INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2015; 5(1):79-101

PERFORMANCE-BASED MULTI-OBJECTIVE OPTIMUM DESIGN FOR STEEL STRUCTURES WITH INTELLIGENCE ALGORITHMS

J. C. Liang, L. J. Li[*,†] and J. N. He
School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, 510006, China

ABSTRACT

A multi-objective heuristic particle swarm optimiser (MOHPSO) based on Pareto multi-objective theory is proposed to solve multi-objective optimality problems. The optimality objectives are the roof displacement and structure weight. Two types of structure are analysed in this paper, a truss structure and a framework structure. Performance-based seismic analysis, such as classical and modal pushover analysis, is carried out for the structures. Four optimality algorithms, namely, NSGA-II, MOPSO, MGSO, and MOHPSO, were used for structural optimisation to compare the effectiveness of the algorithms. The calculation results indicate that MOHPSO outperformed the other algorithms in terms of solution stability, universality, and consistency of the distribution of the Pareto front and the ability to consider constraints. The population can converge to the true Pareto front in the latter generations, which indicates that MOHPSO is effective for engineering multi-objective optimality problems.

Received: 20 November 2014; Accepted: 13 January 2015

KEY WORDS: performance-based seismic design; structural optimisation; heuristic particle swarm optimiser; multi-objective optimisation.

1. INTRODUCTION

In recent years, swarm intelligence optimality algorithms, which have advantages over traditional algorithms with respect to the consideration of practical engineering problems, have been widely adopted and applied to structural optimisation problems [1]. Previous studies have largely focused on the optimisation of a single goal [2, 3]. However, practical

[*]Corresponding author: School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, 510006, China
[†]E-mail address: lilj@gdut.edu.cn (L. J. Li)

engineering problems are complex, and in many cases, more than one objective must be considered to satisfy the design requirements. Although swarm intelligence algorithms have several limitations, they are capable of solving complex problems, including multi-objective optimality design tasks. For example, Gholizadeh [4] reported the seismic optimality design of framework by a group search optimiser (GSO) and pushover analysis. Kaveh et al. [5] combined the nondominated sorting genetic algorithm II (NSGA-II) with an RBF network to solve multi-objective problems considering initial costs and life-cycle cost. Li et al. [6] analysed the life-cycle cost of structures based on a modified modal pushover using a simulated annealing algorithm. Oskouei et al. [7] used a genetic algorithm to determine the minimum weight of semi-rigid-connection steel frames. CMOPGA was proposed to solve multi-objective problems considering minimum weight and dynamic strain energy for steel frames [8]. Liu et al. [9] presented an improved multi-objective GSO for multi-objective optimality problems. Swarm intelligence-based bionic simulated evolutionary algorithms have attracted increasing attention from researchers in the field of optimisation and have been widely used in multi-objective optimisation problems, particularly for structural seismic optimal design.

Linear analysis is the technique suggested for use in the Chinese Code [10] for the seismic design of building structures. However, the results obtained via linear analysis do not accurately estimate the inelastic response of structures. At present, ATC-40 [11], FEMA-273 [12], FEMA-356 [13], and FEMA-440 [14] are the most practical guidelines for nonlinear static analysis methods in the United States. The use of these guidelines is simpler and more efficient than nonlinear response history analysis. Chopra et al. [15] proposed a modal pushover analysis (MPA) procedure that, in addition to considering the first modal form, accounts for the contributions of higher modal forms in nonlinear static analysis. Chopra et al. [16] proposed a modified MPA to improve the accuracy of the MPA results in certain situations.

The particle swarm optimiser (PSO), proposed by Kennedy and Eberhart [17], was first used to solve single-objective problems. However, many researchers have developed multi-objective particle swarm optimisers (MOPSOs) based on PSO. One of the traditional MOPSOs was proposed by Coello et al. [18]. MOPSO, based on the Pareto strategy, chooses the globally optimal particle by an adaptive grid technique. Another successful combination of the Pareto strategy and an intelligence algorithm is NSGA-II [19], which is based on the evolutionary strategy of genetic algorithms. The validity and practicability of nondominated sorting have been illustrated with numerous multidisciplinary examples. The multi-objective group search optimiser (MGSO) [20] chooses an individual with infinite crowding distance in the Pareto front as a producer. The producer-scrounger model was used to update the group information. A multi-objective heuristic particle optimiser (MOHPSO) based on the heuristic particle swarm optimiser (HPSO) [3] is presented in this paper and is combined with multi-objective techniques to solve multi-constraint and multi-objective problems. The performance-based seismic design of a steel frame is presented as a case study.

2. MULTI-OBJECTIVE OPTIMISATION CONCEPTS

2.1 Multi-objective optimisation problems

A multi-objective optimisation problem for minimum objective design is shown below.

$$\textit{minimise } y = f(x) = [f_1(x), f_2(x), \ldots, f_m(x)] \tag{1}$$

$$\begin{aligned} & g_k(x) \geq 0, k = 1, 2, ..., p; \\ \textit{subject to} \quad & h_l(x) = 0, l = 1, 2, ..., q; \\ & x^{\min} \leq x \leq x^{\max}. \end{aligned} \tag{2}$$

Where $x = [x_1, x_2, ..., x_n]$ is an n-dimensional decision variable. Each dimensional decision variable x_i varies from $x_i^{\min}$ to $x_i^{\max}$. $f_j(x)(j = 1, 2, ..., m)$ is the j^{th} objective function satisfying constraints of p inequalities and q equalities.

Definition 1: Pareto dominance. For two-decision vector variables $u = (u_1, u_2, ..., u_n)$ and $v = (v_1, v_2, ..., v_n)$, the dominance of u over v is denoted as $u \succ v$,

$$\begin{aligned} & \forall i \in \{1, ..., m\}, f_i(u) \leq f_i(v) \\ \textit{iff } & \wedge \exists j \in \{1, ..., m\}, f_j(u) \prec f_j(v) \end{aligned} \tag{3}$$

Definition 2: Pareto constraint dominance. For multi-objective optimal problems with constraints, the solution *i* constraint-dominates solution *j*, if: (1) *i* is feasible and *j* is infeasible; or (2) both *i* and *j* are infeasible but the amount of constraint violation is lower for *i* than for *j*; or (3) both *i* and *j* are feasible but the fitness value of *i* is better than *j*.

Definition 3: Pareto optimality. Regarding a set $A \subseteq X_f$, a decision vector $x \in X_f$ is nondominated if $\nexists a \in A : a \succ x$. Moreover, x is considered Pareto optimal if x is non-dominated regarding X_f.

Definition 4: Nondominated set and Pareto-optimal front. Let $A \subseteq X_f$. The function $p(A)$ provides the set of nondominated decision vectors in A: $p(A) = \{a \in A \mid a\}$; then, the set $p(A)$ is the nondominated set regarding A, and the corresponding set of objective vectors $f(p(A))$ is the nondominated front set regarding A. The set $X_p = p(X_f)$ is called the Pareto-optimal set, and the set $Y_p = f(X_p)$ is denoted as the Pareto-optimal front.

Definition 5: The measure of performance for multi-objective optimisation results. A multi-objective optimisation problem has more than one objective, in contrast to a single-objective optimisation problem. The measure of performance for multi-objective optimisation results includes three aspects. First, the distance between the nondominated front and the Pareto-optimal front is minimised. Second, the distribution of the Pareto front obtained is uniform. Third, the distribution front of the nondominated solutions is maximised.

2.2 Non-dominated sorting genetic algorithm II (NSGA-II)

NSGA-II [19] is one of the most widely used multi-objective optimisers that can converge to the true Pareto-optimal front by using a fast nondominated sorting approach based on Pareto dominance. NSGA-II is prior to NSGA and exhibits superior performance.

NSGA-II provides a new generation by combining parent and offspring populations together. In this situation, double individuals are obtained from the combined population, and the calculation quantity decreases from $O(MN^3)$ to $O(MN^2)$, where M is the number of objectives and N is the population size. Elite offspring populations are created by the parent population through selection, crossover, and mutation operations, and the sharing parameter σ_{share} is replaced by crowded-comparison operators.

