Struct Multidisc Optim (2018) 57:799–814
DOI 10.1007/s00158-017-1784-3

RESEARCH PAPER

Large-scale structural optimization using metaheuristic algorithms with elitism and a filter strategy

Hongyou Cao[1] · **Xudong Qian**[1] · **Yunlai Zhou**[1]

Received: 31 March 2017 / Revised: 1 August 2017 / Accepted: 4 August 2017 / Published online: 16 August 2017

Abstract Large-scale structural optimization often requires numerous finite element analyses to assess the feasibility of the derived solutions during the optimization process, which consume most of the computational cost. To enhance the computational efficiency, this study introduces a filter strategy aiming to eliminate the redundant constraint violation evaluations in large-scale structural optimization using metaheuristic algorithms. Based on the solution selection rule, this study separates the metaheuristic algorithms into two categories: replacement and elitism. The filter mechanism founds on elitism of the metaheuristic algorithms and reduces substantially the number of structural analyses without compromising the effectiveness of the optimization algorithms and the constraint handling techniques. This study also defines a parameter, R, to assess the enhancement performance of the computational efficiency improved by the proposed method. Results from both mathematical simulations and two large-scale structural optimization examples using various metaheuristic algorithms demonstrate that the harmony search (HS) leads always to the lowest R value. The R value is less than 0.4 and is even as small as 0.09 for the 942-bar example, which means over 90% of time savings compared with the penalty method and the Deb rule and the quality of the final optimum also does not depend on the value of R.

✉ Xudong Qian
qianxudong@nus.edu.sg

[1] Department of Civil and Environmental Engineering, National University of Singapore, Singapore 117576, Singapore

Keywords Constraint-handling · Structural optimization · Elite selection · Particle swarm optimization · Metaheuristic algorithm · Death penalty method

1 Introduction

Metaheuristic algorithms have extensive applications in structural optimization due to their superior global search capacity and simple iterative mechanism (Astroza et al. 2016; Cao et al. 2017b; Chen et al. 2015; Gandomi et al. 2015). However, their slow convergence rate and enormous computational expenses required in the structural optimization process often desire cost-effective optimization approaches for large-scale structures based on the conventional optimization formulation (Arora and Wang 2005).

Structural optimization using metaheuristic algorithms has undergone tremendous evolutions, which can be summarized into four categories: 1) improving or hybridizing the standard metaheuristic algorithms (Ahrari and Deb 2016; Chen et al. 2015; Farshchin et al. 2016; García-Segura et al. 2015; Kaveh et al. 2014); 2) proposing new metaheuristic algorithms (Abdel-Raouf and Abdel-Baset 2014; Gandomi et al. 2013; Kaveh and Bakhshpoori 2016; Kaveh and Mahdavi 2014; Rashedi et al. 2009); 3) utilizing parallel computing techniques (Agarwal and Raich 2006; Jansen and Perez 2011; Umesha et al. 2005); and 4) incorporating the surrogate modeling approaches (Jin 2011; Lute et al. 2009; Ong et al. 2003; Shan and Wang 2010). The enhancement efficiency in the first two categories is problem dependent in light of the No-Free Lunch theorem (Wolpert and Macready 1997). The third category integrates the metaheuristic algorithms with parallel computing, which, together with a demand for a high performance-computing platform, restricts its applications.

The fourth category normally applies a low-cost surrogate model to substitute the original objective or constraint functions within the optimization process. However, the determination of an accurate surrogate model for high dimensional and strong nonlinear problems often encounters challenges (Li et al. 2016).

Metaheuristic algorithms initially apply to unconstrained problems. In order to accommodate the constraints, various methods have emerged to efficiently handle the constraints (Efren 2009; Jordehi 2015; Mezura-Montes and Coello 2011). The penalty-based methods (Jordehi 2015), including the static penalty functions, dynamic penalty functions, adaptive penalty functions, death penalty, and etc., are widely applicable approaches for their simplicity and ease of implementation in transforming the constrained problem into an unconstrained one where the objective is augmented with penalties proportional to the degree of constraint infeasibility. Unlike the penalty method, another constraint handling techniques like the death penalty approach (Efren 2009; Mezura-Montes and Coello 2011), the Deb rule (Deb 2000) and the bi-objective method (Venter and Haftka 2010) handle the objective function and the constraint violation separately without parameter-tuning. The constraint handling approaches have also undergone certain developments upon using the metaheuristic algorithm search rules, for instance, the flyback mechanism in particle swarm optimization (PSO) (Venter and Sobieszczanski-Sobieski 2003), the segregated genetic algorithm (Le Riche et al. 1995), the constraint-consistent genetic algorithm (Kowalczyk 1997), the filter-genetic algorithm (Tang and Wang 2015), and the mapping strategy for teaching-learning-based optimization (Baghlani et al. 2017). In summary, these constraint-handling approaches guide the search to the feasible regions while assessing both the objective function and the constraint functions for each new trial in the optimization process. Consequently, the indicator measuring the search capability using these approaches is equivalent to that assessing the computational efficiency for a given metaheuristic algorithm since both rely on the amount of objective or constraint violation evaluations in the implemented optimization.

For structural optimization, the objective function often demands low computational cost that may be negligible in comparison with that for constraint evaluations, which require time-consuming structural analyses. Therefore, the computational efficiency of structural optimization mainly depends on the amount of structural analyses and enhancing the computational efficiency of the metaheuristic algorithms will then be feasible by reducing the amount of constraint evaluations during the optimization process but concurrently maintaining the accuracy of the optimal design. Unlike previous constraint handling approaches, which focus on maintaining the feasibility of the solutions and/or facilitating the convergence rate, Kazemzadeh Azad et al. (Kazemzadeh et al. 2013; Kazemzadeh Azad and Hasançebi 2013) developed an upper bound strategy (UBS) to eliminate the structural analyses for the solutions with heavier net weight than the penalized weight of the current history best. Similar to the idea of the UBS, Cao et al. (2017a) introduced a filter strategy into the PSO to filter the redundant structural analyses in the optimization procedure. Compared with the UBS, the filter strategy initializes the swarm with feasible solutions and the upper bound in the filter strategy is the net weight of the best found trials, which can avoid the troublesome selection of the penalty coefficients and always guarantee the feasibility of the designs. Recent researches (Cao et al. 2017b; Kaveh and BolandGerami 2017; Kaveh and Ilchi Ghazaan 2017; Kazemzadeh Azad 2017a; Sheikholeslami et al. 2016) have demonstrated that these strategies can significantly enhance the computational efficiency of the metaheuristic algorithms and show huge potentials in large-scale structure optimization. This study extends the filter strategy proposed in (Cao et al. 2017a) to other metaheuristic algorithms and subsequently develops a general formula for this approach from the basal iterative formulas of the metaheuristic algorithms.

The remainder of this paper is organized as follows. Section 2 presents the general structural optimization formulation, and Section 3 summaries the metaheuristic algorithms from the new solution updating rules and addresses the filter strategy for metaheuristic algorithms with elitism. Section 4 applies the proposed method in five different metaheuristic algorithms and five particle swarm optimization based methods to investigate the performance of the proposed method by mathematical simulations and examine the relationship between the computational efficiency and the quality of the optimal solutions. Section 5 validates the enhancement computational efficiency of the proposed method with two large-scale structural optimization problems compared with the conventional penalty method and Deb rule by computational time. Section 6 summarizes the main conclusions of this study.

2 Structural optimization formulation

Structural optimization aims to identify the lightest or cost-optimal design under various structural functional requirements, e.g., deflection requirements at the serviceability limit sate, stress and stability requirements according to the ultimate limit state design. The optimization model therefore usually contains two parts: the objective function and the constraint functions. As various uncertain factors may affect the final

cost estimation, investigators often define the objective function, Φ, based on the total usage of structural materials,

$$\Phi = \min(W(\mathbf{x})) = \min\left(\sum_{i=1}^{N_m} \rho_i L_i A_i\right) \tag{1}$$

where $\mathbf{x}$ denotes the design variables, W represents the total weight of the structure composed of N_m members. For each member i, the parameters ρ_i, L_i, and A_i refer to the material density, length of the member i, and the cross-sectional area, respectively.

The constraints comprise several inequalities driven by the mechanical requirements of the structure. Assuming that α quantifies the structural performance in term of stresses, displacements, natural frequencies, critical stability coefficients, etc., the constraint functions become,

$$\begin{aligned} &\alpha_j \le \alpha_j^* \quad for \;\; j \in [1, \cdots, N_j] \\ &\alpha_k \ge \alpha_k^* \quad for \;\; k \in [1, \cdots, N_k] \\ &\mathbf{x}_{lower} \le \mathbf{x} \le \mathbf{x}_{upper} \end{aligned} \tag{2}$$

where $\mathbf{x}_{lower}$, $\mathbf{x}_{upper}$ define the lower and upper bounds of the design variables, respectively. α_j and α_k refer to the functional parameters with α_j^* and α_k^* defining the upper and lower bound values, respectively. N_j and N_k indicate the total numbers of the two types of inequality constraints.

