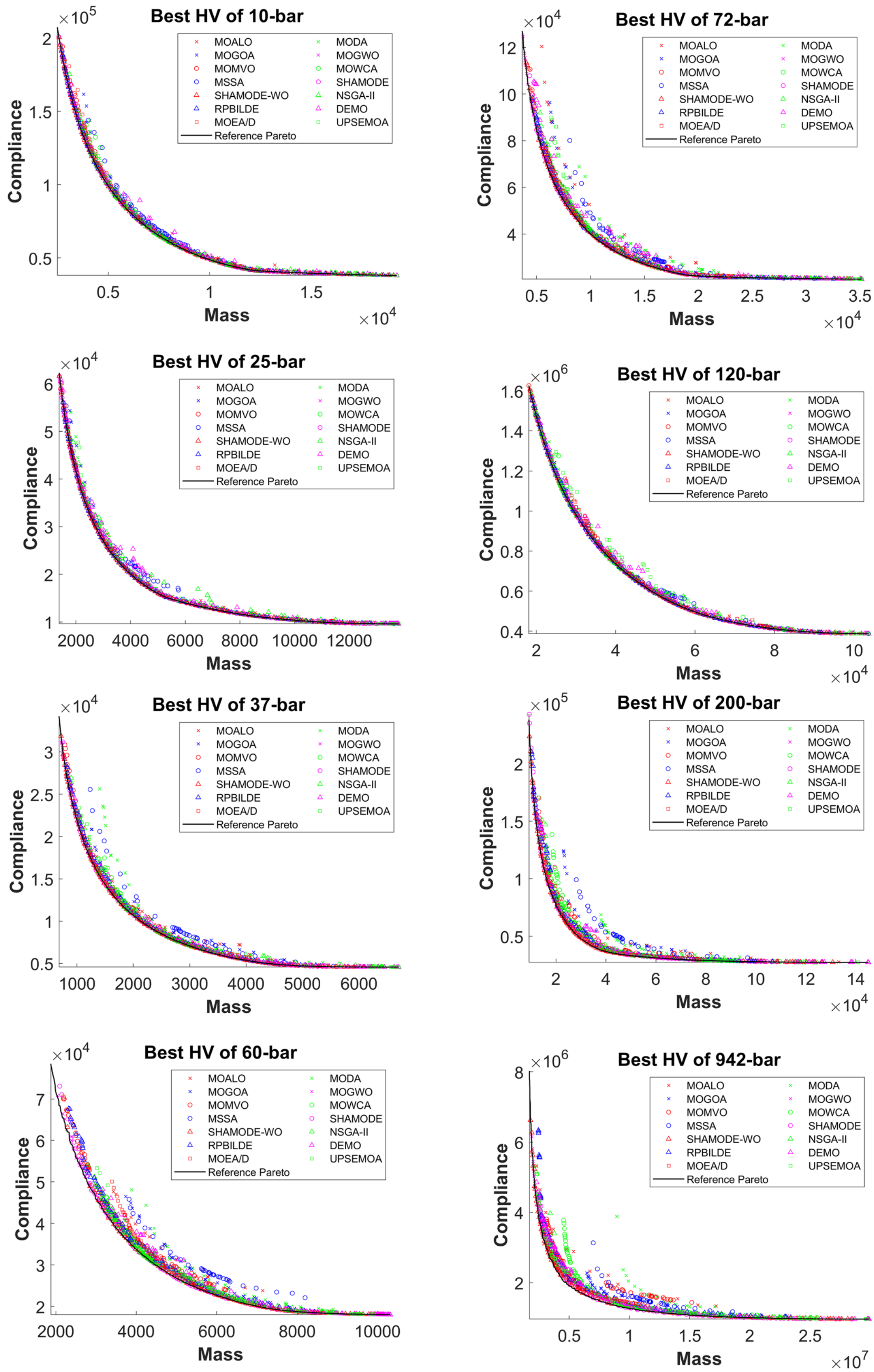


Fig. 3 Pareto fronts obtained by the algorithms for the 10-bar truss problem, 25-bar truss problem, 37-bar truss problem, 60-bar truss problem, 72-bar truss problem, 120-bar truss problem, and 200-bar truss problem

Fig. 4 Mean HV of the 10-bar truss problem

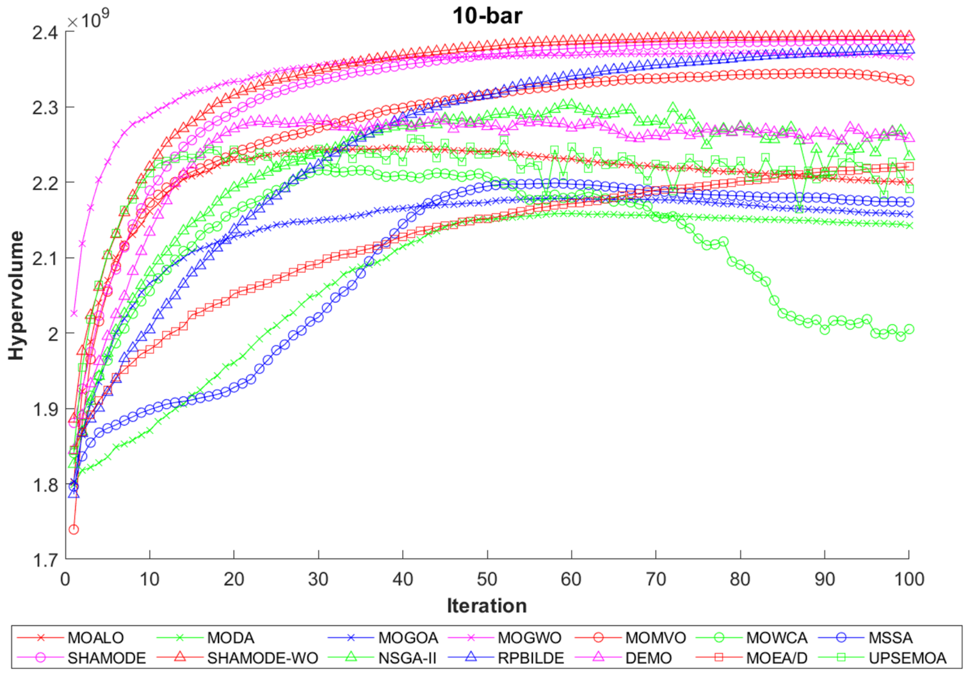
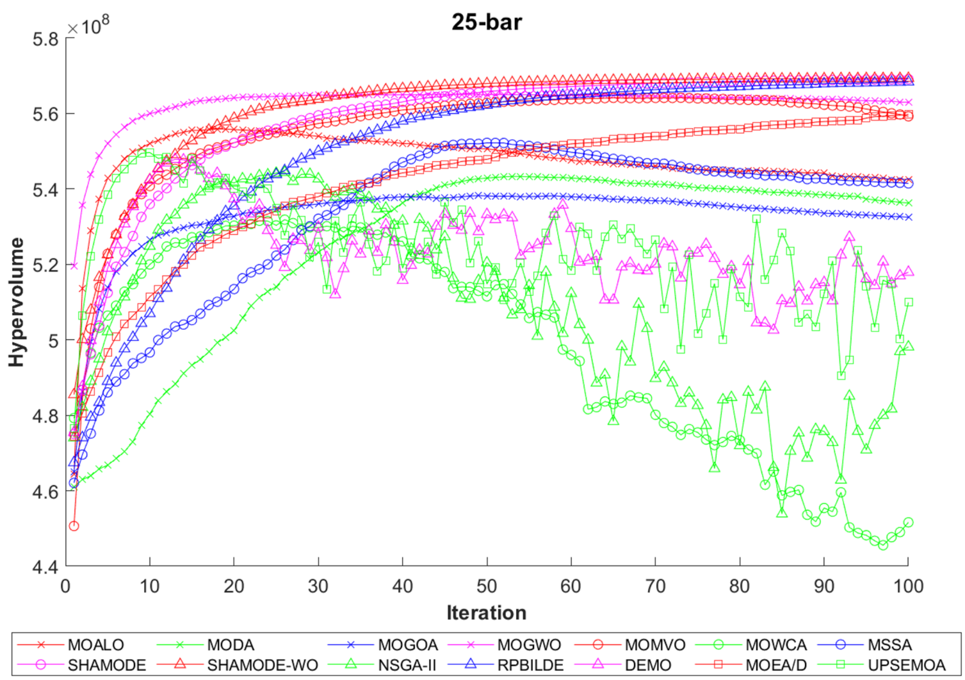


Fig. 5 Mean HV of the 25-bar truss problem



uses the same equations in SSA but the leader is selected from an archive of non-dominated solutions.

