Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 416941, 10 pages
http://dx.doi.org/10.1155/2013/416941

Research Article

Combined Data with Particle Swarm Optimization for Structural Damage Detection

Fei Kang,[1] Junjie Li,[1] and Sheng Liu[2]

[1] *Faculty of Infrastructure Engineering, Dalian University of Technology, Dalian 116024, China*
[2] *College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China*

Correspondence should be addressed to Fei Kang; kangfei@dlut.edu.cn

Received 23 October 2012; Accepted 11 December 2012

Academic Editor: Sheng-yong Chen

This paper proposes a damage detection method based on combined data of static and modal tests using particle swarm optimization (PSO). To improve the performance of PSO, some immune properties such as selection, receptor editing, and vaccination are introduced into the basic PSO and an improved PSO algorithm is formed. Simulations on three benchmark functions show that the new algorithm performs better than PSO. The efficiency of the proposed damage detection method is tested on a clamped beam, and the results demonstrate that it is more efficient than PSO, differential evolution, and an adaptive real-parameter simulated annealing genetic algorithm.

1. Introduction

A structural system or mechanical component continuously accumulates damage during their service life. The presence of damages may reduce the performance of a structure, such as decreasing the service life, or even progressing to catastrophic failure. In recent years, the damage assessment of structures has drawn wide attention from various engineering fields. Structural damage usually causes a decrease in structural stiffness, which produces changes in the vibration characteristics and static displacements of the structure. Major damage detection approaches can be clarified into three major categories, the static identification methods using static test data [1, 2], the dynamic identification methods using vibration test data [3, 4], and the combination methods adopt both vibration data and static data [5]. The vibration data of a structure include natural frequencies, mode shapes, frequency response functions, and modal curvatures. Static responses generally include static displacements and static strain.

The usual damage detection methods minimize an objective function, which is defined in terms of the discrepancies between the vibration data or static data identified by testing and those which are computed from the analytical model. Traditional damage detection methods have some disadvantages such as the damage position and damage extent cannot be detected simultaneously and are not so efficient in detecting the damage extent and adopt local optimization methods which usually lead to a local minimum only.

In recent years, evolutionary algorithms have been extensively applied to damage detection and the related optimal sensor placement problems [6–8]. For example, genetic algorithm (GA) as a stochastic search algorithm using static data [1, 2] or vibration data [3, 4] has been applied to damage detection. Some combination methods using both dynamic data and static data can be seen in [5, 9]. Damage detection methods of GA combining with simulated annealing [10] and artificial neural networks [11] were also proposed. In order to get more accurate structural parameters and structural responses for damage detection, parthenogenetic algorithm approach was proposed for optimal sensor placement [12, 13]. However, the disadvantage of GA is it exhibits a distinguished drop in efficiency as the number of unknown parameters to be identified is more than two [14] and premature convergence is likely to happen [15–19].

Addition to GA, some new techniques are proposed for engineering optimization problems, such as particle swarm optimization (PSO) [20–22], differential evolution [23, 24], artificial bee colony (ABC) [25–28], and so on. Among these optimization algorithms, PSO has obtained the most extensive attention and applications. Although PSO shares many similarities with genetic algorithms, the standard PSO does not use genetic operators such as crossover and mutation. PSO is applied to structural damage detection problems. Meanwhile, to improve the convergence speed and accuracy, some immune mechanisms, such as selection, receptor editing, and vaccination, are incorporated into PSO and propose an immunity enhanced particle swarm optimization (IEPSO). The algorithm is applied to the benchmark function optimization and damage detection problems using combined data. Results show that the performance of PSO is improved, because the convergence speed is accelerated and the default, which is easily getting entrapped in a local optimum when solving complex multimodal problems, is meliorated. Compared with PSO, DE and an adaptive real-parameter simulated annealing genetic algorithm (ARSAGA), IEPSO is the most efficient optimization method for damage identification.

This paper is organized as follows. Section 2 presents the mathematical model for structural damage detection. Section 3 introduces the original PSO, several immune mechanisms and then IEPSO is proposed. Sections 4 and 5 shows the performance of IEPSO on function optimization and damage detection, respectively. Finally, conclusions are given in Section 6.

2. Mathematical Model for Structural Damage Detection Using Combined Data

The analytical static model for an intact structure in the finite-element formulation is

$$\mathbf{K}\mathbf{u} = \mathbf{p}. \tag{1}$$

The characteristic evaluation of a dynamic undamaged structure can be expressed as

$$\mathbf{K}\boldsymbol{\phi}_i - \omega_i^2 \mathbf{M}\boldsymbol{\phi}_i = 0. \tag{2}$$

In (1), $\mathbf{K}$ is the structural stiffness matrix, $\mathbf{u}$ is the displacement vector, and $\mathbf{p}$ is the load vector. In (2), $\mathbf{M}$ is the mass matrix, ω_i is the ith natural frequency, and $\boldsymbol{\phi}_i$ is the corresponding mode shape.

In the context of discretized finite elements, structural damage can be represented by a decrease in the stiffness of the individual elements as

$$\mathbf{k}_d^e = \alpha_e \mathbf{k}^e, \tag{3}$$

where $\mathbf{k}_d^e$ is the eth element stiffness matrix of the damaged structure, α_e is the stiffness damage factor (SDF) [13] of the eth element and is a value between 0 and 1. α_e is 1 with no damage and zero with complete damage in the eth element, respectively.

A uniform damage for the whole element has been assumed in (3). The SDF allows estimating not only the damage severity but also the damage location since the damage identification is carried out at the element level.

In general, a small number of sensors will result in nonunique solutions. The sparsity of measurement can be overcome by increasing the number of loading conditions instead of increasing the number of sensors [29]; So we adopt several load cases which have been used in [2, 10]. To combine the static responses (displacements) and dynamic responses (natural frequencies), the final objective function can be written as

$$f = \sum_{i=1}^{\mathrm{NL}} \sum_{j=1}^{\mathrm{NP}} \left(w_{ij} \left(\frac{u_{m,ij} - u_{\mathrm{th},ij}}{u_{m,ij}} \right) \right)^2 + \sum_{k=1}^{\mathrm{NF}} \left(w_k \left(\frac{\omega_{m,k} - \omega_{\mathrm{th},k}}{\omega_{\mathrm{th},k}} \right) \right)^2, \tag{4}$$

where w_{ij} is a weight factor of the output error at the jth point due to the ith load case. NP is the number of displacement points considered and NL is the number of load cases. $u_{m,ij}$ is the measured displacement of the jth point due to the ith load case, and $u_{\mathrm{th},ij}$ is the theoretically computed counterpart. $\omega_{m,k}$ is the kth measured natural frequency, and $\omega_{\mathrm{th},k}$ is the kth theoretically computed natural frequency. w_k is a weighting factor for the output error of the kth natural frequency. NF is the number of natural frequencies considered.