Metaheuristic algorithms typically utilize a function ψ to evaluate the constraint violation in the optimization process. $\psi = 0$ implies a feasible design, while $\psi > 0$ indicates that the design variables locate in the infeasible space. The definition of the function ψ follows,

$$\psi(\mathbf{x}) = \max(0, \phi) \tag{3}$$

where ϕ denotes the degree of constraint violation,

$$\phi = \sum_{j=1}^{N_j} \phi_j + \sum_{k=1}^{N_k} \phi_k \tag{4}$$

The functions ϕ_j and ϕ_k evaluate the violation against the upper bound and lower bound constraint requirements, respectively,

$$\phi_j = \begin{cases} f_j \le f_j^* & \phi_j = 0 \\ f_j > f_j^* & \phi_j = \left| \dfrac{f_j - f_j^*}{f_j^*} \right| \end{cases} \tag{5}$$

$$\phi_k = \begin{cases} f_k \ge f_k^* & \phi_k = 0 \\ f_k < f_k^* & \phi_k = \left| \dfrac{f_k - f_k^*}{f_k^*} \right| \end{cases} \tag{6}$$

For the given design variables, $\mathbf{x}$, the objective function is directly computable due to its explicit expression in structural optimization problems. However, the constraints are implicit equations with their evaluations relying on the time-consuming structural analyses especially for large-scale structures. In summary, structural optimization owns a low-cost objective function and high-cost constraint violation evaluation process.

3 Filter strategy for metaheuristic algorithms with elitism

Many constraint-handling techniques appeared to solve constrained problems by extending the application of the metaheuristic algorithms, which are designed originally for unconstrained optimization problems. In contrast to existing constraint-handling approaches, which require evaluations of both the objective value and constraint violations for each possible solution in the optimization process, this section illustrates a filter strategy that eliminates the unnecessary feasibility checks of the solutions in the optimization procedure using metaheuristic algorithms to enhance the computational efficiency.

3.1 Metaheuristic algorithms with elitism

Metaheuristic algorithms solve optimization problems with the following common steps: 1) initializing a series of random solutions; 2) seeking the optimal solution iteratively through some special search rules; 3) printing the best solution found in the search. The second step performs as the kernel for the metaheuristic algorithms simulating the mimicked physical or biological process. It also contains three elements: evaluating the fitness of each solution, producing new solutions and selecting solutions for the next iterative step. Different metaheuristic approaches employ distinct methods to deal with the newly generated solutions and the old ones. However, they can be summarized into two categories: 1) replacement, and 2) elitism.

As demonstrated in Fig. 1, assuming n solutions in each iteration step, the search rule generates n new solutions where the second, third and $(n-1)^{th}$ solutions have better fitness than their corresponding old solutions. The replacement method (as shown in Fig. 1a) substitutes all old solutions with the newly generated ones without considering the fitness information at this stage. The gravitational search algorithm (GSA) (Rashedi et al. 2009) belong to this category. On the other hand, the elitism (as shown in Fig. 1b) updates the old solution by selecting only the new solution with improved fitness. The elitism ensures that the algorithm retains the best solutions found in previous iterations. The personal best and global best

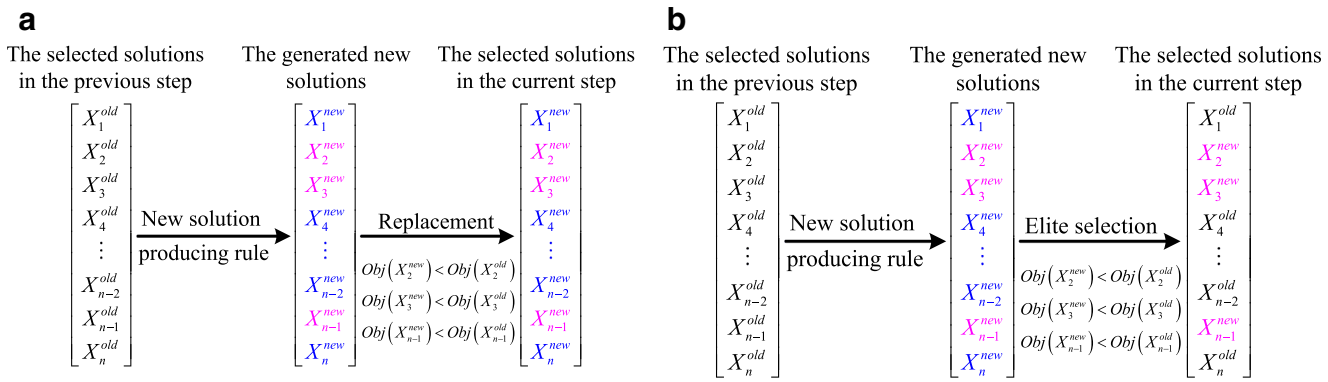

Fig. 1 Two typical solution selection rules in metaheuristic algorithms: (**a**) replacement; (**b**) Elitism

update rules in the PSO and the harmony memory update rule in the harmony search (HS) belong to the elitism category.

Optimization algorithms require calculating each solution's fitness as the search (new solution producing) rule for metaheuristic algorithms based on the fitness information of the current solutions. The number of the objective function evaluations at each iteration step for unconstrained problems thus equals the size of the population in evolutionary algorithms or swarm population in swarm intelligence algorithms. For constrained problems, however, the elitism based selection exhibits apparent advantages over the replacement method while employing the filter strategy proposed in this study.

3.2 Filter strategy

In contrast to unconstrained problems, constrained single-objective optimization problems, as illustrated in Section 2, contain an objective function and a series of constraint functions. The fitness calculation of a solution utilizes both the objective value and the constraint violation information. Thus, the metaheuristic algorithms based on the replacement strategies require the number of constraint violation evaluations equals that of the objective function evaluations in the optimization process. The elitism based selection can obviate this requirement by initializing the solutions in the feasible space and dealing with the objective function and the constraints separately.

In the searching process, each newly generated solution has four possible states compared with its old counterpart. The new solution may locate in four states: 1) in the feasible space with an improved objective value; 2) in the feasible space with a deteriorating objective value; 3) in the infeasible space with an improved objective value; and 4) in the infeasible space with a deteriorating objective value. With all old solutions in the feasible space, the elitism will filter all the new solutions with deteriorating objective values (in state 2 and state 4) without depending on their feasibilities. The constraint violation evaluations for these solutions thus become redundant. Based on this idea, the filter strategy, which predicates on the standard death penalty method (Jordehi 2015), aims to eliminate these redundant constraint violation evaluations. Similar to the death penalty method, this approach also requires initializing all the solutions in the feasible space and employs the following steps in the searching process:

1) Evaluating objective values of the newly generated solutions;

2) Selecting new solutions with improved objective values;

3) Evaluating the constraint violations for these selected new solutions;

4) Choosing the feasible ones and replacing the corresponding old solutions.

These four steps show that the number of the constraint violation evaluations required in the proposed method in each iteration step is smaller than the population of the newly generated solutions. This method founds on the characteristic of the elitism in metaheuristic algorithms and does not compromise the algorithms' search capacity. Compared to the death penalty method, the above approach includes a filter operator to eliminate the redundant constraint evaluations for solutions with deteriorating objective values and does not reduce the efficiency of the constraint handling technique.

Figure 2 depicts the flowchart for filter strategy incorporated in the metaheuristic algorithms with elitism. Unlike the penalty methods that requires a careful fine-tuning of the penalty factor in determining the penalty severity (Kazemzadeh Azad 2017b; Kazemzadeh Azad and Hasançebi 2013), this approach maintains all the performance of the death penalty method and has a highly simplified procedure without additional parameters. The filter mechanism depends solely on the elitism of the metaheuristic algorithms and can thus enhance the computational efficiency of the optimization when coupled with other constraint handling techniques. For instance, it becomes identical to the UBS (Kazemzadeh Azad and Hasançebi 2013) when the filter mechanism is incorporated in the penalty methods. The Deb rule (Deb 2000) can also work with the filter mechanism to reduce the time-