- Success history–based adaptive multi-objective differential evolution (SHAMODE) [28]

The optimiser exploits the successive history-based adaptive strategy for tuning the control parameters of differential evolution. The non-dominated sorting operator is used for collecting non-dominated solutions during and optimisation search. The procedure starts with a population and an initial Pareto archive. The population and the archived are iteratively updated using DE operator with the self-adaptive DE parameters until reaching the termination criterion.

- Success history–based adaptive multi-objective differential evolution with whale optimisation (SHAMODE-WO)

SHAMODE-WO is a hybrid algorithm that integrates the spiral movement of whale optimisation algorithm to the DE binomial crossover. Instead of using a mutant

Fig. 6 Mean HV of the 37-bar truss problem

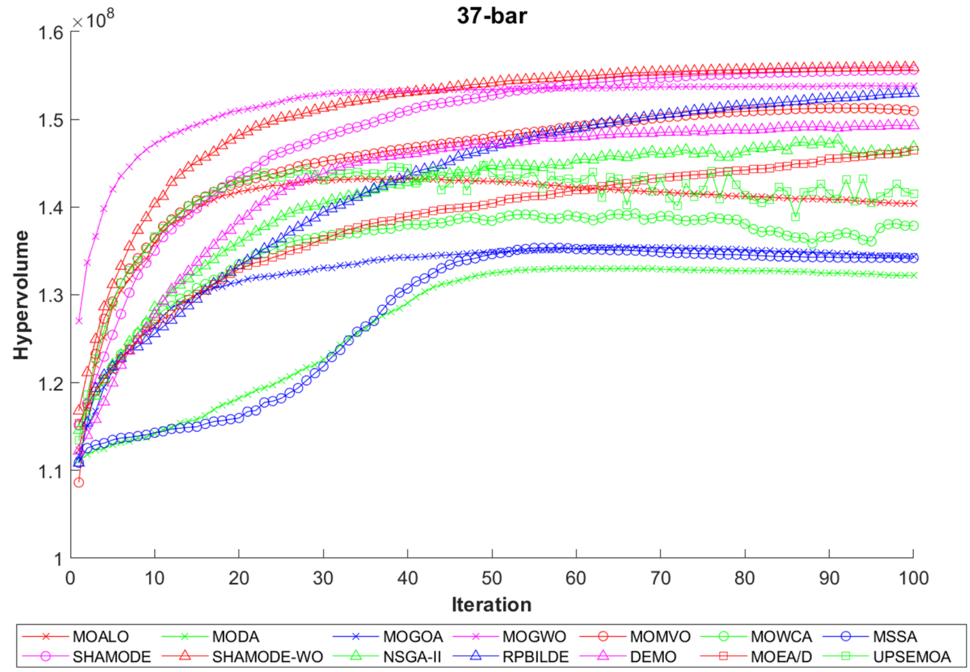
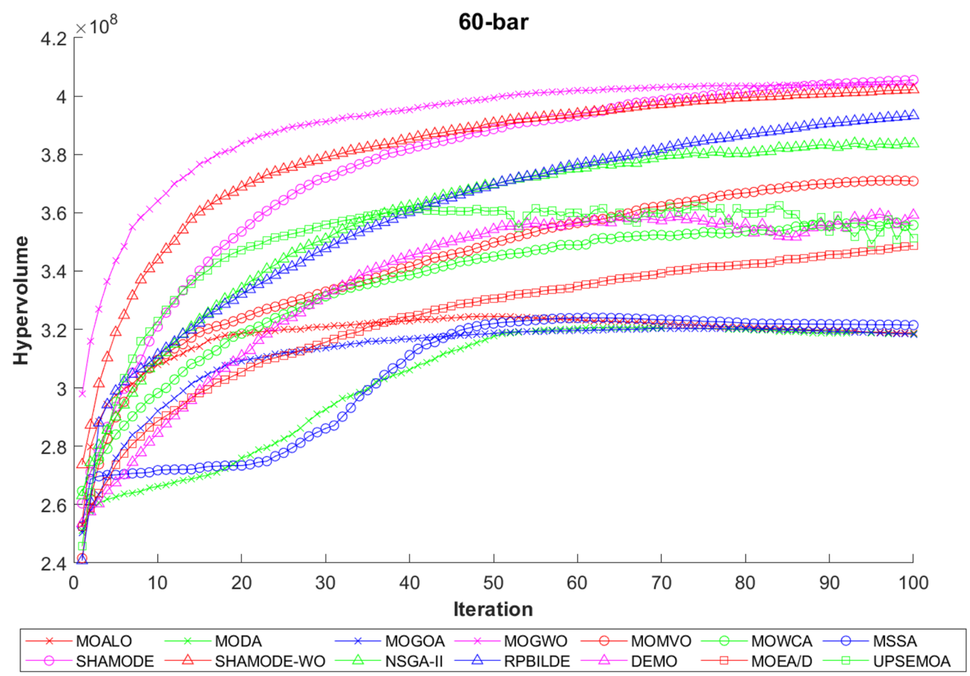


Fig. 7 Mean HV of the 60-bar truss problem



vector and its parent for crossover, SHAMODE-WO replaces the parent with a new solution generated by the whale spiral movement operator.

- Non-dominated sorting genetic algorithm II (NSGA-II) [41]

NSGA-II is probably the most cited multi-objective evolutionary algorithm. The method is based on an elitism strategy. With an initial population, GA operators namely crossover and mutation are applied to create a set of offspring solutions. The next generation is classified by using non-dominated sorting and crowding distance comparison operators. The process is repeated until reaching the termination criterion.

- Hybridisation of real-code population-based incremental learning and differential evolution (RPBILDE) [30]

This is the multi-objective version of real-code population-based incremental learning, which uses a probability matrix to generate a solution population. The solutions