The first part is defined by the normalized difference between the measured and theoretical computed displacements. The second part is defined by the normalized difference between the measured and theoretically computed natural frequencies. Differences between displacements and natural frequencies are normalized to get a better representation of the relative change in response.

To include the uncertainty in the measured data and to study the sensitivity of IEPSO to noise, uniformly distributed random noise [2] is added to measurements in the simulated tests as

$$\begin{aligned} u'_{m,ij} &= u_{m,ij}(1 + \beta e) \\ \omega'_{m,k} &= \omega_{m,k}(1 + \gamma e), \end{aligned} \tag{5}$$

where $u'_{m,ij}$ is the measured displacement of the jth point due to the ith load case with noise; $\omega'_{m,k}$ is the kth measured natural frequency with noise; β and γ are the noise amplitudes; e is a uniformly distributed random variable in the range $[-1, 1]$.

3. Immunity-Enhanced Particle Swarm Optimization

3.1. Particle Swarm Optimization. Inspired by a model of social interactions between independent animals seeking for food, PSO utilizes swarm intelligence to achieve the goal of optimization. Instead of using genetic operators to manipulate the individuals, each individual in PSO flies in the search space with a velocity which is dynamically adjusted according to its own flying experience and flying experience of its companions [20, 21]. Each individual is treated as a volumeless particle (a point) in the D-dimensional search space. The ith particle is represented as $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ in the D-dimensional space, where $x_{id} \in [l_d, u_d]$, $d \in [1, D]$. The best previous position of the ith particle is recorded as $\mathbf{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The index of the best particle among all the particles in the population is represented by the symbol g. The velocity for particle i is represented as $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, which is clamped to a maximum velocity $V_{\max}$. In each time step t, the particles are manipulated according to the following equations:

$$v_{id} = v_{id} + r_1 c_1 (p_{id} - x_{id}) + r_2 c_2 (p_{gd} - x_{id}) \tag{6}$$

$$x_{id} = x_{id} + v_{id}, \tag{7}$$

where c_1 and c_2 are two positive constants, which control how far a particle will move in a single iteration. r_1 and r_2 are random values in the range $[0, 1]$.

Shi and Eberhart [21] later introduced inertia term w term by modifying (6) to

$$v_{id} = w v_{id} + r_1 c_1 (p_{id} - x_{id}) + r_2 c_2 (p_{gd} - x_{id}). \tag{8}$$

They proposed that suitable selection of w will provide a balance between global and local explorations, thus requiring fewer iterations on average to find a sufficiently optimal solution. As originally developed, w often decrease linearly from about 0.9 to 0.4 according to the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}} \times t, \tag{9}$$

where $w_{\max}$ and $w_{\min}$ are the initial weight and final weight, respectively, $t_{\max}$ is the maximum number of allowable generations, and t the current generation number.

An explicit maximum velocity or a constriction factor is usually utilized in PSO algorithms to control the exploration abilities of particles; however, it cannot prevent them from going outside the allowable solution space and hence produce invalid solutions. Four types of boundary conditions, namely, absorbing, reflecting, invisible, and damping, have been reported in the literature [30]. The optimization of a 2D array antenna [31] shows that the damping boundary condition offers more robust and consistent performance.

As for damage detection problem, generally only few elements are damaged and most elements are still intact. The parameter values of these intact elements are always in the upper bound [9]. In order to avoid oscillating around the upper boundary and to quickly return to the feasible region around the lower boundary, a combined boundary condition is used. This boundary is also convenient to the comparison of different algorithms. The upper bound adopts damping bound [30] and the low bound adopts the bound described in [24], which can be described as follows:

$$\begin{gathered} x_{id} = \begin{cases} l_d + (l_d - x_{id}) & x_{id} < l_d \\ u_d & x_{id} > u_d \end{cases} \\ v_{id} = -\text{rand}\,() \cdot v_{id} \quad x_{id} < l_d \text{ or } x_{id} > u_d, \end{gathered} \tag{10}$$

where l_d and u_d are the lower and upper bounds of dimension d, respectively, rand( ) is a random number in the range $[0, 1]$.

3.2. Artificial Immune System. AIS can be defined as abstract computational systems inspired by theoretical immunology and observed immune functions, principles, and models, applied to solve problems [32]. In AIS, an antigen is used to represent the programming problem to be addressed. An antibody is set (a repertoire), wherein each member represents a candidate solution. Affinity is used to represent the fit of an antibody (a solution candidate) to the antigen (the problem) [33]. Several immune properties, selection, receptor editing, and vaccination, can be adopted to improve the performance of PSO.

3.2.1. Selection. The antibodies present in a population set contain much information regarding the solution of the problem. Based on their affinity, the antibodies are selected to proliferate and produce clones. Traditionally, deterministic selection rule is adopted to select better antibodies for proliferation. However, deterministic selection rule selects only the best antibodies for proliferation, and that may lead to the premature convergence of the algorithm [34].

To overcome this difficulty, rank-based fitness assignment and roulette wheel selection rule is adopted. The fitness of an individual can be calculated as

$$\text{fitpos} = 2 - \text{sp} + \frac{2(\text{sp} - 1)(\text{pos} - 1)}{(N - 1)}, \tag{11}$$

where N is the population size, pos is the order number of the individual in the whole population, sp is the selection pressure, $\text{sp} \in [1.0, 2.0]$, fitpos is the rank-based fitness of the individual.

3.2.2. Receptor Editing. In AIS, receptor editing allows an antibody to take large steps, landing in a locale where the affinity might be lower. However, occasionally the leap will lead to an antibody on the side where the region is more promising. From this locale, point mutations followed by selection can drive the antibody to reach the global optimum. Receptor editing offers the ability to escape from local optima [32]. In order to simulate receptor editing, we introduce non-uniform mutation [35] into PSO.