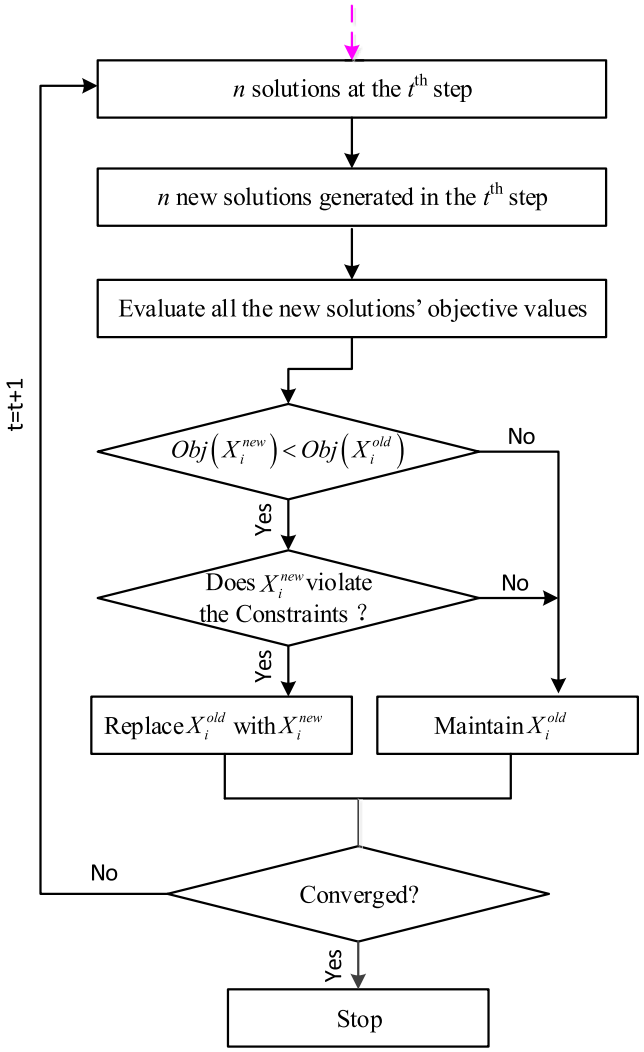

Fig. 2 Flowchart of the metaheuristic algorithms with elitism incorporated with the filter strategy

consuming structural analyses in large-scale structural optimization. Compared to the original constraint handling methods, these approaches coupled with the filter mechanism maintain their original performances with enhanced computational efficiency.

This study defines a factor, R, to measure the computational efficiency of the optimization algorithm,

$$R = \frac{N_{con}}{N_{obj}} \tag{7}$$

where N_{obj} denotes the total number of the objective evaluations. N_{con} refers to the total number of constraint violation evaluations and dominates the computational efficiency in an optimization process. A smaller value of R implies a larger improvement in the computational efficiency using the proposed method. R is equal to 1 for the penalty function method and the Deb rule.

3.3 Initialization of feasible solutions

The success of the search in metaheuristic based optimization also depends on the starting solutions (Maaranen et al. 2007). Similar to the death penalty method, the proposed filter strategy requires initializing solutions in the feasible space. Kazemzadeh Azad et al. (2017b) have demonstrated that seeding the initial population with feasible solutions can improve the computational efficiency of metaheuristic based structural optimization algorithms. This study employs the improved opposition-based initialization strategy (Cao et al. 2017a; Cao et al. 2017b) as outlined below to generate feasible solutions.

1) Randomly produce a solution P in the search space;
2) Calculate the opposition point of P, denoted by OP, in the design space by,

$$\mathbf{x}_{OP} = \mathbf{x}_{upper} + \mathbf{x}_{lower} - \mathbf{x}_P \tag{8}$$

where $\mathbf{x}_{upper}$ and $\mathbf{x}_{lower}$ denote the upper and lower bound of design variables, respectively. $\mathbf{x}_P$ represents the randomly generated solution in step 1 and $\mathbf{x}_{OP}$ refers to the opposite positon of P in the design space.

3) Evaluate the constraint violation of solutions P and OP and calculate their objective values. If both particles reside in the feasible space, select the one with a better objective value, and go back to step 1; if only one of them is feasible, preserve the feasible one as an initial feasible particle and go back to step 1; if both two particles violate the constraints, go back to step 1. Repeat the loop until the number of selected solutions equals the predefined population of the solutions.

Amplifying the values of the size design variables in the lower bound vector, $\mathbf{x}_{lower}$, facilitates the feasible solutions for problems with small feasible ranges. Besides the above generic and stochastic approach, the mapping strategy (Baghlani et al. 2017) can also generate random feasible solutions for structural optimization problem with constraints varying linearly against the design variables.

4 Application on classical problems

This section utilizes five widely used numerical optimization problems to test the performance of the proposed constraint handling method by five standard metaheuristic algorithms with elite selection, including PSO (Trelea 2003), HS (Degertekin 2012), cuckoo search (CS) (Gandomi et al. 2013), flower pollination algorithm (FPA) (Yang 2012) and

teaching-learning-based optimization (TLBO) (Farshchin et al. 2016). This study also employs four improved versions of the PSO including APSO (Nickabadi et al. 2011), BBPSO (Kennedy 2003), IPSO (Wu et al. 2011) and PSOPC (He et al. 2004) to demonstrate the relationship between the search efficiency and the computational efficiency. Each numerical test contains 50 independent runs of the optimization process and the algorithmic parameters utilize those recommended in the corresponding literatures.

The Appendix provides the details on the objective and the constraint functions for the five benchmark examples in (Yu et al. 2016). The total number of objective evaluations for Problem 1, Problems 3, Problem 4 and Problem 5, is 10,000 while that for Problem 2, the welded beam design problem, is 20,000. The population size equals 10 in CS and TLBO and that for FPA and PSO is 20 in all numerical tests. The termination criterion for HS is the maximum number of objective evaluations.

Table 1 lists the statistical results of the five problems using five different algorithms. Most of the algorithms, except HS, yield the optimal solution within 0.01% of the best objective value found so far. Both CS and PSO have identified the best solution for the other four problems in the 50 independent runs. Considering the average and standard deviation values, CS provides the best solutions in Problem 2 and Problem 5, and TLBO performs the best in Problem 1. FPA leads to the best solution in Problem 3 and PSO shows the strongest search ability in Problem 4. Even though HS fails to identify the best solutions for all five problems, the average and standard deviation values in problem 4 demonstrate that it performs better than FPA and TLBO. The results demonstrate that the search capabilities of the algorithms are problem dependent while the total number of objective evaluations remains the same.

Figure 3 compares the average value of R calculated from the five different optimization algorithms. HS has the lowest R value while TLBO gains the largest R value among the five methods. This implies that HS has the highest computational efficiency among the five algorithms. Figure 3 and Table 1 also demonstrate that a larger R value does not correspond to a better solution. For Problem 4 as an example, PSO has the second smallest R, while performing the best among all five methods.

Table 2 compares the statistical results of the five problems using PSO and its variations. PSOPC provides the best solution in Problem 1, Problem 2, Problem 3 and Problem 5. APSO performs the best in Problem 4. The advanced PSO approaches do not always perform better than the standard PSO for all the problems. This implies again that the algorithm search ability is problem dependent. Figure 4 shows the average R values calculated by different PSO methods. In contrast to Fig. 3, the variations of the R values for the same problem among PSO-based methods become much smaller. IPSO shows the highest computational efficiency among the five algorithms. PSOPC has the lowest R value in Problem 2 and the second lowest R values in Problem 1 and Problem 3. For these problems, PSOPC also provides the optimal solutions. PSOPC not only exhibits strong search ability, but also has a high computational efficiency in these three problems. This implies again that the accuracy of the optimal solution is independent of the R value.

Table 1 Comparison of the statistical results of the five problems using different metaheuristic algorithms

Algorithms		Problem 1	Problem 2	Problem 3	Problem 4	Problem 5
CS	Best	−6961.616	***1.725***	***1.267E-02***	***2994.471***	***263.896***
	Average	−6956.657	**1.725**	1.272E-02	2994.475	**263.896**
	Standard deviation	6.78	**4.27E-04**	7.18E-05	4.95E-03	**2.56E-10**
FPA	Best	−6961.624	***1.725***	***1.267E-02***	2996.131	***263.896***
	Average	−6958.854	1.726	**1.270E-02**	3001.268	263.896
	Standard deviation	3.27	4.42E-04	**2.42E-05**	3.02	1.00E-05
HS	Best	−6901.341	1.858	1.333E-02	2996.065	263.897
	Average	−6579.284	2.042	1.484E-02	2997.982	264.018
	Standard deviation	1.62E + 02	1.22E-01	6.63E-04	1.23	1.35E-01
PSO	Best	−6961.698	***1.725***	***1.267E-02***	***2994.471***	***263.896***
	Average	−6959.402	1.778	1.336E-02	**2994.471**	263.899
	Standard deviation	2.11	9.69E-02	7.56E-04	**1.58E-03**	8.21E-03
TLBO	Best	−6961.745	1.744	***1.267E-02***	2997.463	263.897
	Average	**−6960.557**	1.804	1.271E-02	3019.219	263.989
	Standard deviation	**1.90**	3.13E-02	3.61E-05	2.48E + 01	1.77E-01
The best solution so far		−6961.814	1.725	1.267E-02	2994.471	263.896

The data in bold denote the best average and standard deviation results for each problem among different methods.
The data in bold-italic indicates that the best solution obtain by the algorithm equals that of the best solution so far.