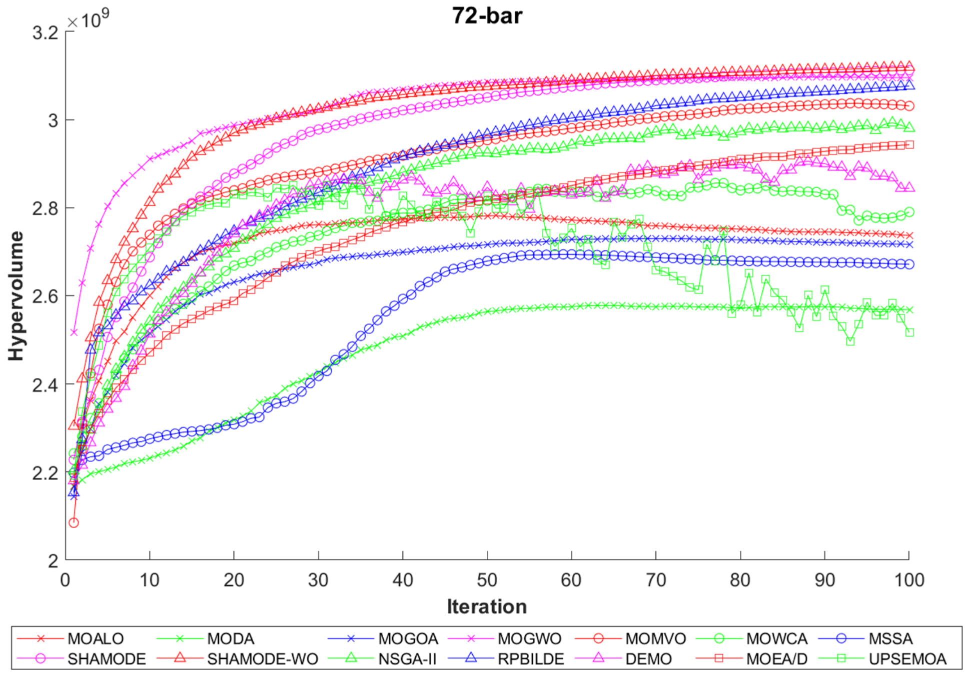


Fig. 8 Mean HV of the 72-bar truss problem

created from the probability matrix are then hybridised with the existing non-dominated solutions by using the DE operators. The probability matrix and the set of non-dominated solutions are iteratively updated until meeting the termination criterion.

- Differential evolution for multi-objective optimisation (DEMO) [42]

  DEMO uses the elitism concept as with NSGAII. The only difference is it exploits DE mutation and binomial crossover for reproduction instead of using the operators of real-code GA.
- Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [43]

  MOEA/D is another popular concept for multi-objective evolutionary algorithms. The technique exploits a scalarisation technique e.g. the weighted sum method to decompose the optimisation problem, and reproduce new solutions. Similarly to most MOMHs, the method is population-based, and uses a non-dominated sorting technique to keep a set of non-dominated solutions. The population and the set of non-dominated solutions are improved iteratively until reaching the stopping condition.
- Unrestricted population size evolutionary multi-objective optimisation algorithm (UPS-EMOA) [44]

UPS-EMOA exploits the idea that a greater size of non-dominated solutions set can improve the search diversity rather than filtering some of them to keep a constant archive size as with most MOMHs. The method uses the DE operators to create offspring while the archive for non-dominated solutions are allowed to have many members as long as the computer memory can handle it.

MATLAB codes of the abovementioned algorithms are mostly provided by their authors while some are coded by the authors of this paper. We use the same programming language and hardware to have a fair comparison between all algorithms.

4 Results and Discussion

In this study, 30 independent runs of each algorithm for solving each test problem are performed while the implemented multi-objective optimisers include MOALO, MODA, MOGOA, MOGWO, MOMVO, MOWCA, MSSA, SHAMODE, SHAMODE-WO, NSGA-II, RPBILDE, DEMO, MOEA/D, and UPSEMOA. Four performance indicators, HV, GD, IGD, and STE, are evaluated to measure the search performance of all competitors. Additional statistical results, Friedman ranking based on the four metrics, are also carried out to confirm search performance of the optimisation algorithms.

The results of mean HV, GD, IGD, and STE are provided in Tables 1, 3, 5, and 7 respectively. Their related Friedman ranks are presented in Tables 2, 4, 6, and 8 respectively. As mentioned in the previous section, the results with higher values HV are the better while the results with lower values of the remaining indicators are the better. In Tables 1, 2, 3, 4, 5, 6, 7 and 8, the best result for each test problem is highlighted with bold font. From Tables 1 and 2, best means and Friedman ranks based on HV are coincided. SHAMODE provided the best mean HV and Friedman rank for the 60-bar test problem while the best means and Friedman ranks of the rest are obtained by SHAMODE-WO.

From Tables 3 and 4, the best GD means and Friedman ranks results are coincided in most test problems. The best results for the 37-bar, 60-bar, and 120-bar problems are obtained from using SHAMODE while the best results for the 72-bar, 200-bar, and 942-bar problems are found by SHAMODE-WO. The best results for the 25-bar problem is found by MOEA/D while the best GD mean and Friedman rank for the 10-bar problem are obtained by MOWCA and SHAMODE respectively.

For the comparative IGD results in Tables 5 and 6, the best algorithms based on both mean value and Friedman rank aspects are similar for most test problems. SHAMODE-WO provided the best mean values and the best Friedman ranks in cases of the 10-bar, 25-bar, 37-bar, 200-bar, and 942-bar problems while the best IGD mean and Friedman rank for 60-bar and 72-bar problems are obtained from MOGWO, SHAMODE. The best mean and Friedman rank for the 120-bar problem are found by different algorithms which are SHAMODE-WO and SHAMODE in that order.

For the STE results in Tables 7 and 8, the best STE means and Friedman ranks are all coincided. SHAMODE-WO gives best mean and Friedman rank of the 942-bar problem while RPBILDE provided best results for the remaining test problems.

From the results, the performance of the optimisation algorithms in multi-objective truss optimisation problems are measured in many different aspects using the HV, GD, IGD, and STE metrics. The HV results in most problems coincide to the IGD results, SHAMODE-WO is the best

Table 1 Mean hypervolume from 30 independent runs of all algorithms (The bold numbers are the best results)

HV mean	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	2.2009E+09	5.4256E+08	1.4040E+08	3.1883E+08	2.7371E+09	7.8035E+10	2.5057E+10	1.6704E+14
MODA	2.1426E+09	5.3627E+08	1.3219E+08	3.1849E+08	2.5684E+09	7.7712E+10	2.0598E+10	1.2779E+14
MOGOA	2.1575E+09	5.3251E+08	1.3438E+08	3.1834E+08	2.7166E+09	7.7959E+10	2.2445E+10	1.4225E+14
MOGWO	2.3665E+09	5.6298E+08	1.5378E+08	4.0365E+08	3.0957E+09	8.3834E+10	2.7019E+10	1.8330E+14
MOMVO	2.3347E+09	5.5925E+08	1.5095E+08	3.7083E+08	3.0311E+09	8.2753E+10	2.5917E+10	1.6751E+14
MOWCA	2.0053E+09	4.5162E+08	1.3787E+08	3.5575E+08	2.7902E+09	5.7039E+10	2.4729E+10	1.6390E+14
MSSA	2.1739E+09	5.4146E+08	1.3419E+08	3.2144E+08	2.6712E+09	7.8291E+10	2.2714E+10	1.4500E+14
SHAMODE	2.3893E+09	5.6909E+08	1.5565E+08	**4.0551E+08**	3.1124E+09	8.4826E+10	2.7161E+10	1.8249E+14
SHAMODE-WO	**2.3934E+09**	**5.6932E+08**	**1.5591E+08**	4.0219E+08	**3.1194E+09**	**8.4834E+10**	**2.7375E+10**	**1.8611E+14**
NSGA-II	2.2343E+09	4.9814E+08	1.4687E+08	3.8366E+08	2.9804E+09	7.2689E+10	2.5807E+10	1.6782E+14
RPBILDE	2.3757E+09	5.6836E+08	1.5298E+08	3.9333E+08	3.0765E+09	8.4367E+10	2.6760E+10	1.7874E+14
DEMO	2.2584E+09	5.1793E+08	1.4928E+08	3.5917E+08	2.8442E+09	8.0265E+10	2.5018E+10	1.7862E+14
MOEA_D	2.2208E+09	5.5974E+08	1.4650E+08	3.4868E+08	2.9428E+09	7.7887E+10	2.5820E+10	1.7247E+14
UPSEMOA	2.1916E+09	5.0990E+08	1.4151E+08	3.5126E+08	2.5166E+09	7.4092E+10	2.5269E+10	1.7751E+14