For each individual $\mathbf{x}_i(t)$ in a population of generation t, create an offspring $\mathbf{x}_i(t+1)$ through a non-uniform mutation as follows: if $\mathbf{x}_i(t) = \{x_{i1}, \ldots, x_{id}, \ldots, x_{iD}\}$ is a solution and the element x_{id} is selected for this mutation, the result is a vector $\mathbf{x}_i(t+1) = \{x_{i1}, \ldots, x'_{id}, \ldots, x_{iD}\}$, where

$$x'_{id} = \begin{cases} x_{id} + \Delta(t, u_d - x_{id}) & \text{if a random } \xi \text{ is } 0 \\ x_{id} - \Delta(t, x_{id} - l_d) & \text{if a random } \xi \text{ is } 1, \end{cases} \tag{12}$$

where l_d and u_d are the lower and upper bounds of the variable x_{id}, respectively. The function $\Delta(t, y)$ returns a value in the range $[0, y]$ such that $\Delta(t, y)$ approaches to zero as t increases. This property allows this operator to search the space uniformly at early stages, and very locally at later stages. We use the following function:

$$\Delta(t, y) = y \cdot \left(1 - r^{(1-t/T)^b}\right), \tag{13}$$

```
Algorithm IEPSO
Begin;
Initialize the population (generate a random population of $N$ solutions);
For $i = 1$ to the maximum number of generations;
    For each individual $j \in N$;
        Evolving individual $j$ according to (7) and (8);
            If (rand () $< p_r$) receptor editing;
            If (rand () $< p_v$) vaccination;
        Evaluate individual $j$;
            If (fit ($j$) $< \mathbf{p}_j$) Update the best position of individual $j$;
            If (fit ($j$) $< \mathbf{p}_g$) Update the best position of the whole population;
    End;
    Selection;
    Calculate $w$ of each individual according to (9);
Next $i$;
End;
```

Algorithm 1: Pseudocodes for IEPSO algorithm.

where r is a uniform random number in the range $[0, 1]$, and b is a system parameter determining the degree of nonuniformity.

3.2.3. Vaccination. Given an individual $\mathbf{x}$, vaccination means modifying the genes on some bits in accordance with a priori knowledge so as to gain a better solution with greater probability [36]. As for real-parameter optimization problems, a vaccination means modifying the elements of some individuals. A vaccine is abstracted from the prior knowledge of the problem. During the actual operation, a detailed analysis is firstly carried out on the problem, simultaneously, as many basic characteristics of the problem as possible ought to be found. Then, the characteristics are abstracted to be a vaccine. A vaccine can also be regarded as estimation on some elements of the optimal individual.

Two types of vaccines are adopted for function optimization and damage detection. For the function optimization problem, the vaccine can be abstracted from the best individuals of the population. If $\mathbf{x}_i(t) = \{x_{i1}, \dots, x_{id}, \dots, x_{iD}\}$ is an individual and the element x_k is selected for vaccination, the result is a vector $\mathbf{x}_i(t + 1) = \{x_{i1}, \dots, x_{id\,\text{best}}, \dots, x_{iD}\}$, where $x_{id\text{best}}$ means the dth element of the best individual. For the damage detection problem, generally only few elements are damaged and most elements are still intact. The parameter values of these intact elements are always in the upper bound. So boundary mutation is adopted as a vaccine for damage detection problem. If $\mathbf{x}_i(t) = \{x_{i1}, \dots, x_{id}, \dots, x_{iD}\}$ is an individual and the element x_{id} is selected for vaccination, the result is a vector: $\mathbf{x}_i(t + 1) = \{x_{i1}, \dots, u_d, \dots, x_{iD}\}$.

3.3. Immunity-Enhanced Particle Swarm Optimization. In IEPSO, a population of particles is sampled randomly in the feasible space. Then the population executes PSO or its variants, including the update of position and velocity. After that, it executes receptor editing operator (nonuniform mutation) according to a certain probability p_r, and vaccination operator according to probability p_v. The new generation is obtained by the selection operator after the flying of particles and two immune operators (receptor editing and vaccination). Pseudocodes for IEPSO are shown in Algorithm 1.

In Algorithm 1, p_r is the receptor editing probability; p_v is the vaccination probability; $\mathbf{p}_j$ is the best position of individual j; $\mathbf{p}_g$ is the best position of the whole population; fit(j) is the fitness of individual j.

In the new algorithm, selection and vaccination can improve the convergence speed and receptor editing, helping the algorithm to avoid premature convergence.

4. Tests on Benchmark Functions

Three nonlinear benchmark functions that are commonly used in literatures are adopted. Their formulas and variable ranges are shown in Table 1.