The NSGA-II procedure is shown in Fig. 1. First, an initial parent population P_0 with a size N is randomly generated in the searching space. The parent population is then sorted to several ranks in a nondominated manner. Then, the individuals in each rank are sorted again based on crowding distance. Excellent individuals are placed in the front of the population by the bi-level sorting scheme. The first offspring population Q_0 with a size N is created by the elites in the parent population using crossover and mutation operators. Finally, a new population R_0 with a size $2N$, which is the combination of the current parent population and the offspring population, is created. The best N individuals in the new population are chosen as the parent population for the next generation.

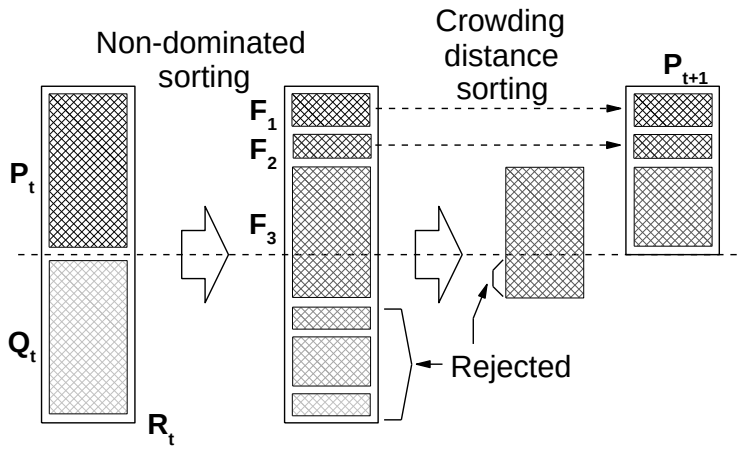


Figure 1. NSGA-II procedure

2.3 Multi-objective particle swarm optimiser (MOPSO)

Coello et al. [18] proposed a MOPSO due to the success of applying Pareto dominance to the evolutionary multi-objective optimiser. An external repository *REP* was used to store nondominated solutions in MOPSO. The update of *REP* is based on an adaptive grid method, and the global best position, which leads the evolutionary direction of the population, is also selected based on the adaptive grid method. The adaptive grid employs the following techniques:

(a) Maintenance and update rules. The new solution is accepted if *REP* is empty; the new solution is automatically rejected if it is dominated by the individual in *REP*; the new solution is accepted and the individuals are rejected if the new solution dominates individuals in *REP*; the new solution is accepted if it does not dominate any individual in *REP* and vice versa. In the case of accepting a new solution, if *REP* has reached its allowable capacity, then one hypercube is randomly selected from those hypercubes containing the maximum number of particles, and one particle in the selected hypercube is rejected randomly. If the accepted new solution lies outside the current bounds of the grid, then the grid must be recalculated, and each individual within it must be relocated. These maintenance and update rules are shown in Figs. 2 and 3.

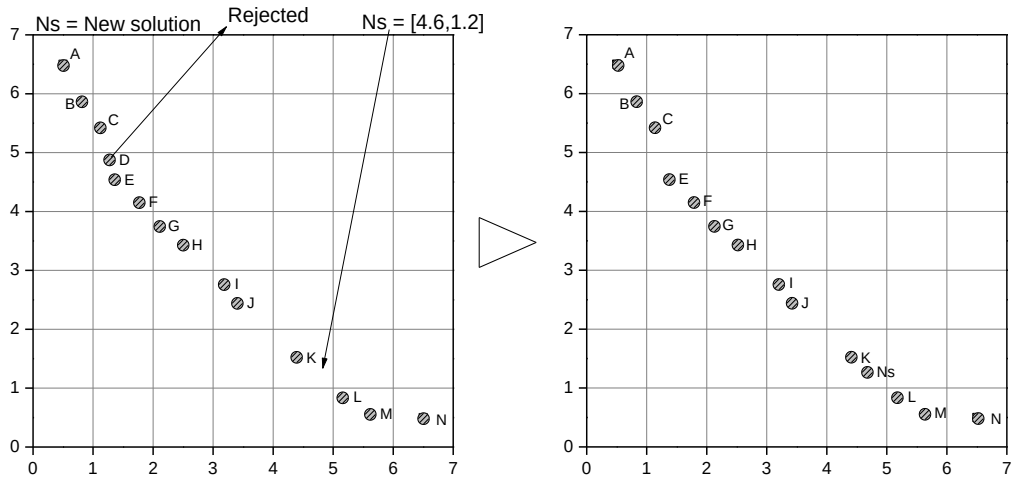


Figure 2. The insertion of a new element in the adaptive grid with the individual within the boundaries of the grid (Ns = [4.6, 1.2])

(b) Selection of the global best position, *GBEST*. A hypercube containing particles is chosen by the roulette-wheel selection method. However, a hypercube with more particles assumes a smaller portion of the wheel. One particle in a hypercube will be chosen randomly as *GBEST* when it is selected. The MOPSO procedure is as follows:

Step 1. Randomly generate the initial population with a size *N* in the search space.

Step 2. Initialise the velocity of each particle.

Step 3. Evaluate the fitness of all particles and record the best personal position *PBEST* based on Pareto dominance.

Step 4. Store nondominated solutions to the *REP* based on the adaptive grid.

Step 5. Choose a particle in *REP* as *GBEST* based on the adaptive grid, and then, compute the speed V of each particle and update the position X using Equations (4) and (5), respectively.

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad 1 \le i \le n \tag{4}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad 1 \le i \le n \tag{5}$$

Where P_i^k and P_g^k are $PBEST$ and $GBEST$ in the k^{th} iteration, respectively. c_1 and c_2 are the learning and accelerating coefficients, respectively. r_1 and r_2 are uniform random sequences in the range (0, 1). ω is the inertia weight, and n is the size of the population.

Step 6. Stop the procedure if the stopping criteria are fulfilled. Otherwise, continue to step 3.

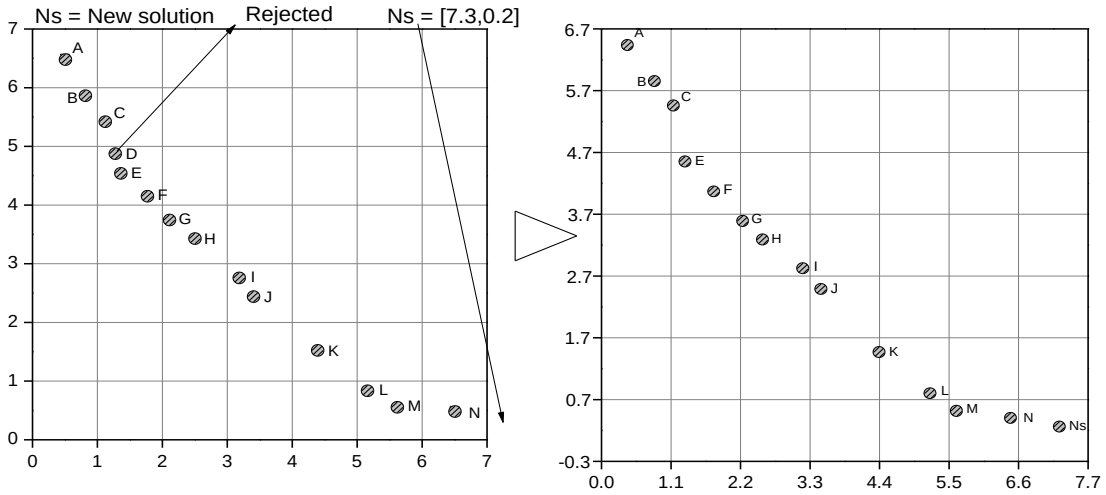


Figure 3. The insertion of a new element in the adaptive grid with the individual without the boundaries of the grid (Ns = [7.3, 0.2])

2.4 Multi-objective group search optimiser (MGSO)

The multi-objective group search optimiser (MGSO) was proposed by Li and Liu [20]. MGSO updates the population based on the PS model. A nondominated set $NDset$ is constructed to store the nondominated solutions. The elites in the $NDset$ constitute an external elite set named EES. An individual with the largest crowding distance in EES is chosen as a producer, which has a function similar to $GBEST$ in the PSO. The procedure of the MGSO is as follows:

Step 1. Randomly initialise the position and head angles of all members in the search space.

Step 2. Calculate the fitness values for all members. Select the elites in the $NDset$ as EES and maintain and update the $NDset$ based on Pareto dominance.