Table 2 Mean Friedman rank of hypervolume from 30 independent runs of all algorithms (The bold numbers are the best results)

HV rank	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	9.6000	8.5667	10.2000	12.2667	10.6667	9.7000	9.4667	9.2667
MODA	11.7667	9.8000	12.9000	12.2333	13.1667	9.6000	13.8667	14.0000
MOGOA	11.2333	10.4667	12.2333	12.6333	11.1000	9.3667	12.4333	12.7000
MOGWO	3.9000	4.4000	3.3667	2.1000	3.0333	3.9000	2.9000	2.3667
MOMVO	5.2000	5.5000	5.0333	6.4667	5.1000	5.1333	6.9333	9.2000
MOWCA	11.4000	12.8000	10.8333	8.3667	9.1000	13.0000	9.7667	10.0333
MSSA	10.8667	8.8000	12.3000	12.3000	11.5667	9.2333	12.1333	12.3000
SHAMODE	1.9333	2.1333	1.7333	**1.6667**	1.9000	1.6000	2.3000	2.8333
SHAMODE-WO	**1.1333**	**1.4333**	**1.2667**	2.3667	**1.5667**	**1.4333**	**1.0333**	**1.0000**
NSGA-II	8.4333	12.4333	7.5667	5.1333	6.4667	10.9333	7.2000	9.0333
RPBILDE	3.0333	2.6667	3.9667	3.9333	3.7667	3.0667	3.9000	5.0000
DEMO	7.6333	10.3667	6.1333	7.9333	8.6667	7.0333	8.4667	5.1333
MOEA_D	9.0667	4.9333	7.9000	8.9667	7.2667	9.7000	6.9667	7.1000
UPSEMOA	9.8000	10.7000	9.5667	8.6333	11.6333	11.3000	7.6333	5.0333

Table 3 Mean generational distance from 30 independent runs of all algorithms (The bold numbers are the best results)

GD mean	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	1.0750E+02	4.7307E+01	4.3242E+01	9.4015E+01	2.8828E+02	3.3392E+02	9.0548E+02	9.7771E+04
MODA	8.0166E+01	4.3430E+01	4.1741E+01	9.0713E+01	2.6179E+02	1.9008E+02	1.1135E+03	5.1003E+04
MOGOA	9.3015E+01	3.7380E+01	4.2384E+01	1.0474E+02	2.2947E+02	2.4224E+02	1.0567E+03	5.5540E+04
MOGWO	3.3629E+01	2.4774E+01	1.0918E+01	2.6304E+01	5.6767E+01	1.5826E+02	2.3520E+02	2.0849E+04
MOMVO	3.4572E+01	1.6919E+01	1.4515E+01	3.6262E+01	8.4438E+01	1.6983E+02	4.2620E+02	5.4865E+04
MOWCA	**1.6669E+01**	1.7911E+01	1.2827E+01	3.1257E+01	6.8105E+01	1.1078E+02	3.1080E+02	4.3154E+04
MSSA	7.3215E+01	5.1633E+01	3.9725E+01	1.0034E+02	2.2276E+02	3.0058E+02	8.9872E+02	5.0559E+04
SHAMODE	1.6705E+01	8.1491E+00	**4.1474E+00**	**9.4632E+00**	2.4552E+01	**9.5672E+01**	8.8201E+01	1.6579E+04
SHAMODE-WO	1.7350E+01	8.1428E+00	4.2693E+00	1.0179E+01	**2.4464E+01**	9.7443E+01	**6.6234E+01**	**7.8342E+03**
NSGA-II	4.2469E+01	3.4780E+01	1.1000E+01	2.2281E+01	5.3190E+01	2.6677E+02	1.6912E+02	2.4668E+04
RPBILDE	1.8719E+01	8.3392E+00	8.0436E+00	2.5977E+01	3.5301E+01	9.9116E+01	1.7516E+02	2.9317E+04
DEMO	5.1627E+01	3.2649E+01	1.6059E+01	3.8994E+01	1.3084E+02	2.3173E+02	2.8757E+02	3.3437E+04
MOEA_D	2.5784E+01	**7.2692E+00**	7.4276E+00	2.7510E+01	2.8965E+01	1.2487E+02	1.2728E+02	1.8207E+04
UPSEMOA	2.5355E+01	1.1474E+01	1.0613E+01	2.2670E+01	6.1775E+01	1.5187E+02	1.0916E+02	2.5146E+04

Table 4 Mean Friedman rank of generational distance from 30 independent runs of all algorithms (The bold numbers are the best results)

GD rank	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	10.2833	9.4917	9.9417	12.0000	10.8750	10.1833	10.0917	10.2583
MODA	11.8333	10.9417	11.9667	11.3750	12.3167	10.1000	12.9500	12.5000
MOGOA	9.8583	9.2750	10.6250	10.2083	9.7000	9.0333	10.6083	10.3417
MOGWO	5.6083	5.5333	4.9417	4.0750	4.7750	5.5917	4.9500	5.1167
MOMVO	6.0417	6.0417	6.3250	6.4417	6.6417	5.9500	7.7250	9.9583
MOWCA	10.0417	11.2750	10.6917	9.1833	9.8500	10.7833	9.6667	9.5583
MSSA	10.5750	9.9833	11.2750	10.2667	10.3583	9.8833	11.0250	10.8583
SHAMODE	**2.6917**	2.9333	**2.7083**	**2.8917**	3.3833	**2.7667**	3.5250	3.6000
SHAMODE-WO	2.2583	2.2667	1.9667	3.9833	**3.3750**	2.7250	**1.9083**	**1.4000**
NSGA-II	8.0917	10.8250	8.2667	6.4333	6.8000	9.6083	7.5417	7.8250
RPBILDE	3.7083	3.1750	4.1417	4.2083	4.1917	3.4333	4.3500	6.1167
DEMO	7.6750	9.3083	7.0500	8.1083	7.8500	7.9500	7.2833	5.1750
MOEA_D	7.8083	**5.4667**	6.9333	7.8083	6.1917	8.0750	6.9750	7.8750
UPSEMOA	8.5250	8.4833	8.1667	8.0167	8.6917	8.9167	6.4000	4.4167

algorithms based on HV and IGD while RPBILDE provided best STE results. SHAMODE and SHAMODE-WO are consider joint winners in the GD results. SHAMODE-WO is also among one of the runners-up in the STE results, thus, the results of all metrics could still be considered being coincident. There are only the results from the GD metric which somewhat differ from the others. The reason is that GD only measures Euclidian distances from the obtained solutions to the reference front, thus, the spread of the obtained fronts is ignored. STE, on the other hand, measures the front extension and distribution. However, IGD, and HV can measure both the advancement and the spread of the front. IGD used all solutions in the reference front to compute the distances, therefore, a wider spread front will result in overall lower distances and consequently lower IGD, which means it is more reliable than GD.

For overall performance comparison, the mean Friedman ranks of all metric results (Tables 2, 4, 6, 8) are summed up and reported in Table 9. The best result in each test problem is bold and underlined while the runners-up (the 2nd and the 3rd) are presented with the bold font. The overall best algorithm based on the Friedman rank is SHAMODE-WO while SHAMODE and RPBILDE are the runners-up. SHAMODE-WO obtained the best mean ranks in 7 out of 8 problems. SHAMODE provided best mean rank in 1 out of 8 test and is the runner-up in 7 out of 8 test problems. RBILDE is the runner-up in 6 out of 8 problems.