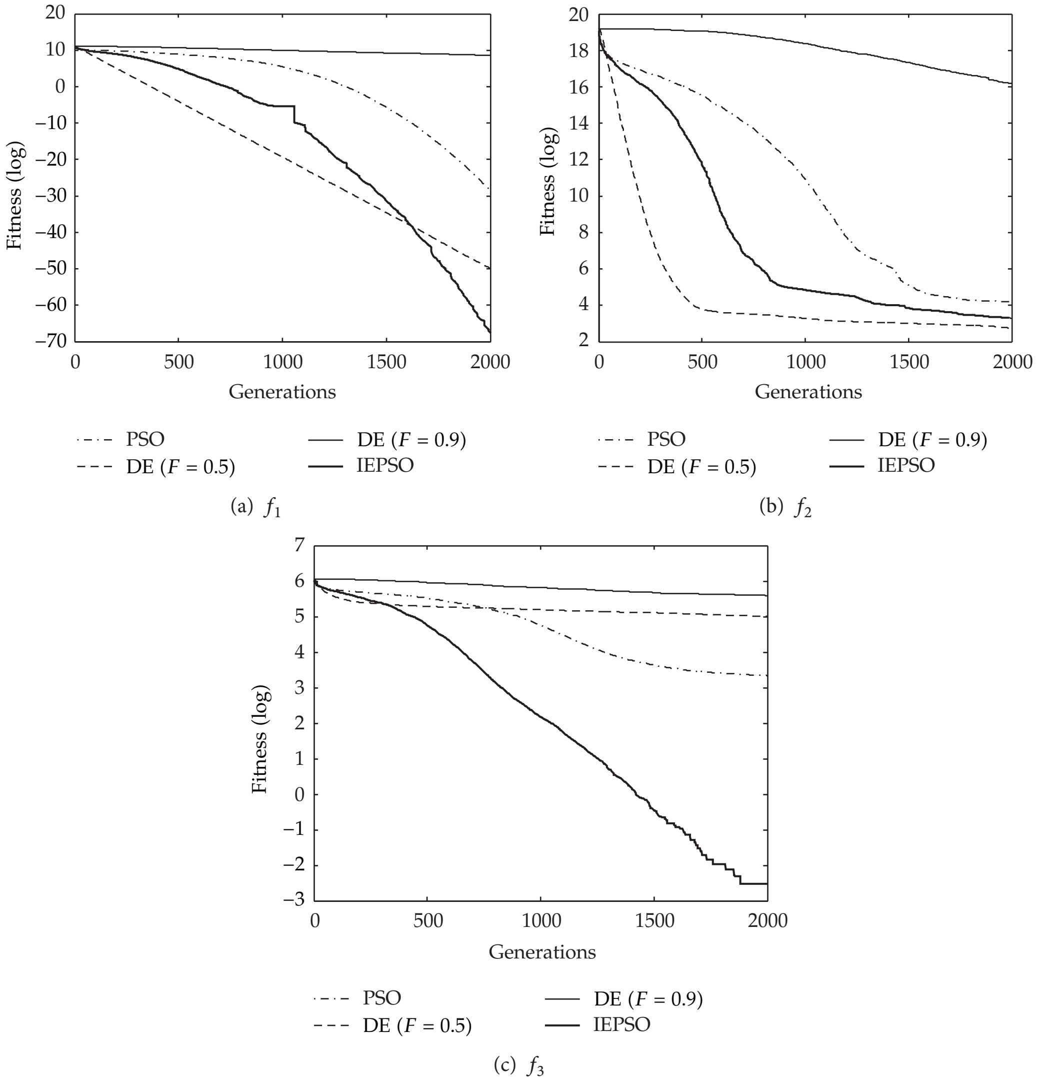
To evaluate the performance of the proposed IEPSO, the basic PSO and DE are used for comparisons. The version of DE used in this paper is known as DE/rand/1/bin, or "classic DE" [23].

The parameters used for IEPSO and PSO are recommended in [21, 22]. The parameter w used is recommended by Shi and Eberhart [21] with a linearly decreasing, which changes from 0.9 to 0.4 according to (9). The maximum velocity $V_{\max}$ and minimum velocity $V_{\min}$ are set at half value of the upper bound and lower bound, respectively. The acceleration constants c_1 and c_2 are both 2.0. For DE algorithm, the parameters are recommended in [23, 24]. The control parameters are $F = 0.9$ or 0.5 and $\text{Cr} = 0.9$. The other parameters for IEPSO are listed in Table 2. The generation number and population size of PSO and DE are the same as IEPSO.

The three algorithms are executed in 50 independent runs. The mean fitness values of the best individual found during the 50 runs for the three functions are listed in Table 3. Evolutionary processes for three functions are shown in Figure 1. From Table 3 and Figure 1, it can be seen that IEPSO outperforms PSO and DE in most cases. By introducing

Table 1: Benchmark test functions.

Name	Formula	n	Range
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$	30	$[-30, 30]^n$
Rastrigin	$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$[-5.12, 5.12]^n$



(a) f_1

(b) f_2

(c) f_3

Figure 1: Convergence processes of mean best fitness for three benchmark functions.

selection operator and vaccination operator, the convergence speed is accelerated and by introducing receptor editing operator, the ability of escaping from a local optimum is enhanced.

5. Damage Detection Examples

A clamped beam adopted from Wang et al. [9] is used to verify the efficiency of IEPSO on damage identification. The initial geometry of the beam is shown in Figure 2. The total numbers of elements and nodes are 20 and 21, respectively. There are two applied load cases. The first load case $P_1 = 50$ N and acts at node 11, and the second load case $P_2 = 50$ N and acts at node six. The material properties are $E = 70.0$ GPa, $\rho = 2.70 \times 10^3$ kg/m^3. In the following tests the objective functions include nineteen node displacements in each load case and the first 10 natural frequencies. The weight factor in (13) is chosen as $w_{ij} = w_k = 1000$.

Statistical analysis is a good way to compare different stochastic algorithms. But the damage detection of a structure

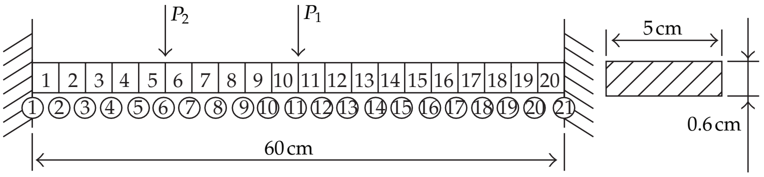
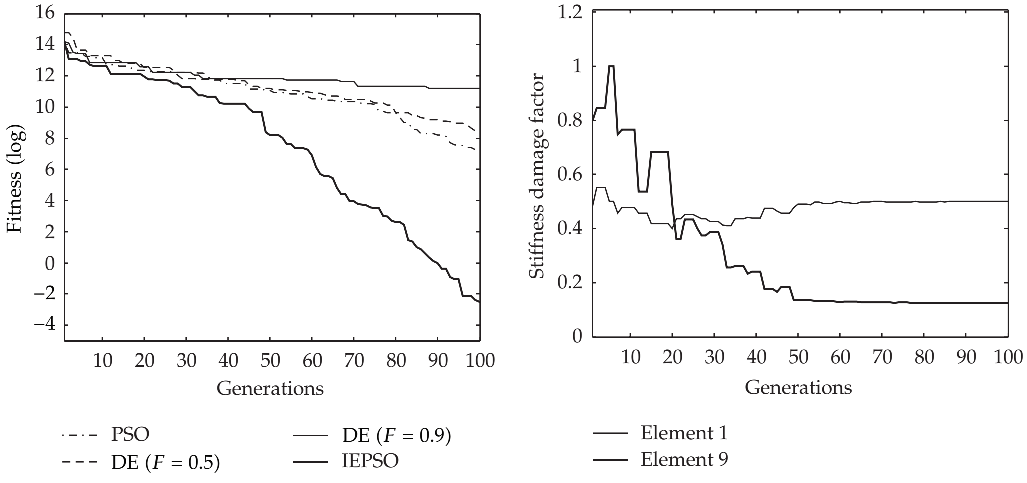


Figure 2: A clamped beam and its cross-section.



