Research Article

Accelerated Particle Swarm for Optimum Design of Frame Structures

S. Talatahari,1 E. Khalili,2 and S. M. Alavizadeh3

1 Marand Faculty of Engineering, University of Tabriz, Tabriz 51666-14766, Iran
2 Department of Engineering, Islamic Azad University, Ahar Branch, Ahar 54516, Iran
3 Department of Structural Engineering, Islamic Azad University, Shabestar Branch, Shabestar 57168-14758, Iran

Correspondence should be addressed to S. Talatahari; talatahari@tabrizu.ac.ir

Received 29 November 2012; Revised 29 December 2012; Accepted 30 December 2012

Copyright © 2013 S. Talatahari et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accelerated particle swarm optimization (APSO) is developed for finding optimum design of frame structures. APSO shows some extra advantages in convergence for global search. The modifications on standard PSO effectively accelerate the convergence rate of the algorithm and improve the performance of the algorithm in finding better optimum solutions. The performance of the APSO algorithm is also validated by solving two frame structure problems.

1. Introduction

Optimum design of frame structures are inclined to determine suitable sections for elements that fulfill all design requirements while having the lowest possible cost. In this issue, optimization provides engineers with a variety of techniques to deal with these problems [1]. These techniques can be categorized as two general groups: classical methods and metaheuristic approaches [2]. Classical methods are often based on mathematical programming, and many of metaheuristic methods make use of the ideas from nature and do not suffer the discrepancies of mathematical programming [3–8].

Particle swarm optimization (PSO), one of meta-heuristic algorithms, is based on the simulation of the social behavior of bird flocking and fish schooling. PSO is the most successful swarm intelligence inspired optimization algorithms. However, the local search capability of PSO is poor [9], since premature convergence occurs often. In order to overcome these disadvantages of PSO, many improvements have been proposed. Shi and Eberhart [10] introduced a fuzzy system to adapt the inertia weight for three benchmark test functions. Liu et al. [11] proposed center particle swarm optimization by adding a center particle into PSO to improve the performance. An improved quantum-behaved PSO was proposed by Xi et al. [12]. Jiao et al. [13] proposed the dynamic inertia weight PSO, by defining a dynamic inertia weight to decrease the inertia factor in the velocity update equation of the original PSO. Yang et al. [14] proposed another dynamic inertia weight to modify the velocity update formula in a method called modified particle swarm optimization with dynamic adaptation.

A number of studies have applied the PSO and improved it to be used in the field of structural engineering [15–21]. In this study, we developed an improved PSO, so-called accelerated particle swarm optimization (APSO) [22], to find optimum design of frame structures. The resulted method is then tested by some numerical examples to estimate its potential for solving structural optimization problems.

2. Statement of Structural Optimization Problem

Optimum design of structures includes finding optimum sections for members that minimizes the structural weight W.
This minimum design should also satisfy inequality constraints that limit design variables and structural responses. Thus, the optimal design of a structure is formulated as [23]

\[
\text{minimize } W \left( x \right) = \sum_{i=1}^{n} y_i \cdot A_i \cdot l_i,
\]

subject to: \( g_{\text{min}} \leq g_i \left( x \right) \leq g_{\text{max}}, \quad i = 1, 2, 3, \ldots, m, \)

where \( W \left( x \right) \) is the weight of the structure; \( n \) and \( m \) are the number of members making up the structure and the number of total constraints, respectively; max and min denote upper and lower bounds, respectively; \( g_i \left( x \right) \) denotes the constraints considered for the structure containing interaction constraints as well as the lateral and interstory displacements, as follows.

The maximum lateral displacement:

\[
g_{\text{max}} = \frac{\Delta_T}{H} - R \geq 0.
\]

The interstory displacements:

\[
g_j^d = \frac{d_j}{h_j} - R_i \geq 0, \quad j = 1, 2, \ldots, ns,
\]

where \( \Delta_T \) is the maximum lateral displacement; \( H \) is the height of the frame structure; \( R \) is the maximum drift index; \( d_j \) is the inter-story drift; \( h_j \) is the story height of the \( j \)th floor; \( ns \) is the total number of stories; \( R_i \) is the inter-story drift index permitted by the code of the practice.