The best Pareto fronts based on the hypervolume indicator obtained from the various optimisers for all multi-objective truss optimisation problems are illustrated in Fig. 3. In cases that the number of design variables are low, the fronts obtained from using the various optimisers are somewhat

Table 5 Mean inverted generational distance from 30 independent runs of all algorithms (The bold numbers are the best results)

IGD mean	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	6.9404E+02	1.0377E+02	9.2570E+01	4.4351E+02	5.0234E+02	6.6225E+03	1.0726E+03	8.5551E+04
MODA	9.8329E+02	1.8724E+02	1.4298E+02	2.8263E+02	6.7376E+02	7.8846E+03	2.0890E+03	3.2706E+05
MOGOA	1.0278E+03	2.5069E+02	1.6138E+02	3.2652E+02	5.1096E+02	6.9073E+03	2.1578E+03	3.0564E+05
MOGWO	1.7364E+02	2.7312E+01	3.4052E+01	**8.0818E+01**	1.4674E+02	1.3504E+03	5.3741E+02	8.7258E+04
MOMVO	1.9550E+02	4.7407E+01	4.8165E+01	1.9504E+02	2.4182E+02	9.8877E+02	1.1882E+03	2.6579E+05
MOWCA	1.8219E+03	6.1500E+02	2.2160E+02	3.9864E+02	6.4706E+02	1.6538E+04	1.7337E+03	2.0246E+05
MSSA	1.0807E+03	1.7830E+02	1.7381E+02	3.2017E+02	5.8565E+02	7.4525E+03	2.0079E+03	2.9332E+05
SHAMODE	1.0126E+02	1.6100E+01	1.8091E+01	1.3400E+02	**1.2876E+02**	6.9801E+02	4.8163E+02	7.2950E+04
SHAMODE-WO	**5.2364E+01**	**1.2185E+01**	**1.3833E+01**	1.6464E+02	1.3298E+02	**6.4704E+02**	**1.4317E+02**	**1.3902E+04**
NSGA-II	4.6497E+02	1.6439E+02	1.0163E+02	2.1872E+02	2.9935E+02	4.2818E+03	1.2977E+03	1.7264E+05
RPBILDE	3.2842E+02	5.0636E+01	7.1395E+01	1.6951E+02	3.1126E+02	2.2942E+03	8.1144E+02	1.7535E+05
DEMO	3.0249E+02	1.3039E+02	4.5812E+01	2.5829E+02	2.8862E+02	1.9849E+03	8.6287E+02	4.0698E+04
MOEA_D	1.2892E+03	2.1637E+02	1.7140E+02	4.4136E+02	6.5605E+02	9.4462E+03	1.8823E+03	3.3507E+05
UPSEMOA	7.0315E+02	1.6106E+02	1.0603E+02	3.1757E+02	5.5010E+02	5.3107E+03	1.1108E+03	4.7123E+04

Table 6 Mean Friedman rank of inverted generational distance from 30 independent runs of all algorithms (The bold numbers are the best results)

IGD rank	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	9.0000	7.4333	7.6333	12.5333	9.1000	9.9333	6.9000	5.0333
MODA	10.8000	10.0000	10.2667	8.1667	11.6333	10.8667	12.3667	12.6333
MOGOA	10.9000	10.9667	10.8667	9.5333	9.7333	10.1000	12.0333	11.9000
MOGWO	3.6667	3.3667	3.5333	**1.6667**	2.8667	3.7000	3.2667	5.1333
MOMVO	4.0667	4.6667	4.6667	5.0000	5.1667	3.1667	7.6000	10.7667
MOWCA	12.9333	13.5667	13.1000	11.5333	10.8000	13.5000	10.2333	8.6000
MSSA	11.2667	9.4667	11.3000	9.4333	10.4667	10.5667	11.9667	11.5000
SHAMODE	2.3333	2.2333	2.0333	3.0333	**2.3667**	**2.0667**	2.7333	4.5333
SHAMODE-WO	**1.1000**	**1.1667**	**1.2667**	4.0667	2.4000	2.1667	**1.1333**	**1.0000**
NSGA-II	6.9333	9.7000	8.5000	6.2000	6.1333	7.3333	7.7333	7.8000
RPBILDE	5.4667	4.4667	6.5333	4.5667	6.6667	5.3333	5.0000	7.8000
DEMO	5.5000	8.0333	4.7333	7.3333	5.9000	5.2000	5.5000	2.5000
MOEA_D	12.0000	10.6000	11.6333	12.3333	11.7667	12.1000	11.5000	12.6667
UPSEMOA	9.0333	9.3333	8.9333	9.6000	10.0000	8.9667	7.0333	3.1333

comparable, however, when the problem has a greater number of design variables, the best fronts obtained from using the best and worst optimisers clearly fall apart.

Figures 4, 5, 6, 7, 8, 9, 10 and 11 show the search history of the implemented optimisers for the eight test problems as the average hypervolume from 30 runs versus the iteration numbers. The figures illustrate that there can be roughly two groups of MHs according to the final HV values. The first group has good consistency and convergence rate including SHAMODE-WO, SHAMODE, MOGWO, and RPBILDE. The second group is somewhat inconsistent and has low values of HV. These include MOALO, MODA, MOGOA, MOWCA, MSSA, NSGA-II, DEMO, MOEA/D, and UPS-EMOA while MOMVO is at the borderline. RPBILDE is more efficient if the design problem has a lower number of design variables while its performance slightly drops when the design problem is large-scale. UPS-EMOA, on the other hand, has low performance for the cases with low number of design variables but its performance increases when solving a large-scale problem. The results of MODA show that it is somewhat inefficient for solving truss multi-objective optimisation.

For reproduction capability, MOGWO arguably has the most efficient reproduction scheme as it has the highest HV in the early stage of an optimisation run. The second and third best reproductions are SHAMODE-WO and SHAMODE respectively, which implies that the integration of whale spiral movement into the DE binomial crossover helps increase its performance. The reproduction of RPBILDE has low performance at the early stage but robustly improves as the search goes on. It is possible that if MOGWO and RPBILDE can exchange some features, their performance

Table 7 Mean spacing-to-extent from 30 independent runs of all algorithms (The bold numbers are the best results)

STE mean	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	1.0934E−02	1.1810E−02	1.0554E−02	1.4908E−02	1.1970E−02	9.8899E−03	1.4286E−02	1.9312E−02
MODA	2.2309E−02	1.8872E−02	2.2303E−02	2.3382E−02	1.7376E−02	1.7507E−02	2.0228E−02	1.6884E−02
MOGOA	6.9239E−03	5.9176E−03	8.1910E−03	6.7033E−03	7.9155E−03	7.4130E−03	6.2342E−03	5.9736E−03
MOGWO	8.3516E−03	6.6940E−03	7.7681E−03	7.5748E−03	7.8240E−03	8.8079E−03	7.3142E−03	8.5219E−03
MOMVO	8.5326E−03	8.3224E−03	8.5202E−03	7.4489E−03	9.2224E−03	8.7959E−03	7.4215E−03	8.5069E−03
MOWCA	2.8303E−02	2.6596E−02	1.9744E−02	1.3859E−02	1.9559E−02	3.2262E−02	1.3473E−02	1.0292E−02
MSSA	1.0074E−02	1.0174E−02	1.0964E−02	8.0179E−03	9.4310E−03	9.1962E−03	9.3203E−03	9.9281E−03
SHAMODE	5.7010E−03	5.1342E−03	6.9634E−03	7.2410E−03	8.4742E−03	5.7440E−03	7.2350E−03	4.7271E−03
SHAMODE−WO	5.4654E−03	4.6108E−03	5.4730E−03	9.1518E−03	8.9589E−03	5.5198E−03	5.4558E−03	**4.2746E−03**
NSGA−II	1.0903E−02	1.7464E−02	1.3428E−02	1.3483E−02	9.9105E−03	1.3439E−02	1.3180E−02	9.3690E−03
RPBILDE	**4.8950E−03**	**4.1377E−03**	**4.1182E−03**	**4.4865E−03**	**4.3165E−03**	**4.4327E−03**	**4.6357E−03**	5.2054E−03
DEMO	9.4242E−03	1.0477E−02	9.2925E−03	1.0128E−02	8.8808E−03	1.0650E−02	1.0740E−02	5.8071E−03
MOEA_D	6.3166E−03	5.4176E−03	6.2777E−03	7.1558E−03	5.0736E−03	7.1561E−03	6.3385E−03	9.3219E−03
UPSEMOA	1.3134E−02	1.5373E−02	9.9493E−03	1.3923E−02	1.3424E−02	1.1533E−02	1.2814E−02	5.4328E−03