(a)

(b)

Figure 3: Convergence processes in case 1: (a) best fitness; (b) SDFs of damaged elements.

Table 2: Parameter values used in IEPSO.

Parameter	Function optimization	Damage detection
Generation number	2000	100
Population size	80	40
p_r	0.1	0.05
p_v	0.1	0.25
SP	1.8	1.8
b	5	5

Table 3: Mean best fitness values of three functions.

Function	DE	PSO	IEPSO
f_1	$2.06e-22$	$4.42e-13$	$\mathbf{4.61}e\mathbf{-30}$
f_2	$\mathbf{1.56}e\mathbf{+01}$	$6.73e+01$	$2.69e+01$
f_3	$1.49e+02$	$2.82e+01$	$\mathbf{7.96}e\mathbf{-02}$

needs a long time. To compare different stochastic algorithms on these time-consuming problems, usually adopt the best results obtained from several runs, and this can avoid chanciness in some extent [37, 38]. In this paper each stochastic algorithm performs three independent runs [14, 26], and the best results obtained from each algorithm are presented as representative ones to yield more clear conclusions because values obtained are not much different.

5.1. Double Damage Occurrence. The first case (case 1) has two damages in which element 1 has 50% reduction in Young's modulus and element 9 has a severe 87.5% reduction in Young's modulus. That means $\alpha_1 = 0.5$ and $\alpha_9 = 0.125$ in this case.

The comparison of logarithmic best fitness values of three algorithms is shown in Figure 3(a). From this figure, we can see that IEPSO converges the fastest and PSO converges faster than DE. The convergence processes of SDF of damaged elements are shown in Figure 3(b). It can be seen that the SDF of element 1 and element 9 quickly converges to 0.49998 and 0.12504 which are quite close to the theoretical values. The comparison of SDF of each element between theoretical values and IEPSO detected values is shown in Figure 4(a). It can be seen that the detected values are quite close to the theoretical values and no extra damage is detected. The comparison of damage detection results using three algorithms after 100 generations is shown in Figure 4(b). It can be seen that IEPSO outperforms PSO and DE.

5.2. Triple Tiny Damage Occurrence. A triple damage occurrence with different damage extents (case 2) is considered. In this example, element 2 has 10% reduction in Young's

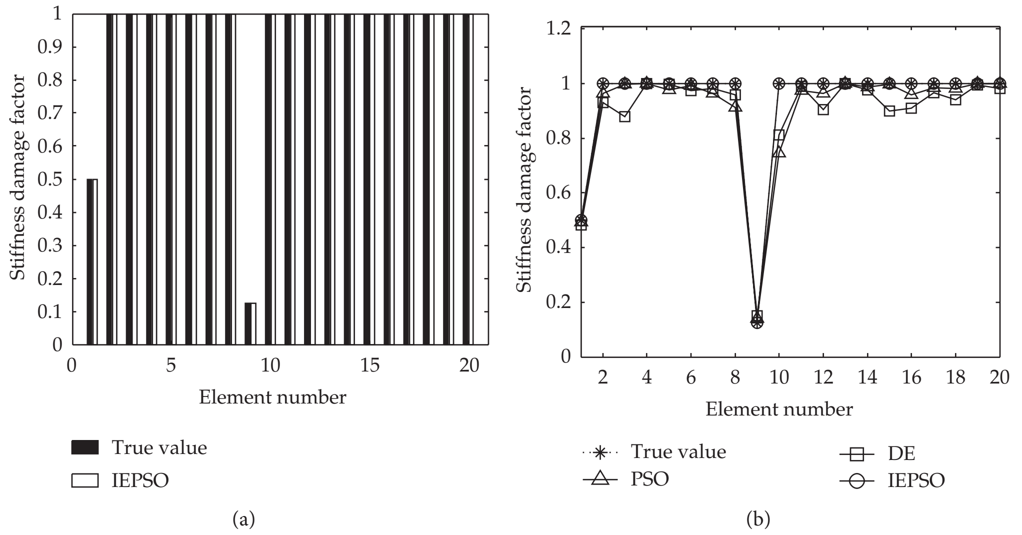


Figure 4: Comparison of damage detection results in case 1: (a) between theoretical value and IEPSO detected value; (b) between different algorithms.