LRFD interaction formula constraints (AISC 2001 [24, Equation H1-la,b]):

\[
g_i^a = \frac{P_u}{2\phi_nP_n} + \left( \frac{M_{nx}}{\phi_bM_{nx}} + \frac{M_{ny}}{\phi_bM_{ny}} \right) - 1 \geq 0 \quad \text{for } \frac{P_u}{\phi_nP_n} < 0.2,
\]

\[
g_i^b = \frac{P_u}{\phi_bP_n} + 8 \left( \frac{M_{nx}}{\phi_bM_{nx}} + \frac{M_{ny}}{\phi_bM_{ny}} \right) - 1 \geq 0 \quad \text{for } \frac{P_u}{\phi_bP_n} \geq 0.2,
\]

where \( P_u \) is the required strength (tension or compression); \( P_n \) is the nominal axial strength (tension or compression); \( \phi_c \) is the resistance factor (\( \phi_c = 0.9 \) for tension, \( \phi_c = 0.85 \) for compression); \( M_{nx} \) and \( M_{ny} \) are the required flexural strengths in the \( x \) and \( y \) directions, respectively; \( M_{nx} \) and \( M_{ny} \) are the nominal flexural strengths in the \( x \) and \( y \) directions (for two-dimensional structures, \( M_{ny} = 0 \)); \( \phi_b \) is the flexural resistance reduction factor (\( \phi_b = 0.90 \)).

For the proposed method, it is essential to transform the constrained optimization problem to an unconstrained one. A detailed review of some constraint-handling approaches is presented in [25]. In this study, a modified penalty function method is utilized for handling the design constraints which is calculated using the following formulas [2]:

\[
g_i \leq a \implies \Phi_g^{(i)} = 0,
\]

\[
g_i > a \implies \Phi_g^{(i)} = g_i,
\]

The objective function that determines the fitness of each particle is defined as

\[
\text{Mer}^k = \varepsilon_1 \cdot W^k + \varepsilon_2 \cdot (\sum \Phi_g^{(i)})^\gamma,
\]

where Mer is the merit function to be minimized; \( \varepsilon_1, \varepsilon_2 \), and \( \gamma \) are the coefficients of merit function; \( \Phi_g^{(i)} \) denotes the summation of penalties. In this study, \( \varepsilon_1 \) and \( \varepsilon_2 \) are set to 1 and \( W \) (the weight of structure), respectively, while the value of \( \gamma \) is taken as 0.85 in order to achieve a feasible solution [26]. Before calculating \( \Phi_g^{(i)} \), we first determine the weight of the structures generated by the particles, and if it becomes smaller than the so far best solution, then \( \Phi_g^{(i)} \) will be calculated; otherwise the structural analysis does not perform. This methodology will decrease the required computational costs, considerably.

3. Canonical Particle Swarm Optimization (PSO)

The PSO algorithm, inspired by social behavior simulation [27,28], is a population-based optimization algorithm which involves a number of particles that move through the search space, and their positions are updated based on the best positions of individual particles (called \( x_i^\ast \)) and the best of the swarm (called \( g^\ast \)) in each iteration. This matter is shown mathematically as the following equations:

\[
\begin{align*}
\dot{v}_i^{t+1} &= w \cdot \dot{v}_i^t + \alpha \cdot \text{rand}_1 \left( x_i^t - x_i^\ast \right) + \beta \cdot \text{rand}_2 \left( g_i^\ast - x_i^t \right), \\
x_i^{t+1} &= x_i^t + v_i^{t+1},
\end{align*}
\]

where \( x_i \) and \( v_i \) represent the current position and the velocity of the \( i \)th particle, respectively; \( \text{rand}_1 \) and \( \text{rand}_2 \) represent random numbers between 0 and 1; \( x_i^\ast \) is the best position visited by each particle itself; \( g^\ast \) corresponds to the global best position in the swarm up to iteration \( k \); \( \alpha \) and \( \beta \) represent cognitive and social parameters, respectively. According to Kennedy and Eberhart [27], these two constants are set to 2 in order to make the average velocity change coefficient close to one. \( W \) is a weighting factor (inertia weight) which controls the trade-off between the global exploration and the local exploitation abilities of the flying particles. A larger inertia weight makes the global exploration easier, while a smaller inertia weight tends to facilitate local exploitation. The inertia weight can be reduced linearly from 0.9 to 0.4 during the optimization process [29].