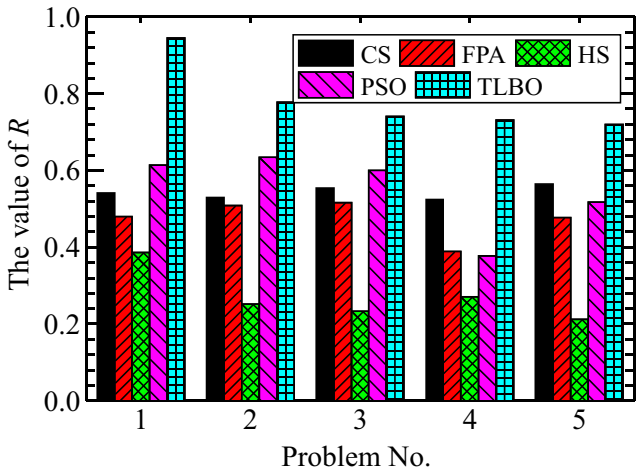

Fig. 3 The average R calculated by different types of metaheuristic algorithms

5 Applications on large-scale structure optimization

This section employs two large-scale structure optimization problems, including a 26-storey, 942-bar truss tower (Hasançebi 2008; Hasançebi and Erbatur 2002) and a three-span suspension bridge with a total length of 1800 m (Cao et al. 2017b), to demonstrate the computational efficiency of the proposed constraint handling approach. This study utilizes the former benchmark example to examine the computational efficiency of different optimization algorithms. The suspension bridge example aims to investigate the efficiency of the proposed method in comparison with the penalty function approach and the Deb rule.

5.1 26-Storey 942-bar truss tower

Figure 5 shows a 26-story space truss tower consisting of 942 bars and 244 nodes. This problem aims to identify the lightest design with the design variables defined as the member cross sectional areas and divided into 59 groups as shown in Fig. 5. The loading on the tower includes,

1) The vertical loads at each node in the first, second and third sections are −13.344 kN (−3.0 kips), −26.688 kN (−6.0 kips) and −40.032 kN (−9 kips), respectively;

2) The horizontal loads equal 6.672 kN (1.5 kips) in the x-direction at each node on the left half of the tower and 4.448 kN (1.0 kips) in the x-direction at each node on the right half;

3) The horizontal load in the y-direction has a magnitude of 4.448 kN (1 kips) applied on all nodes.

The density and elastic modulus of the material are 2767.99 kg/m^3 (0.1 lb./in^3) and 69 GPa (1.0 × 10^4 ksi), respectively. The constraint conditions include allowable stresses and displacements for the truss tower. The maximum allowable stress in each member under tension and compression equals 172.37 MPa (25 ksi) while the maximum allowable displacement in x, y, z direction for the all the nodes is 38.1 cm (15.0 in). The cross-sectional area of each bar is an integer in in.^{2}, which ranges in 1.0 in.^{2} (6.45 cm^2) and 200 in.^{2} (1290.32 cm^2).

Figure 6 depicts the convergence curves of the 942-bar problem using CS, FPA, HS, IPSO and TLBO. The maximum number of the objective function evaluations in each optimization run is 80,000. Figure 6a illustrates the evolution of the weight of the tower versus the number of the objective evaluations, where TLBO fails to solve this high-dimensional

Table 2 Comparison of the statistical results of the five problems using different PSO methods

Algorithms		Problem 1	Problem 2	Problem 3	Problem 4	Problem 5
APSO	Best	−6961.013	***1.725***	***1.267E-02***	***2994.470***	***263.896***
	Average	−6953.387	1.858	1.344E-02	**2994.470**	263.901
	Standard deviation	5.73	2.03E-01	1.01E-03	5.88E-05	1.03E-02
BBPSO	Best	−6958.416	***1.725***	***1.267E-02***	***2994.470***	***263.896***
	Average	−6947.995	1.858	1.384E-02	2995.102	263.899
	Standard deviation	1.00E + 01	1.66E-01	1.45E-03	2.56	3.89E-03
IPSO	Best	−6952.918	***1.725***	1.268E-02	2994.471	***263.896***
	Average	−6924.154	1.845	1.329E-02	2994.471	263.899
	Standard deviation	2.89E + 01	1.45E-01	4.38E-04	2.01E-07	4.84E-03
PSOPC	Best	−6961.625	***1.725***	***1.267E-02***	***2994.470***	***263.896***
	Average	**−6959.736**	**1.725**	**1.305E-02**	2994.986	**263.896**
	Standard deviation	3.76	2.23E-04	4.56E-04	1.49	1.39E-03
PSO	Best	−6961.698	***1.725***	***1.267E-02***	2994.471	***263.896***
	Average	−6959.402	1.778	1.336E-02	2994.471	263.899
	Standard deviation	**2.11**	9.69E-02	7.56E-04	1.58E-03	8.21E-03
The best solution so far		−6961.814	1.725	1.267E-02	2994.4706	263.896

The data in bold denote the best average and standard deviation results for each problem among different methods.
The data in bold-italic indicates that the best solution obtain by the algorithm equals that of the best solution so far.

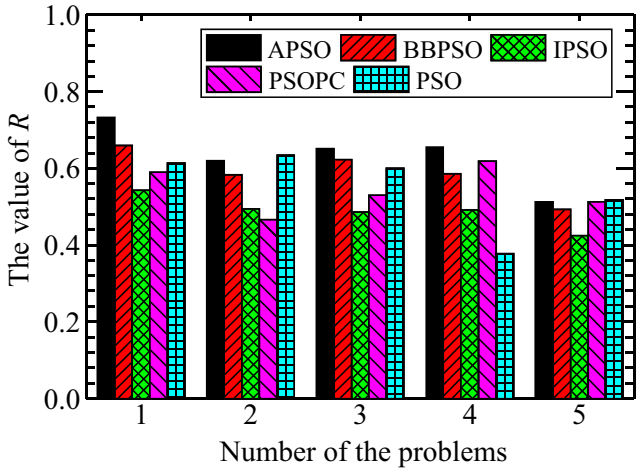

Fig. 4 The average *R* calculated by different types of PSO algorithms

optimization problem, while the other four algorithms converge to a narrow weight range, between 1.20×10^5 lb. (5.44×10^4 kg) and 1.40×10^5 lb. (6.35×10^4 kg). Compared with the randomly generated feasible solutions, these optimizations lead to about 90% of material savings. Figure 6b shows the evolution of the objective value versus the number of structural analyses (constraint violation evaluations). To accomplish the 80,000 objective evaluations, the numbers of structural analyses required by different algorithms are apparently different. HS owns the highest computational efficiency while TLBO requires the most computational efforts even though it failed to find the optimum solution. Table 3 lists the value of *R* and the computational time for this 942-bar problem. As this problem has a wide search domain, the *R* value for each algorithm decreases compared with those in Fig. 3. The *R* value for HS is only 0.09, which implies the proposed constraint handling method leads to more than 90% of time savings. The computational time by HS is only about 24.4 min while that by TLBO (with an *R* value of 0.77) is 117.0 min.

Hasançebi and his co-workers have utilized the simulated annealing (SA) (Hasançebi and Erbatur 2002) and an adaptive evolution strategy (AES) (Hasançebi 2008) to seek the optimal design for this 942-bar tower. Table 3 compares the best results obtained in (Hasançebi 2008; Hasançebi and Erbatur 2002) and this study. The lightest weight gained by Hasançebi (2008) is about 141,243.7 lb. using AES with 150,000 structural analyses. Besides TLBO, all the other four methods yield much lighter optimal designs than the best result by AES. The