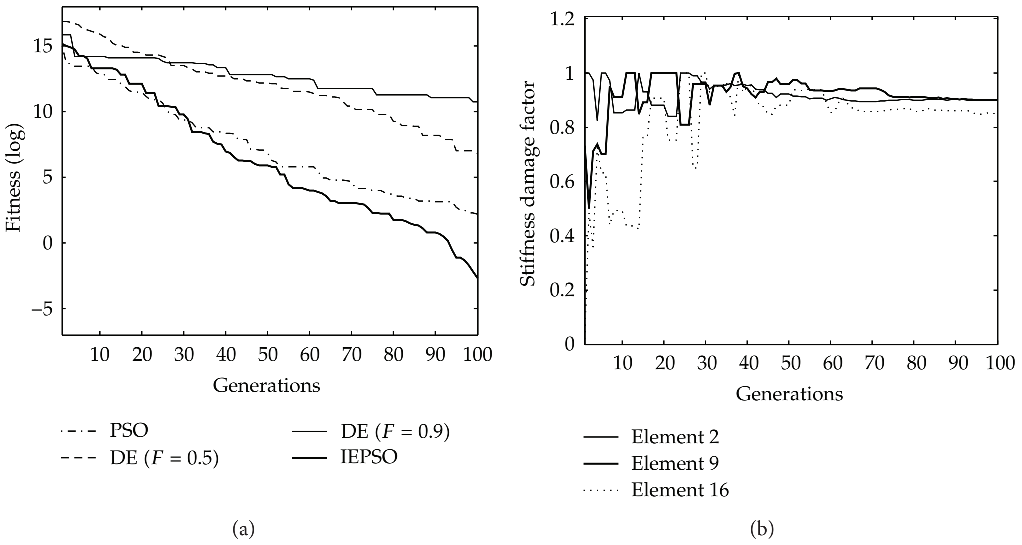


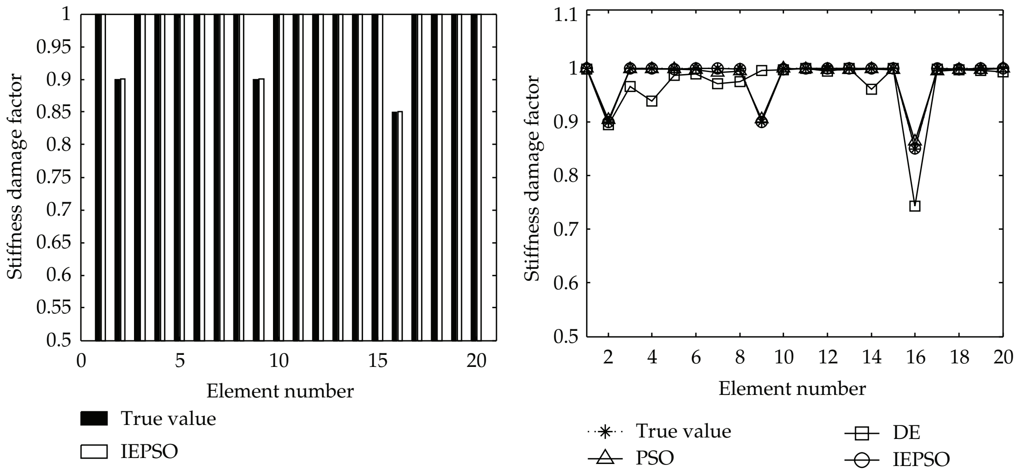
Figure 5: Convergence processes in case 3: (a) best fitness; (b) SDFs of damaged elements.

modulus, element 9 has 10% reduction in Young's modulus, and element 16 has 15% reduction in Young's modulus. This means that $\alpha_2 = 0.9$, $\alpha_9 = 0.9$, and $\alpha_{16} = 0.85$. It is noted that all the three damage extents are very tiny, and the difficulty in damage detection is increased.

The comparison of logarithmic best fitness values of three algorithms is shown in Figure 5(a). From this figure, we can see that IEPSO still converge the fastest and PSO converges faster than DE. The convergence processes of SDF of damaged elements are shown in Figure 5(b). It can be seen that the SDF of element 2, element 9, and element 16 quickly converges to 0.90015, 0.90101, and 0.85091 which are quite close to the theoretical values. The comparison of SDF of each element between theoretical values and IEPSO detected values is shown in Figure 6(a). It can be seen that the detected values are quite close to the theoretical values and no extra damage is detected. The comparison of damage detection results using three algorithms after 100 generations is shown in Figure 6(b). It can be seen that IEPSO still performs better than PSO and DE.

Damage detection performance of IEPSO is also compared with ARSAGA [10] in this case and the result is shown in Table 4. The value in the parenthesis represents the relative error. The damaged elements and the maximum errors of different algorithms are shown in bold. It can be seen that damage detection accuracy of IEPSO is higher than ARSAGA.

5.3. The Effect of Noise. To include the uncertainty in the measured data and to study the sensitivity of IEPSO to



(a) (b)

Figure 6: Comparison of damage detection results in case 3: (a) between theoretical value and IEPSO detected value; (b) between different algorithms.

Table 4: Comparison of damage detection results of case one by ARSAGA and IEPSO.

Element no.	Stiffness damage factors		
	Exact	ARSAGA	IEPSO
1	1.0	**0.963 (−3.7%)**	1.0
2	**0.9**	**0.914**	**0.9**
3	1.0	1.0	1.0
4	1.0	0.984	0.999
5	1.0	0.992	1.0
6	1.0	1.0	1.0
7	1.0	1.0	1.0
8	1.0	0.978	0.999
9	**0.9**	**0.905**	**0.901**
10	1.0	1.0	1.0
11	1.0	1.0	1.0
12	1.0	0.987	1.0
13	1.0	1.0	1.0
14	1.0	0.991	1.0
15	1.0	1.0	1.0
16	**0.85**	**0.852**	**0.851 (0.12%)**
17	1.0	1.0	1.0
18	1.0	1.0	0.999
19	1.0	1.0	1.0
20	1.0	1.0	1.0

noise, different uniformly distributed random noise is added to measured data in the simulation as described (5). It is commonly believed that the relative measurement error of frequency is about 1% [3]. Take case 1 as an example, in the first test 1% noise is added to both static displacements and natural frequencies. In the second test 5% noise is added to static displacements and 2% noise is added to natural frequencies.

The comparison of SDF of each element between theoretical values and IEPSO detected values under the condition 1% noise is shown in Figure 7(a). It can be seen that the detected values are still quite close to the true values and a tiny extra damage is detected in element 19. The comparison of SDF of each element between theoretical values and IEPSO detected values under the condition with 5% noise is shown in Figure 7(b). It can be seen that the two damages can still be detected under this condition, but several tiny extra damages are also detected.

It can be seen from these figures that noise decreases the accuracy of the algorithm. When noise increases, the misidentifications also increase. This may be because that large noise increases the uncertainty in theoretical response data and noise may be taken as damage by the procedure. From this point, the results detected by the procedure are still reasonable.

6. Conclusions

In this paper, a new IEPSO algorithm for structural damage identification is presented. It bases on a particle swarm optimization algorithm and introduces immune properties selection, receptor editing, and vaccination into it. Selection and vaccination can improve the convergence speed and receptor editing helps the algorithm to avoid premature convergence.

Boundary conditions for PSO are also discussed in the paper, and a combined boundary is adopted for damage detection problems. This boundary condition can avoid oscillating around the upper boundary and can quickly return to the feasible region around the lower boundary.