Step 3. A member in EES with the largest crowding distance is chosen as producer, and then, 80% of the members are randomly selected to perform as scroungers, and the remaining members act as rangers. The positions of all new members are checked to determine whether they were within the search space; otherwise, the members are pulled back to the border of the search space.

Step 4. The procedure is stopped if the stopping criteria are reached; otherwise, the procedure is repeated from step 2.

The iterative procedure is shown in Fig. 4, where t is the t^{th} iteration, P, S, and R are the

producer, scrounger, and ranger, respectively, and *Pop* is the population.

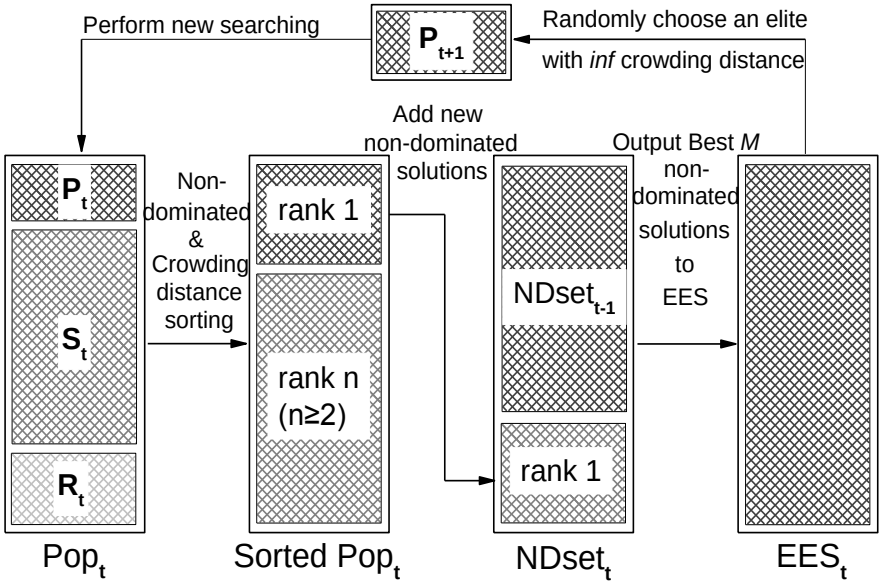


Figure 4. The MGSO procedure

2.5 Heuristic particle swarm optimiser (HPSO)

The heuristic particle swarm optimiser (HPSO) [3] is an improved PSO [17] that combines the harmony search algorithm (HS) with passive congregation [21]. The HS technique is used to handle variables that violate the variable boundaries. The particles that violated boundaries are replaced by the particle $PBEST$, which is similar to the harmony memory mechanism. A flyback mechanism used to handle constraints is used to determine whether any of the particles in the new population are infeasible. The particle is forced to fly back to the previous position if the constraints are not satisfied. The position X of the i^{th} particle is updated according to Equations (6) and (7):

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (R_i^k - X_i^k) \quad 1 \le i \le n \tag{6}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad 1 \le i \le n \tag{7}$$

where R_i is a particle selected randomly from the swarm, c_3 is the passive congregation coefficient, and r_3 is a uniform random sequence in the range (0, 1). The other parameter symbols are the same as in PSO. The diagram of Equation (6) is shown in Fig. 5. The HPSO procedure is as follows:

Step 1. Initialise the positions and velocities of the swarm randomly; then, calculate the fitness of all variables.

Step 2. Determine whether each particle is infeasible or not. If the particle is infeasible, then it is regenerated randomly until it is feasible. Update $PBEST$ and $GBEST$.

Step 3. Update the velocities and positions of the swarms using Equations (6) and (7).

Step 4. Determine whether each particle violates its corresponding boundary. If the particle violates the boundary, then it is replaced by the corresponding component of the vector from $PBEST$ selected randomly.

Step 5. Calculate the fitness values of the new position X_i^{k+1}; then, determine whether the current particle violates the specified constraints. Reset the particle to the previous position X_i^k (flyback mechanism) if it violates the constraints.

Step 6. Update $PBEST$ and $GBEST$.

Step 7. Stop the procedure if the stopping criteria are reached; otherwise repeat the procedure from step 3.

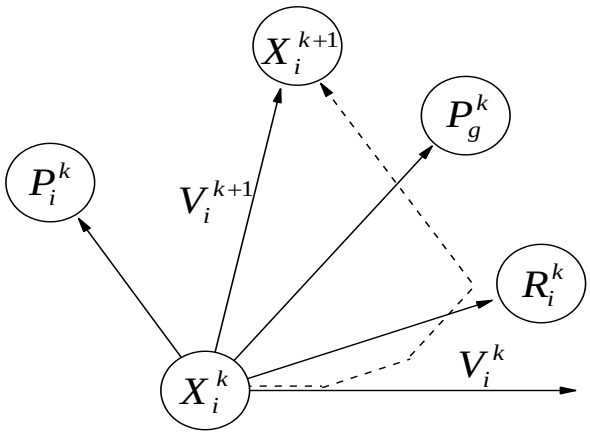


Figure 5. Individual position updating in PSOPC

2.6 Multi-objective heuristic particle optimiser (MOHPSO)

This paper presents a MOHPSO based on classical multi-objective algorithms. Its evolutionary process is based on HPSO. However, $PBEST$ is determined by crowding distance theory in NSGA-II. The REP, also known as EES, and $GBEST$ are maintained and updated by the adaptive grid technique. Due to the difference of mechanism in violating constraints between MOHPSO and HPSO, the flyback mechanism is still useful. However, the boundary values are accepted directly for the condition of violating variable boundaries. The MOHPSO procedure is summarised as follows:

Step 1. Initialise the positions and velocities of the swarms randomly, and then, calculate the fitness of all particles.

Step 2. Determine whether each particle is feasible. If the particle is infeasible, then it is regenerated randomly until it is feasible. Update $PBEST$.

Step 3. Store nondominated solutions in the $NDset$; then, update REP and determine $GBEST$ using the adaptive grid.

Step 4. Update the velocities and positions of the swarms using Equations (6) & (7).

Step 5. Determine whether each component of the current vector satisfies its corresponding boundary. The corresponding component is replaced by the boundary value (pullback technique) if it violates the boundary.

Step 6. Calculate the fitness value of the new position X_i^{k+1}. Then, determine whether the current particle violates the problem-specified constraints. If so, reset X_i^{k+1} to the previous position X_i^k (flyback mechanism).

Step 7. Update $PBEST$ and store the nondominated solutions in the $NDset$; then, update REP and output $GBEST$ using the adaptive grid.

Step 8. Stop the procedure if the stopping criteria are fulfilled; otherwise, return to step 4 and repeat the process.

3. PERFORMANCE-BASED SEISMIC DESIGN

3.1 Loads and constraints for optimum seismic design

This paper adopts a penalty factor measure [5] to specify the level of constraint violation. The constraint parameters were obtained by dividing the earthquake responses by the corresponding allowable values and then subtracting 1 to obtain a dimensionless value.

3.1.1 Gravity load

According to the Chinese Code GB50009-2012 [22], the live load for dwellings is $2.0kN/m^2$, and the dead load varies with the cross sections of the components. According to FEMA-273 [12], the vertical load combination is as follows:

$$Q_G = 1.1(Q_D + Q_L) \tag{8}$$

3.1.2 Constraints on the cross sections of columns between neighbouring floors

Columns located on lower levels of the building should have cross sections that are equal to or greater in size than those of the columns located on upper levels of the building; thus,

s.t. $C_{1i} = \dfrac{A_{i+1}}{A_i} - 1 \le 0$ *i=1,2,...n,* where *n* is the number of columns minus one.

The penalty factor C_1 is expressed as

$$C_1 = \sum_{i=1}^{n} \alpha_{1i} \quad i=1,2,...n \tag{9}$$

Where $\alpha_{1i} = C_{1i}^2$ if $C_{1i} \ge 1$, $\alpha_{1i} = C_{1i}$ if $0 \le C_{1i} \le 1$, and $\alpha_{1i} = 0$ if $C_{1i} \le 0$.

3.1.3 Constraints on the interstory drift

The interstory drift ratio is one of the most important criteria in structural performance. According to FEMA-356 [13], there are four performance levels: Operational (OP), Immediate Occupancy (IO), Life Safety (LS), and Collapse Prevention (CP). Table 1 provides the specific parameters for the different performance levels.