4. Accelerated Particle Swarm Optimization

The standard PSO uses both the current global best \( g^\ast \) and the individual best \( x^\ast \). The reason of using the individual best is primarily to increase the diversity in the quality solutions; however, this diversity can be simulated using some randomness. Subsequently, there is no compelling reason for using the individual best, unless the optimization problem of interest is highly nonlinear and multimodal [22].
A simplified version which could accelerate the convergence of the algorithm is to use the global best only. Thus, in the APSO [22], the velocity vector is generated by a simpler formula as
\[ v_i^{t+1} = v_i^t + \alpha \cdot \text{randn}(t) + \beta \cdot (g^* - x_i^t), \] (9)
where randn is drawn from \( N(0,1) \) to replace the second term. The update of the position is simply like (8). In order to increase the convergence even further, we can also write the update of the location in a single step, as
\[ x_i^{t+1} = (1 - \beta) x_i^t + \beta g^* + \alpha r. \] (10)

This simpler version will give the same order of convergence [30]. Typically, \( \alpha = 0.1 \cdot L - 0.5L \), where \( L \) is the scale of each variable, while \( \beta = 0.2 - 0.7 \) is sufficient for most applications. It is worth pointing out that the velocity does not appear in (10), and there is no need to deal with initialization of velocity vectors. Therefore, the APSO is much simpler. Comparing with many PSO variants, the APSO uses only two parameters, and the mechanism is simple to understand. A further improvement to the accelerated PSO is to reduce the randomness as iterations proceed. This means that we can use a monotonically decreasing function. In our implementation, we use [30]
\[ \alpha = 0.7^t, \] (11)
where \( t \in [0,t_{\max}] \) and \( t_{\max} \) is the maximum number of iterations.

5. Numerical Examples

This section presents the numerical examples to evaluate the capability of the new algorithm in finding the optimal design of the steel structures. The final results are compared to the solutions of other methods to show the efficiency of the present approach. The proposed algorithm is coded in Matlab, and structures are analyzed using the direct stiffness
method. The steel members used for the design consist of 267 W-shaped sections from the AISC database.

5.1. 1-Bay 8-Story Frame. Figure 1 shows the configuration of the 1-bay 8-story framed structure and applied loads. Several researchers have developed design procedures for this frame; Camp et al. [31] used a genetic algorithm, Kaveh and Shojaee [32] utilized ACO, and Kaveh and Talatahari [26] applied an improved ACO to solve this problem.

The APSO algorithm found the optimal weight of the one-bay eight-story frame to be 30.91 kN which is the best one compared to the other method. Table 1 lists the optimal values of the eight design variables obtained by this research and compares them with other results.

5.2. Design of a 3-Bay 15-Story Frame. The configuration and applied loads of a 3-bay 15-story frame structure is shown in Figure 2. The sway of the top story is limited to 23.5 cm (9.25 in). The material has a modulus of elasticity equal to $E = 200$ GPa and a yield stress of $F_y = 248.2$ MPa.

The effective length factors of the members are calculated as $K_x \geq 0$ for a sway-permitted frame, and the out-of-plane effective length factor is specified as $K_y = 1.0$. Each column is considered as non-braced along its length, and the non-braced length for each beam member is specified as one-fifth of the span length.

The optimum design of the frame obtained by using APSO has the minimum weight of 411.50 kN. The optimum designs for PSO [18], HBB-BC [33], and ICA [34] had the weights of 496.68 kN, 434.54 kN, and 417.46 kN, respectively. Table 2 summarizes the optimal results for these different algorithms. Clearly, it can be seen that the present algorithm can find the better design. Figure 3 provides the convergence history for this example obtained by the APSO.

6. Conclusions

The APSO algorithm, as an improved meta-heuristic algorithm, is developed to solve frame structural optimization problems. Optimization software based on the APSO algorithm was coded in the Matlab using object-oriented technology. A methodology to handle the constraints is also developed in a way that we first determine the weight of the structures generated by the particles, and if they become smaller than the so far best solution, then the structural
analyses are performed. Two test problems were studied using the optimization program to show the efficiency of the algorithm. The comparison of the results of the new algorithm with those of other algorithms shows that the APSO algorithm provides results as good as or better than other algorithms and can be used effectively for solving engineering problems.

References


Submit your manuscripts at
http://www.hindawi.com