The feasibility of the methodology is first demonstrated through several numerical examples. Three benchmark functions are used to test the performance of IEPSO in complex function optimization problems. Numerical results show that IEPSO performs better than PSO and DE in most cases

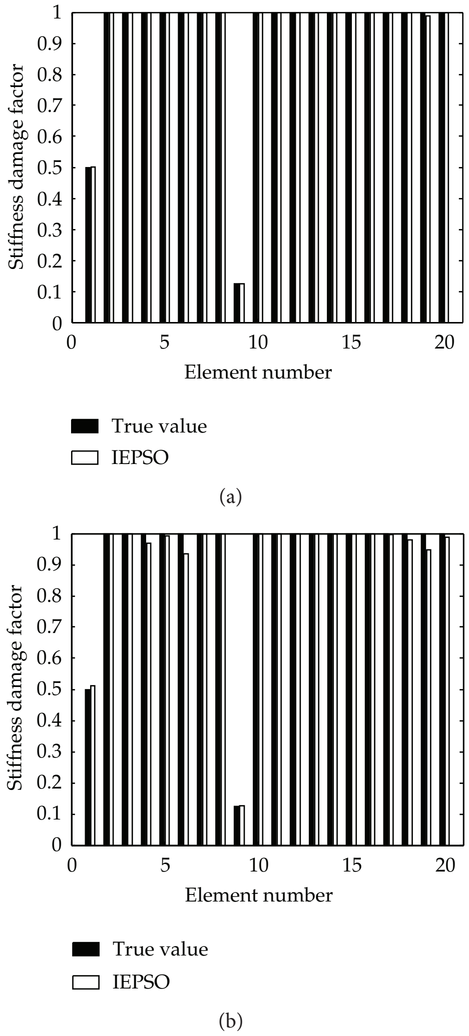


Figure 7: Comparison of damage detection results in case 1 with noise: (a) 1% noise is added to both static displacements and natural frequencies; (b) 5% noise is added to static displacements and 2% noise is added to natural frequencies.

because its convergence speed is the fastest and converges to the best fitness value.

The proposed algorithm is then tested on damage detection problems of a clamped beam. Two damage cases can be detected quickly and accurately by the proposed algorithm when noise is free. Comparing with PSO and DE, IEPSO is more efficient in damage detection problems. The accuracy of IEPSO is also higher than ARSAGA in case 2. When different levels of noise are added, the accuracy of the algorithm is decreased. However, this is still reasonable because noise may be taken as damages in the beam during the damage detection process.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (51109028, 90815024) and the Fundamental Research Funds for the Central Universities (DUT11RC(3)38).

References

[1] F. Bakhtiari-Nejad, A. Rahai, and A. Esfandiari, "A structural damage detection method using static noisy data," *Engineering Structures*, vol. 27, no. 12, pp. 1784–1793, 2005.

[2] B. Kouchmeshky, W. Aquino, J. C. Bongard, and H. Lipson, "Co-evolutionary algorithm for structural damage identification using minimal physical testing," *International Journal for Numerical Methods in Engineering*, vol. 69, no. 5, pp. 1085–1107, 2007.

[3] B. H. Koh and S. J. Dyke, "Structural health monitoring for flexible bridge structures using correlation and sensitivity of modal data," *Computers and Structures*, vol. 85, no. 3-4, pp. 117–130, 2007.

[4] F. Kang, J. J. Li, and Q. Xu, "Damage detection based on improved particle swarm optimization using vibration data," *Applied Soft Computing*, vol. 12, no. 8, pp. 2329–2335, 2012.

[5] B. H. Oh and B. S. Jung, "Structural damage assessment with combined data of static and modal tests," *Journal of Structural Engineering*, vol. 124, no. 8, pp. 956–965, 1998.

[6] S. Y. Chen, Y. J. Zheng, C. Cattani, and W. L. Wang, "Modeling of biological intelligence for SCM system optimization," *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 769702, 10 pages, 2012.

[7] P. Z. Lu, S. Y. Chen, and Y. J. Zheng, "Artificial intelligence in civil engineering," *Mathematical Problems in Engineering*, vol. 2012, Article ID 145974, 22 pages, 2012.

[8] S. Y. Chen and Y. F. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 1, pp. 393–408, 2004.

[9] X. Wang, N. Hu, H. Fukunaga, and Z. H. Yao, "Structural damage identification using static test data and changes in frequencies," *Engineering Structures*, vol. 23, no. 6, pp. 610–621, 2001.

[10] R. S. He and S. F. Hwang, "Damage detection by an adaptive real-parameter simulated annealing genetic algorithm," *Computers and Structures*, vol. 84, no. 31-32, pp. 2231–2243, 2006.

[11] B. Sahoo and D. Maity, "Damage assessment of structures using hybrid neuro-genetic algorithm," *Applied Soft Computing Journal*, vol. 7, no. 1, pp. 89–104, 2007.

[12] F. Kang, J. J. Li, and Q. Xu, "Strategy for optimizing sensor placement related to structural health monitoring," in *Proceeding of International Conference on Health Monitoring of Structure, Material and Environment*, pp. 474–478, Southeast University Press, Nanjing, China, 2007.

[13] F. Kang, J. J. Li, and Q. Xu, "Virus coevolution partheno-genetic algorithms for optimal sensor placement," *Advanced Engineering Informatics*, vol. 22, no. 3, pp. 362–370, 2008.

[14] K. H. Lee, S. W. Baek, and K. W. Kim, "Inverse radiation analysis using repulsive particle swarm optimization algorithm," *International Journal of Heat and Mass Transfer*, vol. 51, no. 11-12, pp. 2772–2783, 2008.

[15] S. Y. Chen, C. Y. Yao, G. Xiao, Y. S. Ying, and W. L. Wang, "Fault detection and prediction of clocks and timers based on computer audition and probabilistic neural networks," in *Computational Intelligence and Bioinspired Systems*, vol. 3512 of *Lecture Notes on Computer Science*, pp. 952–959, 2005.

[16] M. Li, S. C. Lim, and S. Chen, "Exact solution of impulse response to a class of fractional oscillators and its stability,"

Mathematical Problems in Engineering, vol. 2011, Article ID 657839, 9 pages, 2011.

[17] S. C. Lim, C. H. Eab, K. H. Mak, M. Li, and S. Y. Chen, "Solving linear coupled fractional differential equations by direct operational method and some applications," *Mathematical Problems in Engineering*, vol. 2012, Article ID 653939, 28 pages, 2012.

[18] S. Y. Chen, Y. H. Wang, and C. Cattani, "Key issues in modeling of complex 3D structures from video sequences," *Mathematical Problems in Engineering*, vol. 2012, Article ID 856523, 17 pages, 2012.

[19] Z. Teng, J. He, A. J. Degnan et al., "Critical mechanical conditions around neovessels in carotid atherosclerotic plaque may promote intraplaque hemorrhage," *Atherosclerosis*, vol. 223, no. 2, pp. 321–326, 2012.

[20] J. Kennedy and R. C Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[21] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.

[22] Y. Shi and R. C. Eberhart, "Empirical study of particle swarmoptimization," in *Proceeding of the IEEE Congress on Evolutionary Computation (CEC '99)*, pp. 1945–1950, July 1999.