Table 1: Performance levels and interstory drift ratio

Performance level	Interstory drift ratio (IDR)	Probability of exceedance in 50 years
OP	0.65%	50%
IO	1.0%	20%
LS	2.0%	10%
CP	6.1%	2%

s.t. $C_{2i} = \frac{\delta_{i+1} - \delta_i}{h \cdot IDR} - 1 \le 0$ *i=1,2,...n*, where *n* is the number of stories minus one, $\delta_0 = 0$, and *h* is the story height. The penalty factor C_2 is as follows:

$$C_2 = \sum_{i=1}^{n} \alpha_{2i} \quad i=1,2,...n \tag{10}$$

Where $\alpha_{2i} = C_{2i}^2$ if $C_{2i} \ge 1$, $\alpha_{2i} = C_{2i}$ if $0 \le C_{2i} \le 1$, and $\alpha_{2i} = 0$ if $C_{2i} \le 0$.

3.1.4 Constraints on strong columns and weak beams

If a severe earthquake occurs, it is easier to repair damage in a beam than in a column. Furthermore, yielding in columns may trigger the overall collapse of the structure. Thus, the strong column-weak beam design theory is typically adopted:

s.t. $C_{3i} = \frac{I_{bi}}{I_{ci}} - 1 \le 0$ *i=1,2,...n*, where *n* is the number of connections and *I* is the relative moment of inertia of the section.

The penalty factor C_3 is calculated as

$$C_3 = \sum_{i=1}^{n} \alpha_{3i} \quad i=1,2,...n \tag{11}$$

where $\alpha_{3i} = C_{3i}^2$ if $C_{3i} \ge 1$, $\alpha_{3i} = C_{3i}$ if $0 \le C_{3i} \le 1$, and $\alpha_{3i} = 0$ if $C_{3i} \le 0$.

3.1.5 Constraints on strength

According to the Chinese Code GB50017-2003 [23], the stability of columns within the bending matrix of a plane is based on maximum strength theory. The load-carrying capacity of members under pressure and bending is related to the slenderness ratio λ, eccentricity ratio ε, cross section, initial bending matrix, residual stress, and stress-strain relation.

The stability equation of a column within the bending plane is as follows:

$$\frac{N}{\varphi_x A} + \frac{\beta_{mx} M_x}{\gamma_x W_{1x} (1 - 0.8 \frac{N}{N'_{Ex}})} \le f \tag{12}$$

In addition to Equation (12), the stability equation of solid columns under axially loaded compression also considers the following equation:

$$\frac{N}{\varphi A} \leq f \tag{13}$$

The global stability is considered when the strength is calculated for beams, that is, when the global stability equation in the principal plane of the bending components is considered, is expressed as follows:

$$\frac{M_x}{\varphi_o W_x} \leq f \tag{14}$$

In addition to Equation (14), the shear strength of the web component in the principal plane under bending loads is considered for solid beams and is expressed as follows:

$$\tau = \frac{VS}{It_w} \leq f_v \tag{15}$$

Further details of the above equations are provided in Chinese Code GB50017-2003 [23]. Equations (12) to (15) are simplified uniformly as the following expressions:

s.t. $C_{4i} = \dfrac{\sigma_i}{\sigma_{i-allow}} - 1 \leq 0$ *i=1,2,...n,* where *n* is the number of the expected constraint checks for all elements, σ_i is the i^{th} design stress value, and $\sigma_{i-allow}$ is the i^{th} allowable stress value.

The penalty factor C_4 is expressed as follows:

$$C_4 = \sum_{i=1}^{n} \alpha_{4i} \quad i=1,2,...n \tag{16}$$

Where $\alpha_{4i} = C_{4i}^2$ if $C_{4i} \geq 1$, $\alpha_{4i} = C_{4i}$ if $0 \leq C_{4i} \leq 1$, and $\alpha_{4i} = 0$ if $C_{4i} \leq 0$.

Performance-based multi-objective optimisation of large and complex structures requires that all constraints be categorised into two groups [5]. C1 and C_3 constitute the first group, and the fulfilment of these constraints should be determined before the pushover analysis is conducted. If the first group of constraints is not fulfilled, *i.e.*, $C_1+C_3>0$, then its fitness value is *inf*, and no further analyses are required. This mechanism is used to ensure that any feasible solution rigidly satisfies the constraints of the first group. C_2 and C_4 constitute the second group, and the fulfilment of these constraints should be determined after the pushover analysis is conducted. C_2+C_4 is used as the specific level of constraint violation regardless of whether it is equal to zero.

3.2 Nonlinear static procedure

The nonlinear static procedure, also referred to as pushover analysis, is the standard practice for evaluating the seismic demands for the retrofitting of an existing building or the design of a new building. In this method, the structure is loaded with a specific distribution of the lateral loads until the target displacement is reached. The results of the analysis are obtained, including the forces, stresses, hinge states, and displacements.

ATC-40 [12] and FEMA-356 [13] are the two traditional guidelines for pushover analysis. The capacity spectrum method (CSM) in ATC-40 superimposes the capacity spectrum over the response spectrum in ADRS format to obtain the intersection point of the two curves. A fast method of obtaining the performance point is the coefficient method in FEMA-356 [13]. Additionally, the performance point can also be obtained by nonlinear response history analysis (RHA) with a single-degree-of-freedom (SDF) system. This paper uses CSM to perform nonlinear static analysis according to the following procedure:

Step 1. Develop the 5% damped (elastic) response spectrum in ADRS format.

Step 2. Transform the capacity curve into a capacity spectrum using the following equations:

$$S_{ai} = \frac{V_i / G}{\alpha_1} \quad S_{di} = \frac{\Delta_{roof}}{\gamma_1 X_{1,roof}} \tag{17}$$

Step 3. Select a trial performance point (S_{api}, S_{dpi}), as shown in Fig. 6, or any other point chosen based on engineering judgment.

Step 4. Develop a bilinear representation of the capacity spectrum according to the following general process: Set an S_{ay}, which is lower than S_{ai}; then, determine $S_{dy} = S_{d-0.6ay} / 0.6$, where $S_{d-0.6ay}$ is the actual S_d corresponding to $0.6S_{ay}$. If the area A_{BilCur} under the bilinear curve, which should be equal to the area A_{CapCur} under the capacity curve in the range of (0, S_{di}), is within the acceptable tolerance, the bilinear curve is complete; otherwise, set $S_{ay} = S_{ay} \times A_{CapCur} / A_{BilCur}$, and repeat the process.

Step 5. Calculate the spectral reduction factors as given in Equations (18). The demand spectrum and capacity spectrum are shown together in Fig. 6. Fig. 7 is the response spectrum in Chinese code and in ATC-40

$$SR_A = \frac{1}{B_S} = \frac{3.21 - 0.68\ln(\beta_{eff})}{2.12} \quad SR_V = \frac{1}{B_L} = \frac{2.31 - 0.41\ln(\beta_{eff})}{1.65} \tag{18}$$

Step 6. If the intersection (S_{dInter}, S_{aInter}) of the demand spectrum and capacity spectrum is within the acceptable tolerance, the intersection point is considered the performance point. Otherwise, set $S_{ai} = S_{aInter}$, and return to step 4.