Table 8 Mean Friedman rank of spacing-to-extent from 30 independent runs of all algorithms (The bold numbers are the best results)

STE rank	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	9.7000	10.0333	9.5000	10.9667	10.5000	8.5333	12.0000	13.0333
MODA	13.0333	12.4000	12.1333	12.9333	11.7000	11.4000	12.5333	12.4333
MOGOA	5.0667	4.7667	6.7000	5.8333	6.1000	6.0333	5.5000	5.4667
MOGWO	7.1333	5.8667	6.3667	6.2000	6.5333	7.4667	6.4667	8.7333
MOMVO	7.4333	7.4667	7.3000	5.9667	7.8667	7.7000	6.6667	8.5667
MOWCA	12.9667	12.5000	11.4667	9.7667	12.5000	12.6333	10.5000	10.3000
MSSA	8.5667	9.0000	9.2667	6.7000	7.5333	7.4667	7.8333	9.2000
SHAMODE	3.8000	4.2000	5.4333	5.6333	7.0667	4.2333	6.2333	3.6333
SHAMODE-WO	3.6000	3.2667	3.4667	7.6333	7.5333	4.0667	4.1000	**2.6000**
NSGA-II	8.6333	10.9333	10.7667	9.4333	8.5333	10.0333	9.7667	8.9667
RPBILDE	**2.6333**	**2.3000**	**1.7000**	**2.2000**	**2.2667**	**1.8000**	**2.5333**	4.8000
DEMO	7.9000	8.7333	8.4000	8.5000	7.1667	9.1667	8.1000	5.4667
MOEA_D	4.7667	4.0000	4.2667	4.4667	2.9667	5.6000	5.1667	8.0333
UPSEMOA	9.7667	9.5333	8.2333	8.7667	6.7333	8.8667	7.6000	3.7667

Table 9 Overall mean Friedman rank of all metrics for all test problems (The underlined and bold numbers are the best results while the bold numbers are the second and third best)

Mean rank	10-bar	25-bar	37-bar	60-bar	72-bar	120-bar	200-bar	942-bar
MOALO	10.2833	9.4917	9.9417	12.0000	10.8750	10.1833	10.0917	10.2583
MODA	11.8333	10.9417	11.9667	11.3750	12.3167	10.1000	12.9500	12.5000
MOGOA	9.8583	9.2750	10.6250	10.2083	9.7000	9.0333	10.6083	10.3417
MOGWO	5.6083	5.5333	4.9417	**4.0750**	4.7750	5.5917	4.9500	5.1167
MOMVO	6.0417	6.0417	6.3250	6.4417	6.6417	5.9500	7.7250	9.9583
MOWCA	10.0417	11.2750	10.6917	9.1833	9.8500	10.7833	9.6667	9.5583
MSSA	10.5750	9.9833	11.2750	10.2667	10.3583	9.8833	11.0250	10.8583
SHAMODE	**2.6917**	**2.9333**	**2.7083**	**2.8917**	**3.3833**	**2.7667**	**3.5250**	**3.6000**
SHAMODE-WO	**2.2583**	**2.2667**	**1.9667**	**3.9833**	**3.3750**	**2.7250**	**1.9083**	**1.4000**
NSGA-II	8.0917	10.8250	8.2667	6.4333	6.8000	9.6083	7.5417	7.8250
RPBILDE	**3.7083**	**3.1750**	**4.1417**	4.2083	**4.1917**	**3.4333**	**4.3500**	6.1167
DEMO	7.6750	9.3083	7.0500	8.1083	7.8500	7.9500	7.2833	5.1750
MOEA_D	7.8083	5.4667	6.9333	7.8083	6.1917	8.0750	6.9750	7.8750
UPSEMOA	8.5250	8.4833	8.1667	8.0167	8.6917	8.9167	6.4000	**4.4167**

Fig. 9 Mean HV of the 120-bar truss problem

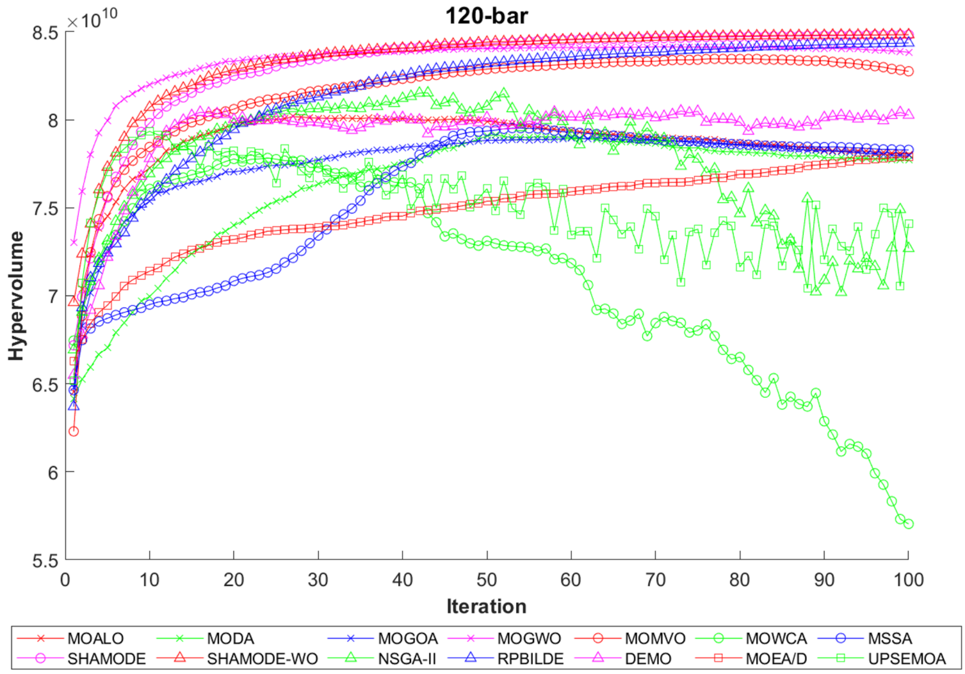
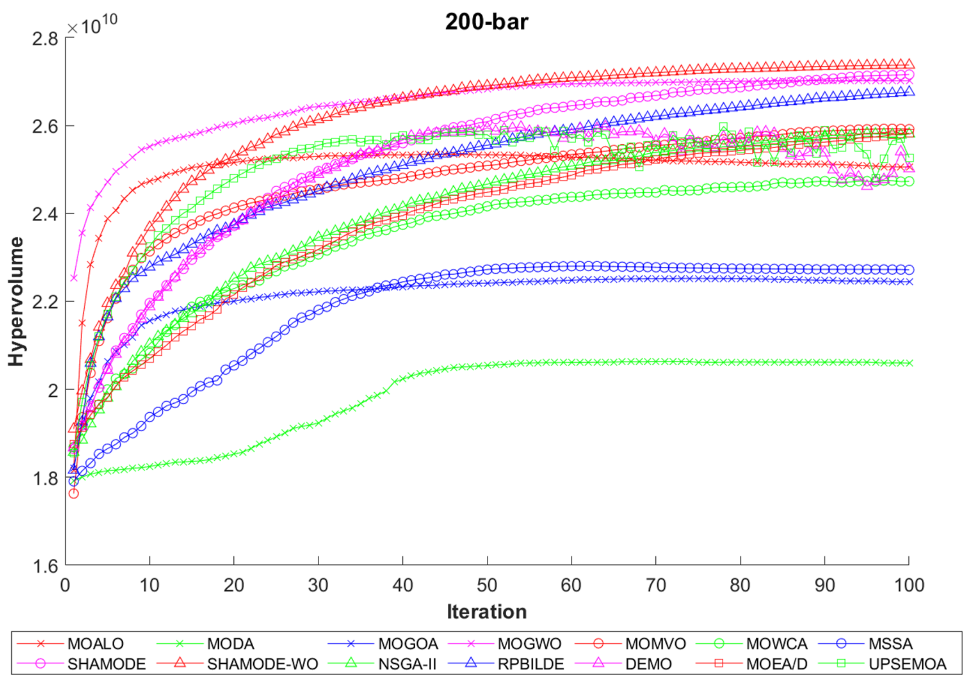