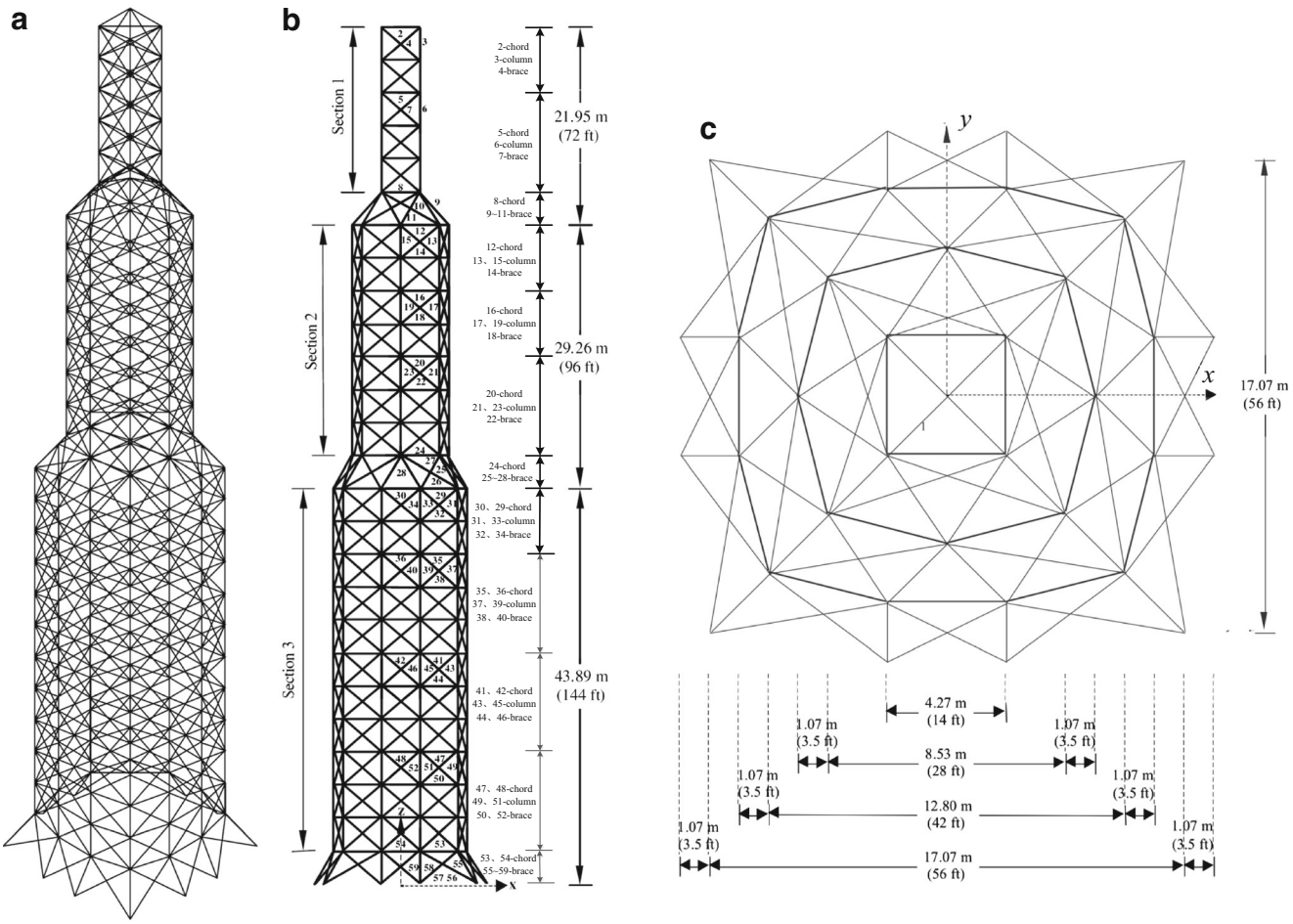

Fig. 5 The configuration of the 942-bar truss: (**a**) isometric view; (**b**) front view; and (**c**) section view of section 3

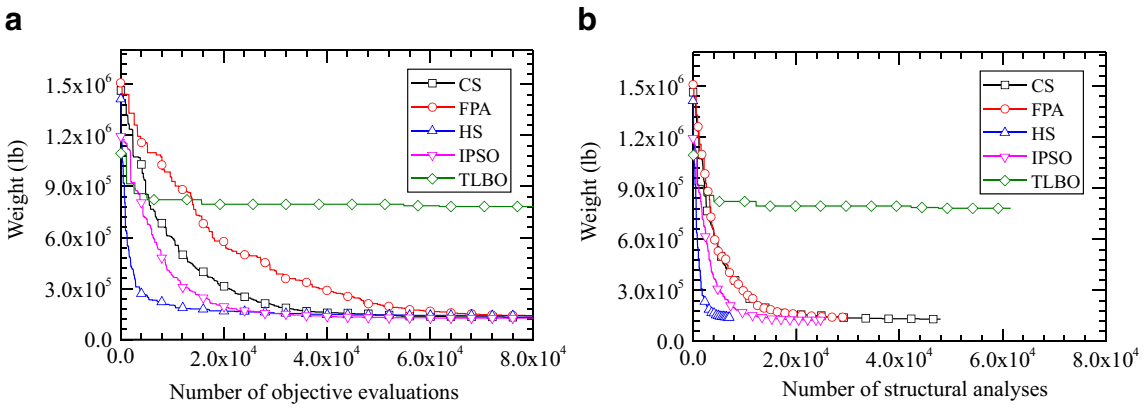

Fig. 6 The convergence curves of the 942-bar truss problem with respect to: (**a**) no. of objective evaluations; and (**b**) no. of structural analyses

optimal weight obtained by IPSO, based on 24,740 structural analyses, is only about 90% of the weight by AES. HS requires 7142 structural analyses while identifying a much better optimum than AES.

5.2 A three-span suspension bridge

Figure 7 depicts the layout of a three-span suspension bridge with a total length of 1800 m and a width of 40 m. L_s and L_m denote the length of the side-span and the main-span, respectively. f refers to the sag of the cable at the mid-span. The shortest hanger at the main-span, H_s, remains 6 m and the distance between the two adjacent hangers is 15 m. Both the horizontal distance between the end of the bridge and the cable anchorage and the height of the pylon below the deck equal 30 m. Figure 7b sketches the cross section of the steel pylon and the cross beam. The geometric and size parameters of the stiffeners are constant to reduce the number of the design variables. The upper and lower cross beams have the same section and their width equal to W_1. Figure 7c shows the section for the girder. This example also assumes all the hangers with the same cross-sectional area. Therefore, this suspension bridge optimization problem includes 16 design variables (as shown in Fig. 7):

Pylon: The width of the pylon (W_1), the height to width ratio (W_2/W_1), the thickness of the plates (T_1 and T_2);

Cross beam: The height of the cross beam (W_3), the thickness of the plates (T_3 and T_4);

Stiffening girder: The bottom width and the height of the girder (W_4 and W_5), the plate thickness of the girder (T_5, T_5 and T_7),

Cable system: The sag-to-span ratio (f/L_m), the area of the main cable (A_1) and hanger (A_2).

Bridge layout: The side-to-central span ratio (L_s/L_m).

(9) to (10) describe the effective areas for the pylon, cross beam and girder, respectively, to include the stiffeners and the diaphragms in these members.

$$A_{pylon} = (2W_1T_2 + 2T_1W_2 - 4T_1T_2) + 2(W_1 + W_2)T_s^p + W_1W_2T_d^p \Big/ D_p \tag{9}$$

$$A_{cross} = 2W_1T_4 + 2W_3T_3 - 4T_3T_4 + 2(W_1 + W_3)T_s^c + W_1W_5T_d^c \Big/ D_c \tag{10}$$

$$A_{girder} = (BT_6 + W_4T_7 + 2W_5T_5 - 2T_5T_6 - 2T_5T_7) + \left(B + W_4 + 2W_5 \Big/ \cos\beta\right)T_s^g \times 84 \Big/ 64 + W_5(W_4 + B) \Big/ 2 \times T_d^g \Big/ D_g \tag{11}$$

The area calculation of the diaphragms assumes the diaphragms without holes. $T_s^p = 15$ mm, $T_s^c = 8$ mm and $T_s^g = 8$ mm denote the thickness of the stiffeners in pylon, cross beam and girder, respectively. $T_d^p = 20$ mm, $T_d^c = 15$ mm and $T_d^g = 15$ mm represent the thickness of the diaphragms in pylon, cross beam and girder while D_p, D_c and D_g refer to the longitudinal distance of the diaphragms in pylon, cross beam and girder and are equal to 3 m.

Table 3 Comparison of the optimization results for the 942-bar problem

Variables	SA	AES	CS	FPA	HS	IPSO	TLBO
A1 (in^2)	1	1.02	3	8	2	1	22
A2 (in^2)	1	1.04	1	5	3	1	46
A3 (in^2)	3	2.94	6	9	2	4	141
A4 (in^2)	1	1.92	3	3	3	1	42
A5 (in^2)	1	1.03	1	3	3	1	31
A6 (in^2)	17	14.96	13	16	11	15	32
A7 (in^2)	3	3.07	3	4	9	3	1
A8 (in^2)	7	6.78	11	15	2	7	40
A9 (in^2)	20	18.58	18	20	23	18	119
A10 (in^2)	1	2.42	2	4	5	3	59
A11 (in^2)	8	6.58	9	5	4	6	1
A12 (in^2)	7	6.29	6	6	6	5	1
A13 (in^2)	19	15.38	18	19	10	13	7
A14 (in^2)	2	2.10	2	2	2	2	73
A15 (in^2)	5	6.02	8	9	11	6	42
A16 (in^2)	1	1.02	1	2	1	1	39
A17 (in^2)	22	23.10	21	13	18	19	145
A18 (in^2)	3	2.89	3	2	3	3	1
A19 (in^2)	9	7.96	10	14	11	12	47
A20 (in^2)	1	1.01	1	1	2	1	7
A21 (in^2)	34	28.55	28	18	24	27	8
A22 (in^2)	3	3.35	3	3	4	3	96
A23 (in^2)	19	16.14	17	19	17	15	40
A24 (in^2)	27	24.82	19	23	20	22	28
A25 (in^2)	42	38.40	39	55	28	35	1
A26 (in^2)	1	3.79	9	1	9	1	89
A27 (in^2)	12	12.32	9	14	7	11	28
A28 (in^2)	16	17.04	11	16	10	14	83
A29 (in^2)	19	14.73	13	17	19	12	11
A30 (in^2)	14	15.03	13	14	10	13	120
A31 (in^2)	42	38.60	29	29	29	35	27
A32 (in^2)	4	3.51	3	4	3	3	37
A33 (in^2)	4	3.00	3	5	2	2	1
A34 (in^2)	4	3.06	3	5	5	2	46
A35 (in^2)	1	1.09	1	1	2	1	20
A36 (in^2)	1	1.46	2	3	3	1	144
A37 (in^2)	62	59.43	57	44	41	54	41
A38 (in^2)	3	3.63	3	3	3	3	2
A39 (in^2)	2	1.89	3	5	4	4	21
A40 (in^2)	4	4.07	3	3	5	3	80
A41 (in^2)	1	1.60	2	2	2	1	1
A42 (in^2)	2	3.67	4	2	3	1	107
A43 (in^2)	77	79.51	60	63	63	65	138
A44 (in^2)	3	3.39	3	3	4	3	38
A45 (in^2)	2	1.58	3	3	3	4	74
A46 (in^2)	3	4.20	3	4	7	3	12
A47 (in^2)	2	1.33	1	4	3	1	84
A48 (in^2)	3	2.24	1	2	2	1	99
A49 (in^2)	100	96.89	70	72	74	91	40