The meaning of the performance point is the responses of the structure obtained under a single earthquake action. The 5% response spectrum used in the CSM or in the uniform

building code (UBC) can be transformed by a Chinese seismic design response spectrum as follows:

$$\eta_2 \alpha_{\max} = 2.5C_A \quad T_g = T_s = C_v / 2.5C_A \tag{19}$$

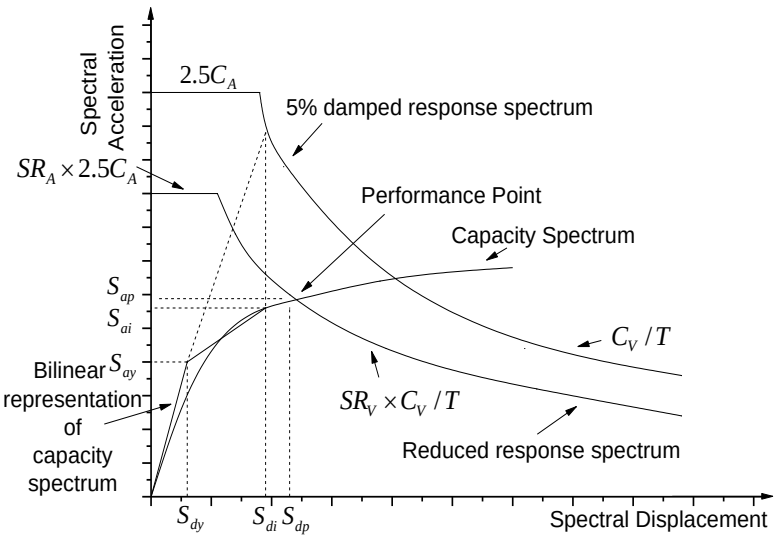
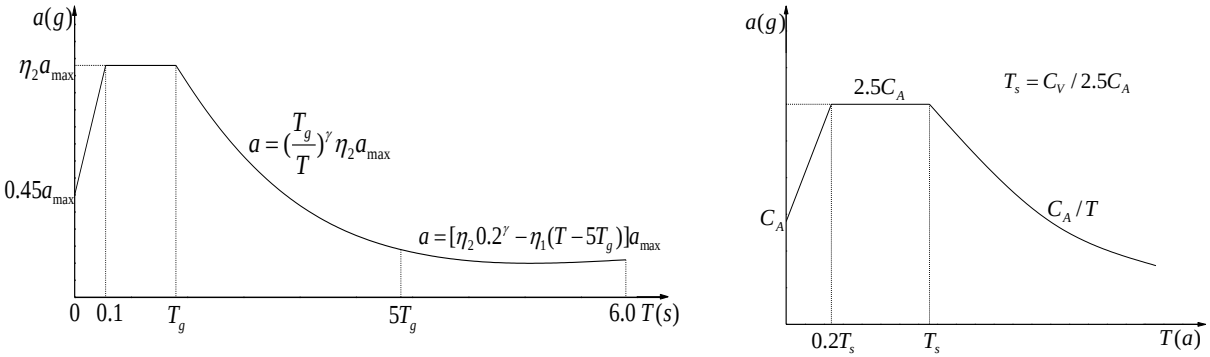


Figure 6. Iteration for determining the performance point



(a) Chinese code (b) ATC-40

Figure 7. Response spectrum in Chinese code and in ATC-40

3.3 Modal pushover analysis

Modal pushover analysis (MPA) [16] was developed to improve the pushover analysis procedure based on structural dynamic theory. Classical pushover analysis considers only the contribution of the fundamental modal, and thus, its results are not sufficiently accurate in mid- to high-rise buildings. The MPA procedure presented in this paper, which includes the contributions of all significant modes of vibration, is as follows:

Step 1. Calculate the natural vibration periods T_n and modes ϕ_n for linear elastic vibration of the building.

Step 2. Develop pushover curves based on the uniform lateral force distribution $s_n^* = M\phi_n$ for the n^{th} mode, where M is the mass matrix of the structure.

Step 3. Obtain the performance point using CSM.

Step 4. Obtain the response r_n of each mode, *e.g.*, forces, stresses, hinge states, and displacements, from the performance point and pushover curves.

Step 5. Repeat steps 2 to 4 for the first n modes (typically the first three modes).

Step 6. Determine the total response using the SRSS rule, $r = \sqrt{\sum r_n^2}$.

4. NUMERICAL EXAMPLES

The multi-objective optimality problems presented in this paper aims to determine the minimum weight of the structure and minimum roof displacements or the minimal maximum node displacement. The basic parameters of all algorithms are set with the same values, *e.g.*, population size, external archive, and number of iterations.

4.1 Linear static analysis examples

4.1.1 The 10-bar planar truss structure

The 10-bar planar truss structure is shown in Fig. 8. Further details on the geometry, material, and load cases are provided by Li and Liu [20]. The design variables are the frame section areas, which are discrete variables. The maximum capacity of EES is 30, and the size of the population is 300. The results obtained with MOHPSO were compared with the results from NSGA-II, MGSO, and MOPSO. Figs. 8 and 10 correspond to the results for 150 and 300 generations, respectively.

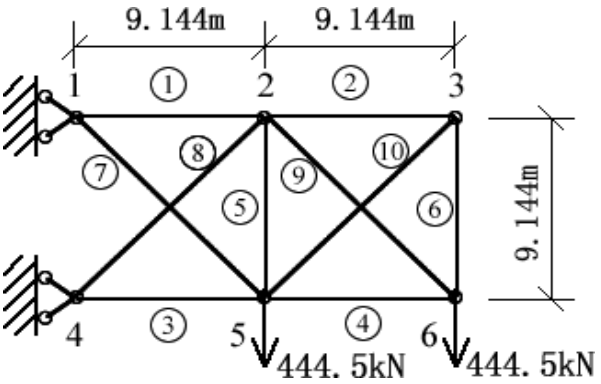


Figure 8. The 10-bar planar truss structure

As shown in Fig. 9, NSGA-II, MOPSO, and MOHPSO clearly outperform MGSO after 150 iterations. Due to the limited capacity of *EES* , the solutions in *EES* are representative, meaning that the Pareto front of the solutions is distributed widely and uniformly. Nondominated solutions in NSGA-II and MGSO are chosen based on the crowded-comparison operator. If most of the individuals are located in part of the Pareto-optimal front, they will not likely be chosen for *EES* even though these individuals have small crowding distances. Thus, the crowded-comparison operator cannot ensure that the Pareto front is always well distributed. Based on Coello [18], the most suitable number of divisions for each dimension in the adaptive grid is 30. The result in Fig. 9 illustrates that the distributions of the Pareto front of MOPSO and MOHPSO are more uniform than that of MGSO. After 300 iterations, although the distributions of the Pareto front of MGSO are the widest and the results of MGSO are closest to the true Pareto optimal front, its result is still dominated by those of the other three algorithms. The results also indicate that MOHPSO is as capable of performing global searches as NSGA-II and MOPSO are, even after many iterations.

4.1.2 The 25-bar spatial truss structure

The 25-bar spatial truss structure is shown in Fig. 11. Further details about the geometry, material, and load cases are provided by Li and Liu [20]. The maximum capacity of *EES* is 30, and the size of the population is 300. The results of MOHPSO were compared with the results of NSGA-II, MOPSO, and MOPSO. Figs. 12 and 13 present the results for 150 and 300 generations, respectively.