Fig. 10 Mean HV of the 200-bar truss problem



can probably be improved. For the large-scale cases of 200-bar and 942-bar trusses, MOALO has the second best reproduction at the early stage after MOGWO, however, as the optimisation process continues, the HV drops. This implies that its selection operator is not sufficiently effective.

The selection process of MOMHs is vital for both selection of the next generation population and the Pareto archive. The Pareto archive is usually assigned to keep non-dominated solutions, which at the final iteration it will be regarded as the obtained approximate Pareto front. Often, during the optimisation run, the Pareto archive will contain an excessive number of non-dominated solutions possibly reaching the computer memory limit. This makes researchers invent the so-called Pareto archiving technique to screen some non-dominated solutions out of the archive. The concept is to maintain the high diversity of the solutions that are not removed as much as possible. With such a concept, it can be seen that some implemented MOMHs have the

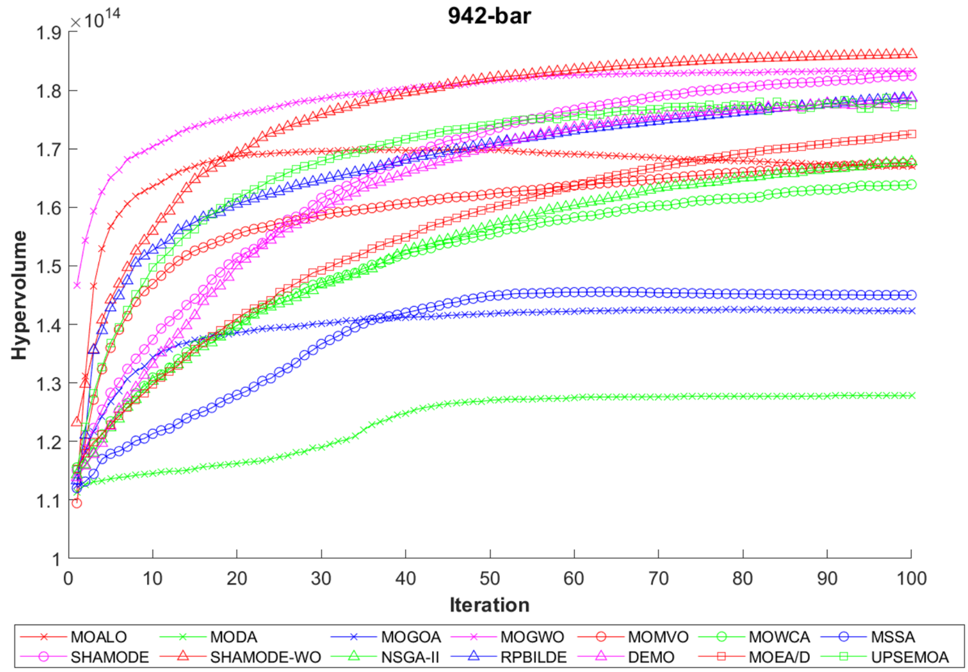


Fig. 11 Mean HV of the 942-bar truss problem

problem of archiving non-dominated solutions leading to the drop of the average HV in later iterations. This includes MOWCA, DEMO, NSGAII, MSSA, MOALO, and UPS-EMOA. It should be noted that, although the original version of UPS-EMOA offers not to limit the archive size, but it is sometimes impractical. Thus, it has to be limited with a very large archive size. The problem of Pareto archiving is very important according to this study as some underperformers may be improved by using another archive scheme.

5 Conclusions

The comparative performance study of a number of new and established multi-objective metaheuristics for truss optimisation has been carried out. Eight classical truss structures usually presented in literature are used to set up the multi-objective test problems. The design problem is posed to minimise structural mass and compliance subject to bound and stress constraints while fourteen MOMHs are implemented to solve the problems. The comparative results based on the four performance indicators i.e. HV, GD, IGD, and STE reveal that SHAMODE-WO is considered the overall best algorithm while SHAMODE and RPBILDE are the second and third best optimisers respectively. It is very challenging to improved the search speed and consistency of a metaheuristic to solve variety of problems. There is still a big room for developing a novel metaheuristic with better performance in future work. In truss multi-objective optimisation, MH operators for both search diversification and intensification need to be invented. The former is used to get a better front spread and deal with a problem with nonconvex feasible region while the latter is used to improve front advancement. The archiving technique is also an important element of MOMHs as discussed in the result section.

Funding Sujin, Natee, and Nantiwat have financial support from the Thailand Research Fund (RTA6180010).

Availability of data and materials The test data, test problems' details, and optimisation parameters settings can be found in https://github.com/Natee-Panagant/Comparative-study-of-truss-sizing-optimisation-problems-in-MATLAB/.

Code availability All the implemented MATLAB codes are available in https://github.com/Natee-Panagant/Comparative-study-of-truss-sizing-optimisation-problems-in-MATLAB/.

Compliance with Ethical Standards

Conflict of interest The authors declare that there are no conflicts of interest related to this paper.

References

1. Pholdee N, Bureerat S, Jaroenapibal P, Radpukdee T (2017) Many-objective optimisation of trusses through meta-heuristics. In: Advances in neural networks—ISNN 2017, Cham, pp 143–152. https://doi.org/10.1007/978-3-319-59072-1_18
2. Yildiz AR, Abderazek H, Mirjalili S (2019) A comparative study of recent non-traditional methods for mechanical design