Table 3 (continued)

Variables	SA	AES	CS	FPA	HS	IPSO	TLBO
A50 (in^2)	4	3.71	4	4	4	3	16
A51 (in^2)	1	1.06	7	6	5	2	154
A52 (in^2)	4	4.57	5	5	6	4	4
A53 (in^2)	6	9.61	21	26	21	13	16
A54 (in^2)	3	2.98	14	38	15	8	106
A55 (in^2)	49	45.92	48	53	47	42	80
A56 (in^2)	1	1.00	1	1	3	1	1
A57 (in^2)	62	62.43	36	35	39	55	47
A58 (in^2)	1	2.98	8	13	4	4	44
A59 (in^2)	3	1.00	9	4	12	1	52
Weight (lb)	143,439.5	141,243.7	131,249.1	139,589.3	139,866.1	**127,231.0**	781,697.1
R	1.00	1.00	0.60	0.38	**0.09**	0.31	0.77
Number of structural analyses	39,834	150,000	47,973	30,263	**7142**	24,740	61,566
Computational time (min)	-	-	104.8	72.5	**24.4**	59.3	117.0

The bold entires denote the best results among the different optimization algorithms.

The total material usage of the bridge follows:

$$\Phi = 4A_{pylon}(f + 36) + 4 \times 40 \times A_{cross} + A_{girder}(2L_s + L_m) + A_1 L_c + A_2 L_h \quad (12)$$

where L_c and L_h represent the total length of the main cable and hangers, respectively. The value of L_c and L_h derive from the analytical form-finding method proposed in (Chen et al. 2015, 2013).

The elastic modulus and density of steel Q345 for the stiffening girder, pylons and cross beams are 205 GPa and 7.85×10^3 kg/m^3, respectively and those for the cable system, which are made of parallel steel wires, are 205 GPa and 8.005×10^3 kg/m^3. The side-to-central span ratio is a discrete variable to maintain the length of the main span is in multiples of 30 m. This example considers three types of loads: the dead load, the live load and the static wind load. The dead load (DL) imposing on the girder equals $\rho g A_{girder}$ (ρ denotes the density of the girder and g refers to the gravity acceleration) with an additional uniform load of 60 kN/m. Figure 8 shows the three typical unfavourable live load (LL) cases for a three-

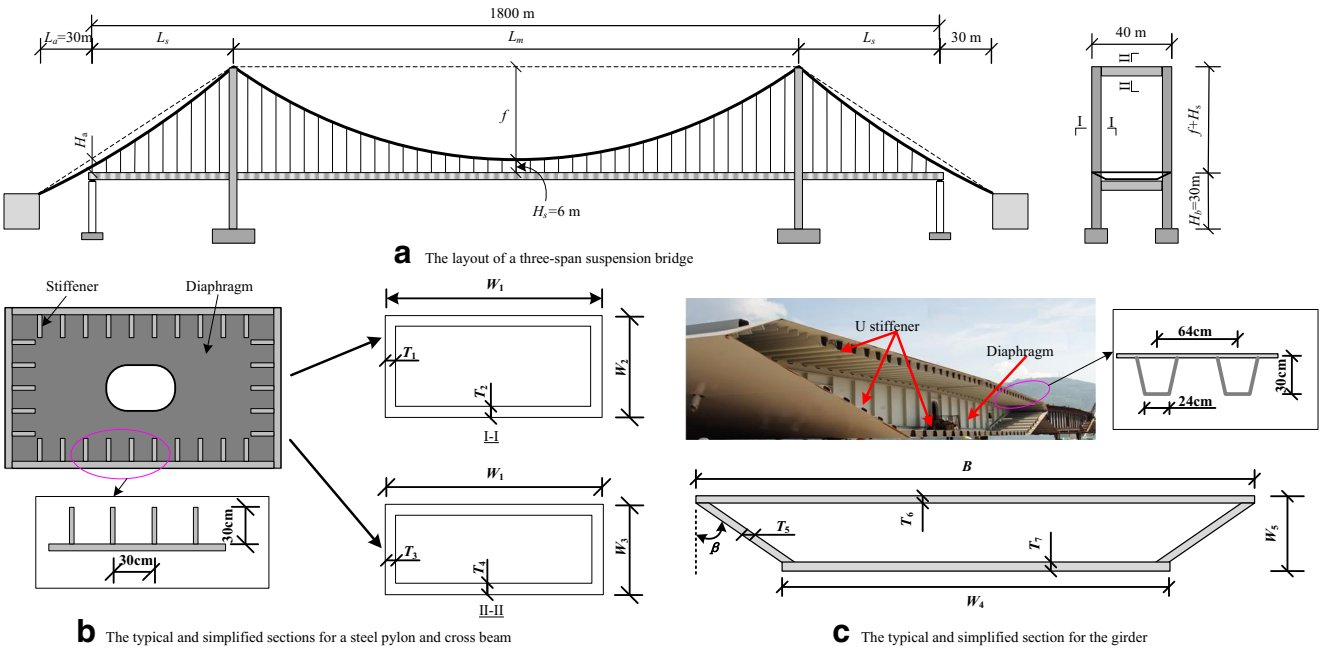

Fig. 7 The geometric parameters of a typical three-span suspension bridge

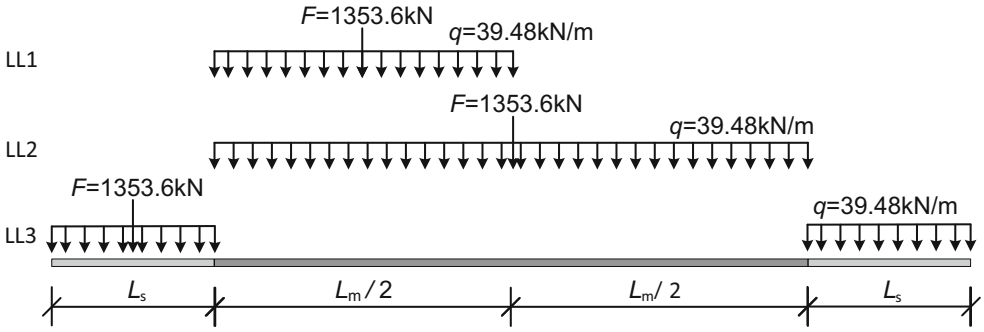

Fig. 8 The three live load cases considered in the optimization

span suspension bridge. The lateral wind load (WL) acting on the pylon equals $1.5W_1$ kN/m and the force acting on the girder follows:

$$q_{wind} = 1.5W_5 \cdot \min\left(0.3, \frac{\beta}{200}\right) \text{ kN/m} \quad (13)$$

where β denotes the angle illustrated in Fig. 7c. This problem considers four load combinations: 1) DL, 2) DL + LL1 + WL, 3) DL + LL2 + WL and 4) DL + LL3 + WL. Table 4 lists the constraints derived from the strength, serviceability and the geometric requirement of the bridge. This study implements the optimization procedure and the form-finding analysis of the suspension bridge in Matlab coupled with FE analyses in ANSYS.

To demonstrate the efficiency of the proposed constraint handling method, this study utilizes the HS for its high computational efficiency and IPSO due to its strong search ability. The two widely used constraint handling approaches, the penalty function method and the Deb rule, coupled with IPSO are also employed for comparison. The maximum number of the objective function evaluations is 40,000 for all the tests and the population of the particle in all IPSO tests equals 40. In the penalty method based optimization, the penalty factor is fixed at 100 during the optimization process.