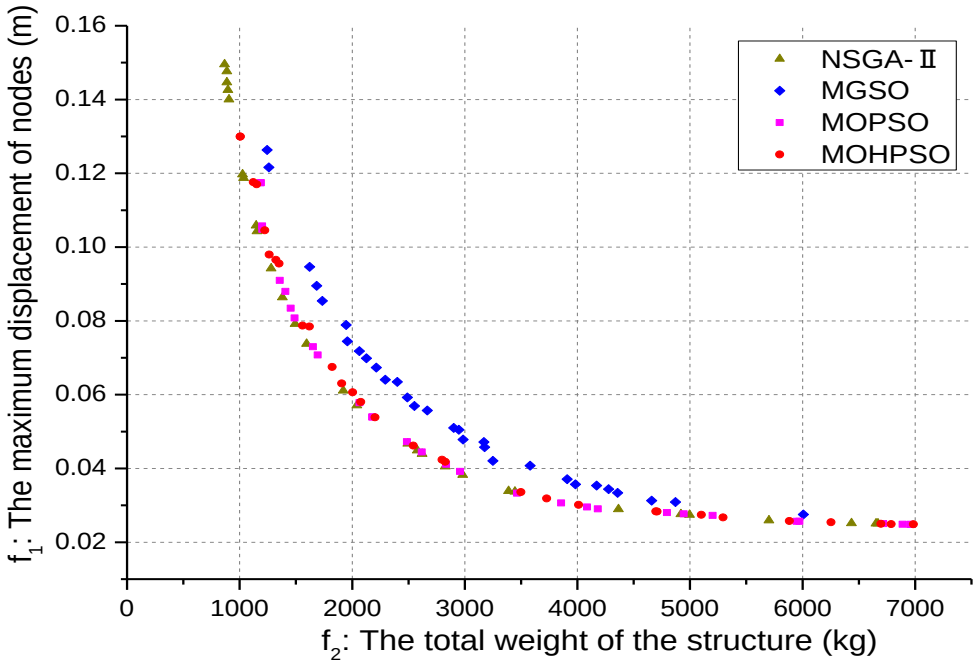


Figure 9. The Pareto-optimal front of the elite set after 150 iterations

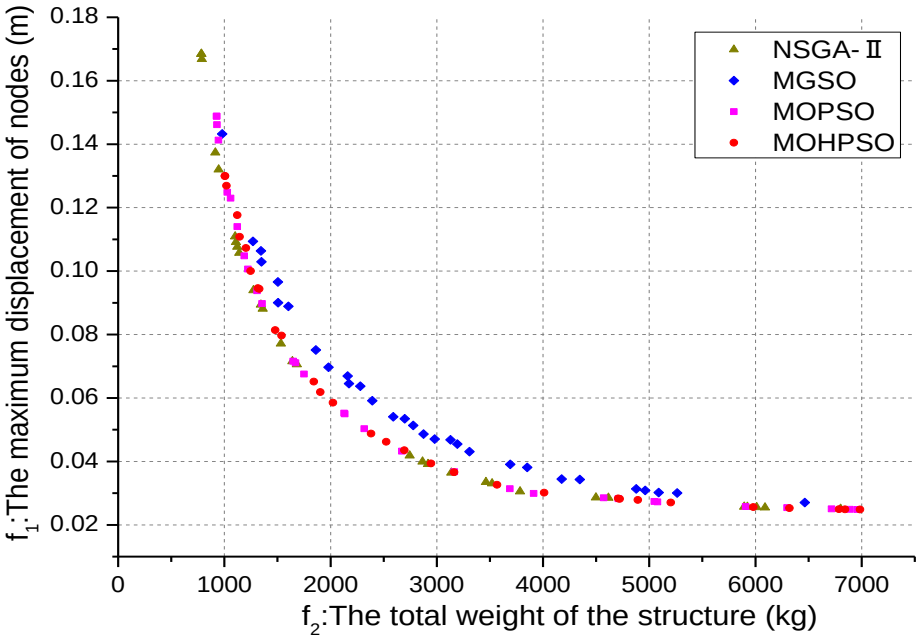


Figure 10. The Pareto-optimal front of the elite set after 300 iterations

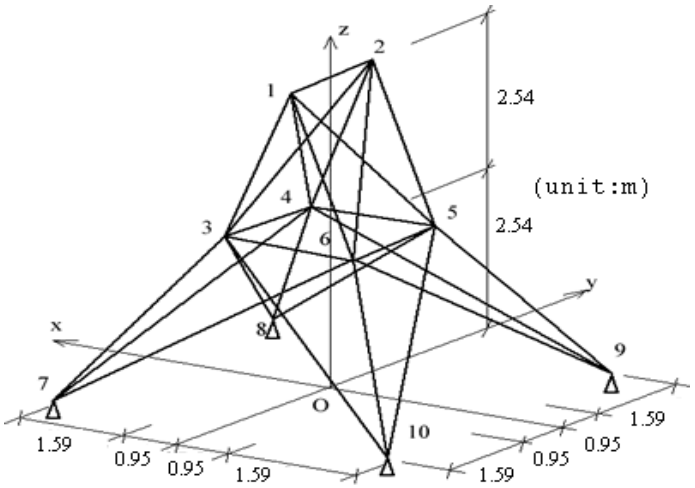


Figure 11. The 25-bar spatial truss structure

This example includes a total of 13 design variables, including 8 discrete section area variables and 5 continuous coordinate variables. Fig. 12 presents the Pareto fronts obtained after 150 iterations, in which the *EES* of NSGA-II dominates those of MGSO, MOPSO, and MOHPSO under the conditions that 20 kg ≤ *weight* ≤ 200 kg and 0.002 m ≤ *max-distance* ≤ 0.025 m, although this superiority is not statistically significant. All of the algorithms display similar performances when 200 kg ≤ *weight* ≤ 500 kg and 0.000 m ≤ *max-distance* ≤ 0.002 m. Figs. 12 and 13 illustrate that MOPSO and MOHPSO are the most uniform, followed by MGSO and then NSGA-II. After 300 generations, the results of

MGSO are closer to the true Pareto-optimal front. However, the superiority of MGSO compared to MOPSO and MOHPSO is not statistically significant.

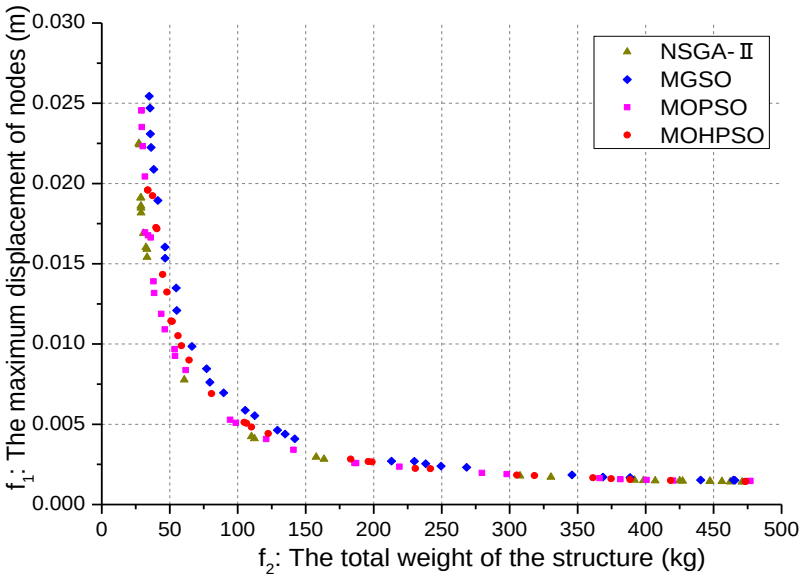


Figure 12. The Pareto-optimal front of the elite set after 150 iterations

EES can be easily maintained and updated using the crowded-comparison operator, from which it is easier to obtain a widely distributed Pareto-optimal front. However, the Pareto-optimal front may not be uniform. The adaptive grid method has the ability to control the evolutionary direction and can obtain a wider and more uniform distribution of the Pareto-optimal front than the crowded-comparison operator method.

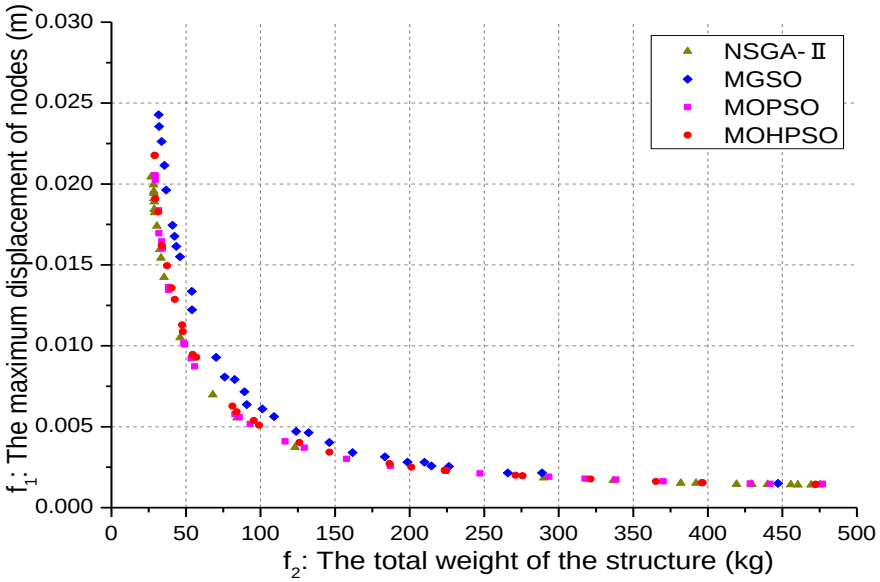


Figure 13. The Pareto-optimal front of the elite set after 300 iterations

4.2 Nonlinear static analysis examples

The geometry of a 5-bay, 9-story steel frame is shown in Fig. 14. The design cases are as follows: the earthquake fortification intensity is 7 degrees, the site classification is Class II, and the design earthquake group is the first group and is a rarely met earthquake condition. The requirement for earthquake performance is LS (life safety) level. Based on Chinese code for the seismic design of buildings [10], the characteristic period *Tg* is 0.35 s and the maximum value of the horizontal seismic influence coefficient $\alpha_{\max}$ is 0.9. The basic parameters for the algorithms are set as follows: the size of the population is 50, the capacity of the *EES* is 50, and the procedure is executed for 100 iterations. The 54 cross sections of the steel members are W-shaped and provided by the manuals of the American Institute of Steel Construction (AISC). The steel grade is Q345 [23]. The first 27 design variables are sections of columns, and the remainder are sections of beams, which are in the ascending order of area, as shown in Table 2.