optimization. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-019-09343-x

3. Xia L, Xia Q, Huang X, Xie YM (2018) Bi-directional evolutionary structural optimization on advanced structures and materials: a comprehensive review. Arch Comput Methods Eng 25(2):437–478. https://doi.org/10.1007/s11831-016-9203-2
4. Patel VK, Raja BD (2020) Comparative performance of recent advanced optimization algorithms for minimum energy requirement solutions in water pump switching network. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-020-09429-x
5. Rao RV, Saroj A, Ocłoń P, Taler J (2020) Design optimization of heat exchangers with advanced optimization techniques: a review. Arch Comput Methods Eng 27(2):517–548. https://doi.org/10.1007/s11831-019-09318-y
6. Greiner D, Periaux J, Emperador JM, Galván B, Winter G (2017) Game theory based evolutionary algorithms: a review with Nash applications in structural engineering optimization problems. Arch Comput Methods Eng 24(4):703–750. https://doi.org/10.1007/s11831-016-9187-y
7. Tang Z, Hu X, Périaux J (2019) Multi-level hybridized optimization methods coupling local search deterministic and global search evolutionary algorithms. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-019-09336-w
8. Abualigah L, Shehab M, Alshinwan M, Mirjalili S, Elaziz MA (2020) Ant lion optimizer: a comprehensive survey of its variants and applications. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-020-09420-6
9. Noilublao N, Bureerat S (2011) Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms. Comput Struct 89(23):2531–2538. https://doi.org/10.1016/j.compstruc.2011.08.010
10. Ohsaki M (1995) Genetic algorithm for topology optimization of trusses. Comput Struct 57(2):219–225. https://doi.org/10.1016/0045-7949(94)00617-C
11. Hajela P, Lee E (1995) Genetic algorithms in truss topological optimization. Int J Solids Struct 32(22):3341–3357. https://doi.org/10.1016/0020-7683(94)00306-H
12. Chen G-S, Bruno RJ, Salama M (1991) Optimal placement of active/passive members in truss structures using simulated annealing. AIAA J 29(8):1327–1334. https://doi.org/10.2514/3.10739
13. Schutte JF, Groenwold AA (2003) Sizing design of truss structures using particle swarms. Struct Multidisc Optim 25(4):261–269. https://doi.org/10.1007/s00158-003-0316-5
14. Wu C-Y, Tseng K-Y (2010) Truss structure optimization using adaptive multi-population differential evolution. Struct Multidisc Optim 42(4):575–590. https://doi.org/10.1007/s00158-010-0507-9
15. Tejani GG, Savsani VJ, Patel VK, Bureerat S (2017) Topology, shape, and size optimization of truss structures using modified teaching-learning based optimization. Adv Comput Des 2(4):313–331
16. Sonmez M (2011) Artificial Bee Colony algorithm for optimization of truss structures. Appl Soft Comput 11(2):2406–2418. https://doi.org/10.1016/j.asoc.2010.09.003
17. Kaveh A, Ahmadi B (2014) Sizing, geometry and topology optimization of trusses using force method and supervised charged system search. Struct Eng Mech 50(3): 365–382
18. Lieu QX, Do DTT, Lee J (2018) An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. Comput Struct 195:99–112. https://doi.org/10.1016/j.compstruc.2017.06.016
19. Kaveh A, Dadras A, Montazeran AH (2018) Chaotic enhanced colliding bodies algorithms for size optimization of truss structures. Acta Mech 229(7):2883–2907. https://doi.org/10.1007/s00707-018-2149-8
20. Yancang L, Zhen Y (2019) Application of improved bat algorithm in truss optimization. KSCE J Civ Eng 23(6):2636–2643. https://doi.org/10.1007/s12205-019-2119-2
21. Gandomi AH, Talatahari S, Tadbiri F, Alavi AH (2013) Krill herd algorithm for optimum design of truss structures. Int J Bio-Inspired Comput 5(5):281–288. https://doi.org/10.1504/IJBIC.2013.057191
22. Kaveh A, Khayatazad M (2013) Ray optimization for size and shape optimization of truss structures. Comput Struct 117:82–94. https://doi.org/10.1016/j.compstruc.2012.12.010
23. Tejani GG, Pholdee N, Bureerat S, Prayogo D, Gandomi AH (2019) Structural optimization using multi-objective modified adaptive symbiotic organisms search. Expert Syst Appl 125:425–441. https://doi.org/10.1016/j.eswa.2019.01.068
24. Tejani GG, Savsani VJ, Bureerat S, Patel VK, Savsani P (2019) Topology optimization of truss subjected to static and dynamic constraints by integrating simulated annealing into passing vehicle search algorithms. Eng Comput 35(2):499–517. https://doi.org/10.1007/s00366-018-0612-8
25. Pholdee N, Bureerat S (2018) A comparative study of eighteen self-adaptive metaheuristic algorithms for truss sizing optimisation. KSCE J Civ Eng 22(8):2982–2993. https://doi.org/10.1007/s12205-017-0095-y
26. Greiner D, Hajela P (2012) Truss topology optimization for mass and reliability considerations—co-evolutionary multiobjective formulations. Struct Multidisc Optim 45(4):589–613. https://doi.org/10.1007/s00158-011-0709-9
27. Techasen T, Wansasueb K, Panagant N, Pholdee N, Bureerat S (2019) Simultaneous topology, shape, and size optimization of trusses, taking account of uncertainties using multi-objective evolutionary algorithms. Eng Compu 35(2):721–740. https://doi.org/10.1007/s00366-018-0629-z
28. Panagant N, Bureerat S, Tai K (2019) A novel self-adaptive hybrid multi-objective meta-heuristic for reliability design of trusses with simultaneous topology, shape and sizing optimisation design variables. Struct Multidisc Optim 60(5):1937–1955. https://doi.org/10.1007/s00158-019-02302-x
29. Pholdee N, Bureerat S (2012) Performance enhancement of multiobjective evolutionary optimisers for truss design using an approximate gradient. Comput Struct 106–107:115–124. https://doi.org/10.1016/j.compstruc.2012.04.015
30. Pholdee N, Bureerat S (2013) Hybridisation of real-code population-based incremental learning and differential evolution for multiobjective design of trusses. Inf Sci 223:136–152. https://doi.org/10.1016/j.ins.2012.10.008
31. Pholdee N, Bureerat S (2014) Hybrid real-code population-based incremental learning and approximate gradients for multi-objective truss design. Eng Optim 46(8):1032–1051. https://doi.org/10.1080/0305215X.2013.823194
32. Tejani GG, Pholdee N, Bureerat S, Prayogo D (2018) Multiobjective adaptive symbiotic organisms search for truss optimization problems. Knowl-Based Syst 161:398–414. https://doi.org/10.1016/j.knosys.2018.08.005
33. Kumar S, Tejani GG, Pholdee N, Bureerat S (2020) Multi-objective modified heat transfer search for truss optimization. Eng Comput. https://doi.org/10.1007/s00366-020-01010-1
34. Mirjalili S, Jangir P, Saremi S (2017) Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Appl Intell 46(1):79–95. https://doi.org/10.1007/s10489-016-0825-8
35. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073. https://doi.org/10.1007/s00521-015-1920-1
36. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization

problems. Appl Intell 48(4):805–820. https://doi.org/10.1007/s10489-017-1019-8
37. Mirjalili S, Saremi S, Mirjalili SM, dos Coelho LS (2016) Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl 47:106–119. https://doi.org/10.1016/j.eswa.2015.10.039
38. Mirjalili S, Jangir P, Mirjalili SZ, Saremi S, Trivedi IN (2017) Optimization of problems with multiple objectives using the multi-verse optimization algorithm. Knowl-Based Syst 134:50–71. https://doi.org/10.1016/j.knosys.2017.07.018
39. Sadollah A, Eskandar H, Kim JH (2015) Water cycle algorithm for solving constrained multi-objective optimization problems. Appl Soft Comput 27:279–298. https://doi.org/10.1016/j.asoc.2014.10.042
40. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002
41. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017
42. Robič T, Filipič B (2005) DEMO: differential evolution for multiobjective optimization. In: Evolutionary multi-criterion optimization, pp 520–533
43. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
44. Aittokoski T, Miettinen K (2010) Efficient evolutionary approach to approximate the Pareto-optimal set in multiobjective optimization, UPS-EMOA. Optim Methods Softw 25(6):841–858
45. Veldhuizen DAV, Lamont GB (2000) Multiobjective evolutionary algorithms: analyzing the state-of-the-art. Evol Comput 8(2):125–147. https://doi.org/10.1162/106365600568158
46. Sierra MR, Coello Coello CA (2005) Improving PSO-based multi-objective optimization using crowding, mutation and ∈-dominance. In: Evolutionary multi-criterion optimization, Berlin, pp 505–519. https://doi.org/10.1007/978-3-540-31880-4_35