Figure 9 compares the evolution of the objective value versus the number of structural analyses for the four optimization procedures. Since the proposed constraint handling method initializes the solutions in the feasible space, the initial objective values in the two proposed methods are much smaller than those in the penalty method and the Deb rule. The objective values for the two commonly used methods do not always decrease as the iteration increases due to the infeasible solutions found at the initial search stage. The proposed constraint handling approach dramatically reduces the number of the structural analyses. Table 5 lists the optimization results for the three-span suspension bridge. IPSO coupled with the proposed constraint handling technique performs the best among the four methods. All the identified optimal results are feasible, except for the penalty method with a constraint violation of 1.59×10^{-4}, which defines an infeasible solution based on 4. The number of structural analyses required by HS equals 9812, which implies more than 75% enhancement in the computational efficiency than the other two methods. As shown above, HS suffers from premature convergence, and leads to the heaviest design, with just 3% heavier than the lightest solution among the other four methods. Thus, HS is still a competitive method due to its high computational efficiency. Among the three IPSO-based methods, the proposed approach has obvious advantages, with the lightest design and the least number of structural analyses, about 65% time savings compared to the penalty method and Deb rule. Table 5 also lists the computational time for each method. The time required by Deb rule in each run is about 57.41 h, slightly longer than the penalty method due to its complex algorithmic structure. The computational time for IPSO and HS using the

Table 4 Constraints and their allowable values

Items	Limited values
Stress of the main cable	≤ 500 MPa
Stress of the hangers	≤ 500 MPa
Stress of the stiffening girder	≤ 120 MPa
Stress of the pylons and cross beams	≤ 150 MPa
The vertical deflection of the girder	≤ L_m/500
The displacement of the pylon	≤ (36 + f)/800
The length of the shortest hanger in side-span	≥ 2 m

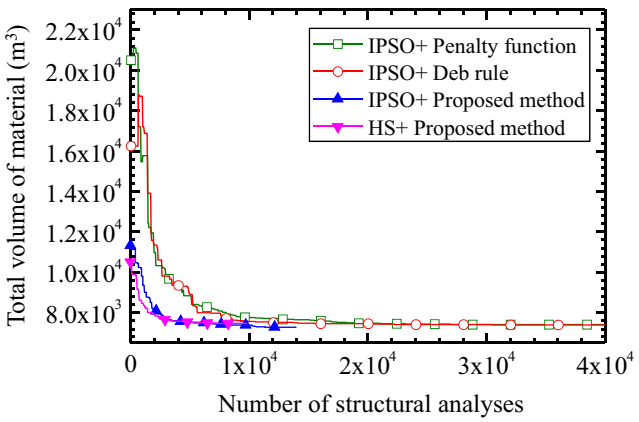

Fig. 9 The evolution curve of the objective value versus the number of structural analyses

Table 5 Comparison of the optimization results for the three-span suspension bridge

Variables	Range	Penalty method	Deb rule	Present work	
		IPSO	IPSO	IPSO	HS
W_1 (m)	[4.0, 10.0]	6.077	5.391	5.55	6.637
W_2/W_1 (m)	[0.6, 0.8]	0.713	0.792	0.797	0.713
W_3 (m)	[3.0, 6.0]	5.945	5.884	6.000	5.950
W_4 (m)	[20, 38]	20.000	20.000	20.000	20.020
W_5 (m)	[2.5, 6.0]	2.500	2.500	2.500	2.501
T_1 (mm)	[20, 200]	34	70	20	49
T_2 (mm)	[20, 200]	148	140	171	127
T_3 (mm)	[8, 80]	79	74	77	71
T_4 (mm)	[8, 80]	79	80	80	73
T_5 (mm)	[8, 80]	8	8	8	8
T_6 (mm)	[8, 80]	8	8	8	8
T_7 (mm)	[8, 80]	8	8	8	9
A_1 (m2)	[0.400, 2.000]	0.7568	0.7631	0.6817	0.7411
A_2 (m2)	[0.003, 0.020]	0.0046	0.0046	0.0046	0.0049
f/L_m	[1/15, 1/6]	1/13.55	1/13.63	1/11.98	1/13.26
L_s/L_m	[0.1, 0.5]	0.250	0.250	0.232	0.250
The volume of steel (m^3)		7414.56	7396.47	**7274.86**	7467.91
Constraint violation		**1.59E-04**	0	0	**0**
Number of structural analyses		40,000	40,000	13,956	**9812**
R		1	1	0.3489	**0.2453**
computational time (hours)		55.62	57.41	20.23	**14.38**

The bold entires denote the best results among the different optimization algorithms.

proposed constraint handling approach decreases to 20.23 h and 14.38 h, respectively. This demonstrates that the proposed approach can greatly enhance the computational efficiency for metaheuristic algorithms with the elite selection in large-scale structure optimization problems.

6 Conclusions

This study proposes a filter strategy, which predicates on the solution updating rule of the metaheuristic algorithms, to reduce the redundant structural analyses in large-scale structural optimization. The feasible condition and the merits of the filter strategy have been discussed theoretically. Both numerical simulations and two large-scale structural optimization examples have demonstrated the efficiency of the metaheuristic algorithms coupled with the filter strategy. The present study supports the following conclusions:

(1) According to the new solution updating rule, the metaheuristic algorithms can be divided into the two categories: replacement and elitism. The elitism characterizes by only preserving the solutions with better fitness in the iteration process, performs as the foundation of the proposed filter strategy.

(2) The filter strategy not only eliminates the unnecessary constraint violation evaluations in the optimization procedure, but also upholds the search ability of the metaheuristic algorithms and retains the merits of the death penalty approach, for instance, without parameter-tuning and always ensuring the feasibility of the optimal solutions. Besides the death penalty approach, the filter mechanism of the filter strategy is also applicable to other constraint handling techniques, such as the penalty method and the Deb rule, to enhance their computational efficiency while maintaining the original performances.

(3) The mathematical simulations show that the HS occupies the lowest R value, which is always smaller than 0.4 and means about 60% of time savings while the TLBO obtains the largest R value among the five methods, including the CS, FAP, HS, PSO and TLBO. The R value also varies with different variations for the same metaheuristic algorithm. Besides the computational efficiency varies with different algorithms, an algorithm with a lower computational efficiency does not guarantee a higher quality optimal solution.

(4) The 942-bar problem shows that the R value for HS is merely 0.09, which implies the proposed filter strategy leads to over 90% of time savings than the standard death penalty approach. To solve the suspension bridge optimization with the same number of iterations, the penalty

method and the Deb rule coupled with the IPSO cost 55.62 h and 57.41 h, respectively. However, the computational time has reduced to 20.23 h for IPSO and 14.38 h for HS using the proposed method. It demonstrates that the filter strategy can significantly enhance the computational efficiency of the metaheuristic algorithms in structural optimization. The illustrated method can also improve the computational efficiency in the problems with high-cost objective function and low-cost constraint functions by reversing the sequence of the objective function evaluation and the constraint violation check.

Acknowledgements The authors would like to acknowledge the financial support from by the Fundamental Research Funds for the Central Universities (WUT: 2015IVA015) and the National Natural Science Foundation of China (Grant No. 51408249).

Appendix: Constrained Engineering Design Problems

Problem 1: An idealized optimization problem

Objective:

$$\min f(x) = (x_1-10)^3 + (x_2-20)^3$$

Constraints:

$$g_1(x) = (x_1-5)^2 + (x_2-5)^2 + 100 \le 0$$

$$g_2(x) = (x_1-5)^2 + (x_2-5)^2 - 82.81 \le 0$$

where

$$13 \le x_1 \le 100\,,\ 0 \le x_2 \le 100$$

Problem 2: the welded beam design problem

Objective:

$$\min f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

Constraints:

$$g_1(x) = \tau(x) - \tau_{\max} \le 0$$

$$g_2(x) = \sigma(x) - \sigma_{\max} \le 0$$

$$g_3(x) = x_1 - x_4 \le 0$$

$$g_4(x) = 0.1047 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0$$

$$g_5(x) = 0.125 - x_1 \le 0$$

$$g_6(x) = \delta(x) - \delta_{\max} \le 0$$

$$g_7(x) = P - P_C(x) \le 0$$

Where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},\ \tau' = \frac{P}{\sqrt{2}x_1x_2},$$

$$\tau'' = \frac{MR}{J},\ M = P\left(L + \frac{x_2}{2}\right),\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_2}{2}\right)^2\right]\right\},\ \sigma(x) = \frac{6PL}{x_4x_3^2},$$

$$\delta(x) = \frac{4PL^3}{Ex_3^3x_4},\ P_C(x) = \frac{4.013E\sqrt{\frac{x_3^2x_6^4}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$$P = 6000, L = 14, E = 3 \times 10^7,\ \ G = 1.2 \times 10^7,$$

$$\tau_{\max} = 13600, \sigma_{\max} = 30000, \delta_{\max} = 0.25,$$

$$0.1 \le x_1 \le 2.0, 0.1 \le x_2 \le 10.0, 0.1 \le x_3 \le 10.0, 0.1 \le x_4 \le 2.0,$$

Problem 3: the tension/compression spring design problem

Objective:

$$\min f(x) = (x_3 + 2)x_2x_1^2$$

Constraints:

$$g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566\left(x_1^3x_2 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \le 0$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

where

$$0.05 \le x_1 \le 2.0, 0.25 \le x_2 \le 1.3, 2.0 \le x_3 \le 15.0,$$

Problem 4: the speed reducer design problem

Objective:

$$\begin{aligned}\min f(x) = {} & 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ & - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\ & + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)\end{aligned}$$