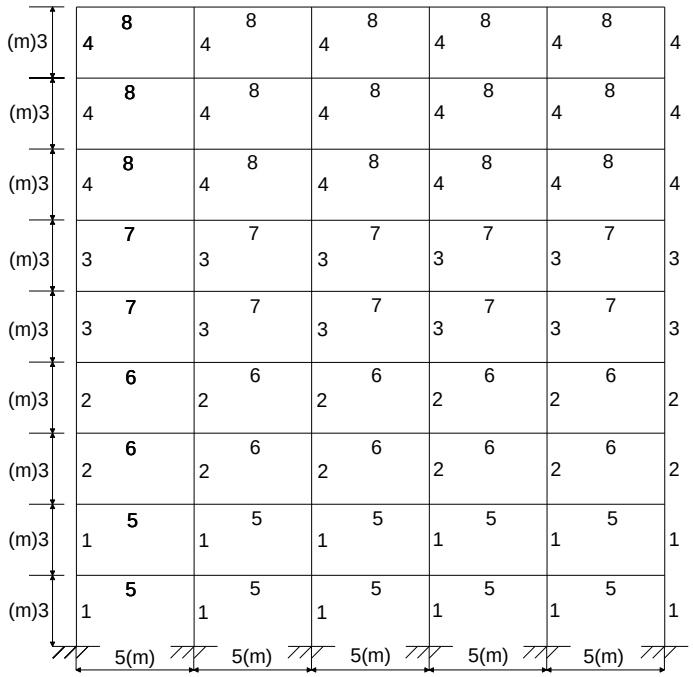


Figure 14. Geometry and member grouping of the 5-bay, 9-story steel frame

Table 2: Discrete sections corresponding to integer numbers

Section	No.	Section	No.
W14×68	1	W16×26	28
W14×74	2	W16×31	29
W14×82	3	W18×35	30
W14×90	4	W18×40	31

W14×99	5	W21×44	32
W14×109	6	W21×50	33
W14×120	7	W24×55	34
W14×132	8	W24×62	35
W14×145	9	W24×68	36
W14×159	10	W24×76	37
W14×176	11	W24×84	38
W14×193	12	W27×84	39
W14×211	13	W30×90	40
W14×233	14	W30×99	41
W14×257	15	W30×108	42
W14×283	16	W30×116	43
W14×311	17	W33×118	44
W14×342	18	W33×130	45
W14×370	19	W36×135	46
W14×398	20	W36×150	47
W14×426	21	W36×160	48
W14×455	22	W36×170	49
W14×500	23	W36×182	50
W14×550	24	W36×194	51
W14×605	25	W36×210	52
W14×665	26	W36×230	53
W14×730	27	W36×232	54

The optimisation procedure is as follows:

Step 1. Initialise the population in the search space randomly.

Step 2. Check the constraint group C_1+C_3 for each individual and perform modal analysis and pushover analysis if C_1+C_3 is equal to zero. Then, calculate the fitness values and the constraint group C_2+C_4. Otherwise, the fitness values will be given *inf* directly, and the calculation for this individual is complete.

Step 3. For MOHPSO, regenerate individuals randomly for those where the constraint values C_1+C_3 and C_2+C_4 are not equal to zero.

Step 4. Maintain and update *NDset* and *EES* based on the current population.

Step 5. Update the population by the strategy of the corresponding algorithm.

Step 6. Calculate the fitness of each population.

Step 7. Repeat step 4, and maintain and update *NDset* and *EES* based on the current population.

Step 8. Stop the procedure if the stopping criteria are fulfilled; otherwise, return to step 5.

4.2.1 Classical pushover analysis result

For the structure in Fig. 14, a pushover analysis is performed using the inertial force distribution of the fundamental mode. The results for MOHPSO, NSGA-II, MOPSO, and

MOPSO after 100 generations are shown in Fig. 15.

As shown in Fig. 15, the Pareto-optimal front of MOHPSO after 100 iterations dominates the solutions of the other algorithms. MOHPSO obtains 37 nondominated solutions even though the capacity of *EES* is limited to the size of 50 individuals. Several of the NSGA-II solutions are not dominated by MOHPSO. However, their front is dispersed and far from the true Pareto-optimal front, illustrating that NSGA-II is not suitable for multi-constraint and multi-objective engineering problems. The front curve of MGSO is smoother than the front curve of NSGA-II but is rather narrow. Furthermore, most of the individuals are dominated by NSGA-II. Coello [18] reported that MOPSO converges rapidly. However, rapid convergence makes it easy to converge to the local optimum, as shown in Fig. 15. The solutions of MOHPSO are demonstrated to dominate those of MOPSO and MGSO.

MOHPSO converges to the true Pareto-optimal front more quickly than the other algorithms because its first generation is feasible. The adopted flyback mechanism contributes more efficient offspring. The adaptive grid increases the uniformity of the external archive, and *GBEST* is more reasonably selected. The total computing time of MOHPSO required for the optimisation is approximately 10 h on an Intel Core i7, 2.4 GHz, but the solution process, which requires 5,000 fitness function evaluations, would have required approximately 100 h if the constraints were not divided into groups.

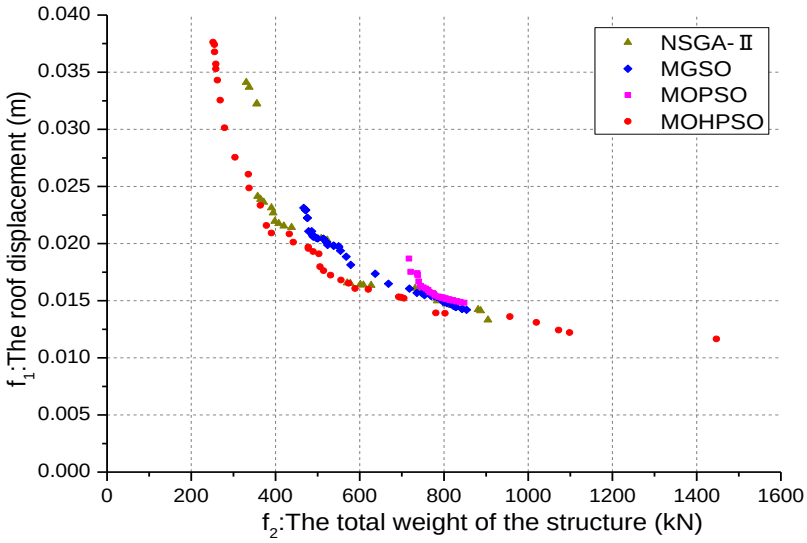


Figure 15. The Pareto-optimal front of the elite set after 100 iterations

4.2.2 Modal pushover analysis result

For the structure given in Fig. 14, analyses were performed using the inertial force distributions of the first three modes, and the final earthquake response results were obtained using SSRS. The solutions for MOHPSO, NSGA-II, MOPSO, and MOPSO after 100 generations are shown in Fig. 16.