Constraints:

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$$

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$$

$$g_3(x) = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \le 0$$

$$g_4(x) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0$$

$$g_5(x) = \frac{\sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 1.69 \times 10^7}}{110 x_6^3} - 1 \le 0$$

$$g_6(x) = \frac{\sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 1.575 \times 10^8}}{85 x_7^3} - 1 \le 0$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \le 0$$

$$g_8(x) = \frac{5 x_2}{x_1} - 1 \le 0$$

$$g_9(x) = \frac{x_1}{12 x_2} - 1 \le 0$$

$$g_{10}(x) = \frac{1.56 x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(x) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0$$

where

$$2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 7.3 \le x_4 \le 8.3,$$
$$7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5,$$

Problem 5: the three-bar truss design problem

Objective:

$$\min f(x) = \left(2\sqrt{2} x_1 + x_2\right) l$$

Constraints:

$$g_1(x) = \frac{\sqrt{2} x_1 + x_2}{\sqrt{2} x_1^2 + 2 x_1 x_2} p - \sigma \le 0$$

$$g_2(x) = \frac{x_2}{\sqrt{2} x_1^2 + 2 x_1 x_2} p - \sigma \le 0$$

$$g_3(x) = \frac{1}{\sqrt{2} x_2 + x_1} p - \sigma \le 0$$

where

$$0 < x_1 \le 1.0, 0 < x_2 \le 1, l = 100, p = 2.0, \sigma = 2.0,$$

References

Abdel-Raouf O, Abdel-Baset M (2014) A new hybrid flower pollination algorithm for solving constrained global optimization problems. Int J App Oper Res-An Open Access Journal 4:1–13

Agarwal P, Raich AM (2006) Design and optimization of steel trusses using genetic algorithms, parallel computing, and human-computer interaction. Struct Eng Mech 23:325–337

Ahrari A, Deb K (2016) An improved fully stressed design evolution strategy for layout optimization of truss structures. Comput Struct 164:127–144

Arora J, Wang Q (2005) Review of formulations for structural and mechanical system optimization. Struct Multidiscip Optim 30:251–272

Astroza R, Nguyen LT, Nestorović T (2016) Finite element model updating using simulated annealing hybridized with unscented Kalman filter. Comput Struct 177:176–191

Baghlani A, Makiabadi M, Maheri M (2017) Sizing optimization of truss structures by an efficient constraint-handling strategy in TLBO. J Comput Civ Eng 31:04017004

Cao H, Qian X, Chen Z, Zhu H (2017a) Enhanced particle swarm optimization for size and shape optimization of truss structures. Eng Optim:1–18. doi:10.1080/0305215X.2016.1273912

Cao H, Qian X, Chen Z, Zhu H (2017b) Layout and size optimization of suspension bridges based on coupled modelling approach and enhanced particle swarm optimization. Eng Struct 146:170–183

Chen Z, Cao H, Zhu H (2013) An iterative calculation method for suspension bridge's cable system based on exact catenary theory. Balt J Road Bridge Eng 8:196–204

Chen Z, Cao H, Ye K, Zhu H, Li S (2015) Improved particle swarm optimization-based form-finding method for suspension bridge installation analysis. J Comput Civ Eng 29:04014047

Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:311–338

Degertekin S (2012) Improved harmony search algorithms for sizing optimization of truss structures. Comput Struct 92:229–241

Efren M-M (2009) Constraint-handling in Evolutionary Optimization, first ed. Springer, New York, USA

Farshchin M, Camp C, Maniat M (2016) Multi-class teaching–learning-based optimization for truss design with frequency constraints. Eng Struct 106:355–369

Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35

Gandomi AH, Kashani AR, Roke DA, Mousavi M (2015) Optimization of retaining wall design using recent swarm intelligence techniques. Eng Struct 103:72–84

García-Segura T, Yepes V, Alcalá J, Pérez-López E (2015) Hybrid harmony search for sustainable design of post-tensioned concrete box-girder pedestrian bridges. Eng Struct 92:112–122

Hasançebi O (2008) Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. Comput Struct 86:119–132

Hasançebi O, Erbatur F (2002) On efficient use of simulated annealing in complex structural optimization problems. Acta Mech 157:27–50

He S, Wu Q, Wen J, Saunders J, Paton R (2004) A particle swarm optimizer with passive congregation. Biosystems 78:135–147

Jansen PW, Perez RE (2011) Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach. Comput Struct 89:1352–1366

Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol Comput 1:61–70

Jordehi AR (2015) A review on constraint handling strategies in particle swarm optimisation. Neural Comput & Applic 26:1265–1275

Kaveh A, Bakhshpoori T (2016) A new metaheuristic for continuous structural optimization: water evaporation optimization. Struct Multidiscip Optim 54:23–43

Kaveh A, BolandGerami A (2017) Optimal design of large-scale space steel frames using cascade enhanced colliding body optimization. Struct Multidiscip Optim 55:237–256

Kaveh A, Ilchi Ghazaan M (2017) A new hybrid meta-heuristic algorithm for optimal design of large-scale dome structures. Eng Optim:1–18. doi:10.1080/0305215X.2017.1313250

Kaveh A, Mahdavi V (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27

Kaveh A, Bakhshpoori T, Afshari E (2014) An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput Struct 143:40–59

Kazemzadeh Azad S (2017a) Enhanced hybrid metaheuristic algorithms for optimal sizing of steel truss structures with numerous discrete variables. Struct Multidiscip Optim 55:2159–2180

Kazemzadeh Azad S (2017b) Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses. Eng Optim 8:1–17.doi:10.1080/0305215X.2017.1284833

Kazemzadeh Azad S, Hasançebi O (2013) Upper bound strategy for metaheuristic based design optimization of steel frames. Adv Eng Softw 57:19–32

Kazemzadeh AS, Hasançebi O, Azad SK, Erol O (2013) Upper bound strategy in optimum design of truss structures: a big bang-big crunch algorithm based application. Adv Struct Eng 16:1035–1046

Kennedy J (2003) Bare bones particle swarms. In: Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE. IEEE, pp 80–87

Kowalczyk R (1997) Constraint consistent genetic algorithms. In: Evolutionary Computation, 1997., IEEE International Conference on. IEEE, pp 343–348

Le Riche R, Knopf-Lenoir C, Haftka RT (1995) A segregated genetic algorithm for constrained structural optimization. In: Eshelman L. J.; (ed.) Proc. 6-th Int. Conf. on Genetic Algorithms, San Francisco, CA: Morgan Kaufmann Publishers, pp. 558–565.

Li E, Wang H, Ye F (2016) Two-level multi-surrogate assisted optimization method for high dimensional nonlinear problems. Appl Soft Comput 46:26–36

Lute V, Upadhyay A, Singh KK (2009) Computationally efficient analysis of cable-stayed bridge for GA-based optimization. Eng Appl Artif Intell 22:750–758

Maaranen H, Miettinen K, Penttinen A (2007) On initial populations of a genetic algorithm for continuous optimization problems. J Glob Optim 37:405–436

Mezura-Montes E, Coello CAC (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 1:173–194

Nickabadi A, Ebadzadeh MM, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. Appl Soft Comput 11:3658–3670

Ong YS, Nair PB, Keane AJ (2003) Evolutionary optimization of computationally expensive problems via surrogate modeling. AIAA J 41:687–696

Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248

Shan S, Wang GG (2010) Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct Multidiscip Optim 41:219–241

Sheikholeslami R, Khalili BG, Sadollah A, Kim J (2016) Optimization of reinforced concrete retaining walls via hybrid firefly algorithm with upper bound strategy. KSCE J Civ Eng 20:2428–2438

Tang J, Wang W (2015) A Filter-Genetic Algorithm for Constrained Optimization Problems. In: Gao D, Ruan N, Xing W (eds) Advances in Global Optimization. Springer International Publishing, Cham, pp 355–362

Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 85:317–325

Umesha P, Venuraju M, Hartmann D, Leimbach K (2005) Optimal design of truss structures using parallel computing. Struct Multidiscip Optim 29:285–297

Venter G, Haftka R (2010) Constrained particle swarm optimization using a bi-objective formulation. Struct Multidiscip Optim 40:65–76

Venter G, Sobieszczanski-Sobieski J (2003) Particle swarm optimization. AIAA J 41:1583–1589

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82

Wu P, Gao L, Zou D, Li S (2011) An improved particle swarm optimization algorithm for reliability problems. ISA Trans 50:71–81

Yang X-S (2012) Flower pollination algorithm for global optimization. In: Durand-Lose J, Jonoska N (eds) Unconventional computation and natural computation. vol 7445. Lecture notes in Computer Science, Springer, Berlin, pp 240–249

Yu K, Wang X, Wang Z (2016) An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. Journal of Intelligent Manufacturing 27:831-843 doi:10.1007/s10845-014-0918-3