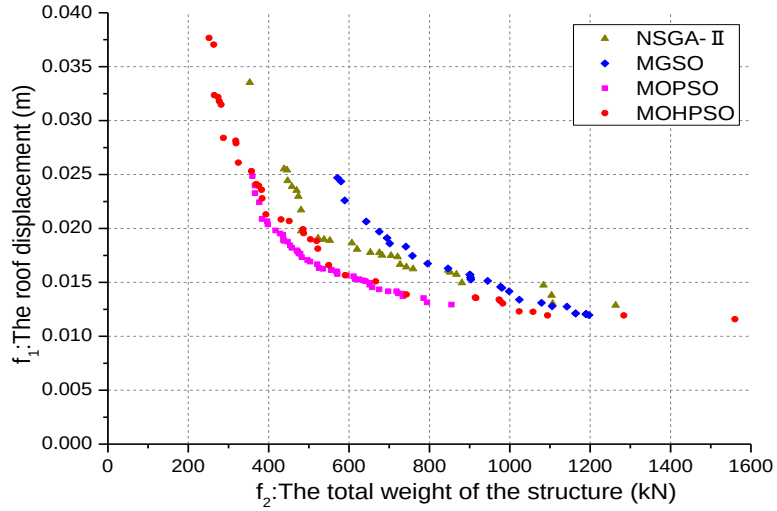


Figure 16. The Pareto-optimal front of the elite set after 100 iterations

Fig. 16 illustrates that the Pareto front curves of NSGA-II and MGSO are similar to the results of classical pushover analysis. The Pareto front curve of MOPSO is even and smooth, in contrast to the Pareto front curve in Fig. 15, it still converges to the local optimum. Although parts of the solutions of MOPSO dominate those of MOHPSO after 100 iterations, the Pareto front curve of MOHPSO is wider and more stable than that of MOPSO. The results illustrate the superiority of the adaptive grid technique, which is more practical than the crowded-comparison operator. In addition, the total computing time MOHPSO requires for optimisation is approximately 20 h on an Intel Core i7, 2.4 GHz, whereas the solution process would be approximately 200 h if the constraints were not divided into groups.

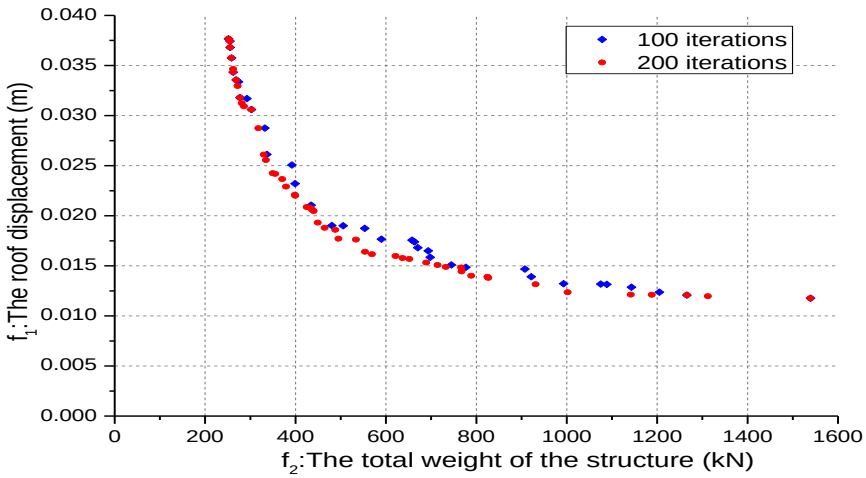


Figure 17. The Pareto-optimal front of the MOHPSO after 100 and 200 iterations

In Fig. 16, MOHPSO obtained 37 nondominated solutions, which did not reach the allowable capacity of *EES* . Fig. 17 presents the solutions of MOHPSO after 200 generations. Most of the solutions after 200 iterations dominate those after 100 iterations,

indicating that MOHPSO is able to obtain a global optimum even in the latter generations. For the complex and time-consuming engineering analysis procedure, it is necessary to develop an optimisation technique that is more efficient, has a faster convergent rate, and requires fewer iterations to obtain the optimal solutions. MOHPSO was designed to fulfil these requirements.

5. CONCLUSION

In this study, a MOHPSO was presented based on HPSO to solve performance-based seismic multi-constraint and multi-objective optimisation problems. Four intelligence algorithms, namely NSGA-II, MOPSO, MGSO, and MOHPSO, were used in this paper for the performance-based optimal design of a plain truss structure, a special structure, and a multilevel frame structure based on weight and displacement optimisation. The results of linear and nonlinear analyses demonstrate that multi-objective optimisation based on swarm intelligence algorithms and the Pareto strategy can provide designers with reasonable choices. The multi-objective optimisation results using the NSGA-II, MOPSO, MGSO, and MOHPSO algorithms demonstrate the superiority and practicability of MOHPSO compared to the other three traditional intelligence algorithms. Although MOHPSO is not the optimal algorithm for all optimisation problems and has some limitations, it is suitable for the multi-objective optimisation of engineering problems.

Acknowledgements: This work is supported by the National Natural Science Foundation of China (51178121) and the Guangdong Natural Science Foundation (S2012020011082).

REFERENCES

1. Fragiadakis M, Lagaros ND. An overview to structural seismic design optimization frameworks, *Comput Struct* 2011; **89**: 1155-65.
2. Cheng W, Liu F, Li LJ. Size and geometry optimization using teaching-learning-based optimization, *Int J Optim Civil Eng* 2013; **3**: 431-44.
3. Li LJ, Huang ZB, Liu F, Wu QH. A heuristic particle swarm optimizer for optimization of pin connected structures, *Comput Struct* 2007; **85**: 340-9.
4. Gholizadeh S, Kamyab R, Dadashi H. Performance-based design optimization of steel moment frames, *Int J Optim Civil Eng* 2012; **3**: 327-43.
5. Kaveh A, Laknejadi K, Alinejad B. Performance-based multi-objective optimization of large steel structures, *Acta Mechanica* 2011; **223**: 355-69.
6. Li G, Jiang Y, Yang DX. Modified-modal-pushover-based seismic optimum design for steel structures considering life-cycle cost, *Struct Multidiscip Optim* 2012; **45**: 861-74.
7. Oskouei AV, Fard SS, Aksogan O. Using genetic algorithm for the optimization of seismic behavior of steel planar frames with semi-rigid connections, *Struct Multidiscip Optim* 2012; **45**: 287-302.
8. Huang JZ, Wang Z. Multiobjective optimization design of a seismic steel frames using genetic algorithm, *Chinese J Theor Appl Mech* 2007; **39**: 389-97.

9. Liu F, Li LJ, Yuan B. Multi-objective optimal design of frame structures with group search optimizer, *The Third International Symposium on Computational Mechanics in Conjunction with the Second Symposium on Computational Structural Engineering*, December, Taipei, 2011, pp. 968-975.
10. GB50011-2010, *Code for Seismic Design of Buildings*, Beijing, China Architecture & Building Press, 2010.
11. Applied Technology Council, ATC-40, *Seismic Evaluation and Retrofit of Concrete Buildings*, Redwood City, CA, Applied Technology Council, 1997.
12. FEMA-273, *NEHRP Guidelines for the Seismic Rehabilitation of Building*, Washington, DC, Federal Emergency Management Agency, SAC Joint Venture, 1997.
13. FEMA-356, *Prestandard and Commentary for the Seismic Rehabilitation of Buildings*, Washington, DC, Federal Emergency Management Agency, SAC Joint Venture, 2000.
14. FEMA-440, *Improvement of Nonlinear Static Seismic Analysis Procedures*, Washington, DC, Federal Emergency Management Agency, SAC Joint Venture, 2005.
15. Chopra AK, Goel RK. A modal pushover analysis procedure for estimating seismic demands for buildings, *Earthquake Eng Struct Dynam* 2002; **31**: 561-82.
16. Chopra AK, Goel RK, Chintanapakdee C. Evaluation of a modified MPA procedure assuming higher modes as elastic to estimate seismic demands, *Earthquake Spectra* 2004; **20**: 757-78.
17. Kennedy J, Eberhart RC. Particle swarm optimization, In: *Proceedings of the IEEE international conference on neural networks*, Perth, Australia, New York: IEEE Press, IV, 1995, pp. 1942-1948.
18. Coello CCA, Pulido GT, Lechuga, MS. Handling multiple objectives with particle swarm optimization, *IEEE Trans Evol Comput* 2004; **8**: 256-79.
19. Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans Evol Comput* 2002; **6**: 182-97.
20. Li LJ, Liu F. *Group Search Optimization for Applications in Structural Desig*n, Heidelberg, Springer-Verlag, 2011.
21. He S, Wu QH, Wen JY, Saunders JR, Paton RC. A particle swarm optimizer with passive congregation, *BioSystems* 2004; **78**: 135-47.
22. GB50009-2012, *Load code for the Design of Building Structures*, Beijing, China Architecture & Building Press, 2012.
23. GB50017-2003, *Code for Design of Steel Structures*, Beijing, China Planning Press, 2003.