[23] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[24] J. Rönkkönen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, pp. 506–513, Edinburgh, Scotland, September 2005.

[25] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[26] F. Kang, J. J. Li, and Q. Xu, "Structural inverse analysis by hybrid simplex artificial bee colony algorithms," *Computers and Structures*, vol. 87, no. 13-14, pp. 861–870, 2009.

[27] F. Kang, J. Li, and Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions," *Information Sciences*, vol. 181, no. 16, pp. 3508–3531, 2011.

[28] F. Kang, J. Li, and Z. Ma, "An artificial bee colony algorithm for locating the criticalslip surface in slope stability analysis," *Engineering Optimization*. In press.

[29] I. Y. Choi, J. S. Lee, E. Choi, and H. N. Cho, "Development of elastic damage load theorem for damage detection in a statically determinate beam," *Computers and Structures*, vol. 82, no. 29-30, pp. 2483–2492, 2004.

[30] T. Huang and A. S. Mohan, "A hybrid boundary condition for robust particle swarm optimization," *IEEE Antennas and Wireless Propagation Letters*, vol. 4, no. 1, pp. 112–117, 2005.

[31] S. H. Xu and Y. Rahmat-Samii, "Boundary conditions in particle swarm optimization revisited," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 760–765, 2007.

[32] L. N. de Castro and J. I. Timmis, *Artificial Immune Systems: A New Computational Intelligence Paradigm*, Springer, New York, NY, USA, 2002.

[33] P. Musilek, A. Lau, M. Reformat, and L. Wyard-Scott, "Immune programming," *Information Sciences*, vol. 176, no. 8, pp. 972–1002, 2006.

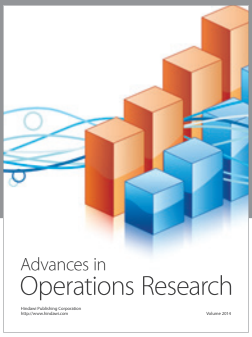
[34] N. Khilwani, A. Prakash, R. Shankar, and M. K. Tiwari, "Fast clonal algorithm," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 1, pp. 106–128, 2008.

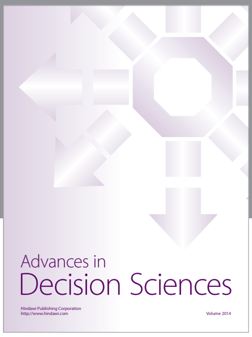
[35] X. C Zhao, X. S. Gao, and Z. C. Hu, "Evolutionary programming based on non-uniform mutation," *Applied Mathematics and Computation*, vol. 192, no. 1, pp. 1–11, 2007.

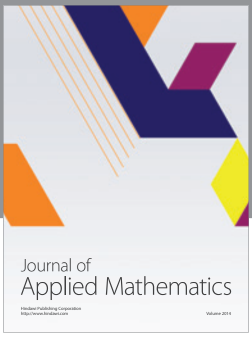
[36] L. C Jiao and L. Wang, "A novel genetic algorithm based on immunity," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 30, no. 5, pp. 552–561, 2000.

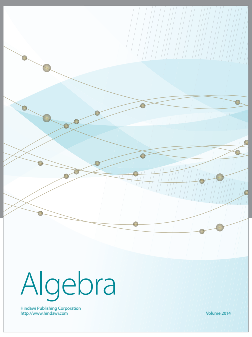
[37] T. Yi, H. Li, and X. Zhang, "Sensor placement on Canton Tower for health monitoring using asynchronous-climbing monkey algorithm," *Smart Materials and Structures*, vol. 21, no. 12, pp. 1–12, 2012.

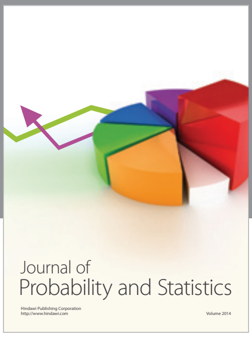
[38] T. Yi, H. Li, and M. Gu, "Optimal sensor placement for structural health monitoring based on multiple optimization strategies," *The Structural Design of Tall and Special Buildings*, vol. 20, no. 7, pp. 881–900, 2011.


Advances in
Operations Research

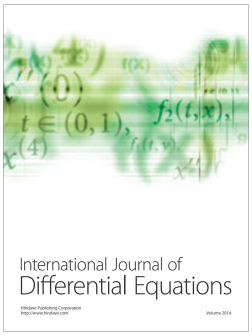

Advances in
Decision Sciences


Journal of
Applied Mathematics

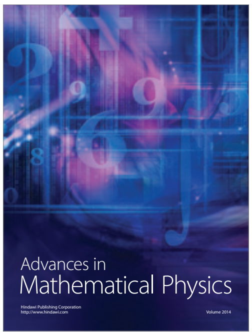

Algebra

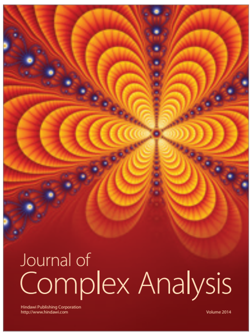

Journal of
Probability and Statistics

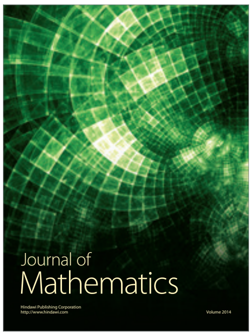
The Scientific
World Journal

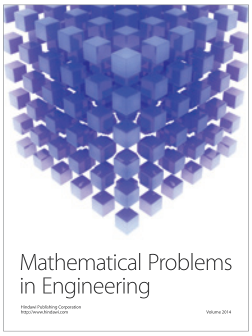

International Journal of
Differential Equations

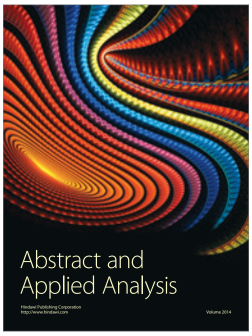
International Journal of
Combinatorics

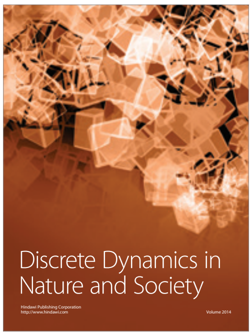

Advances in
Mathematical Physics

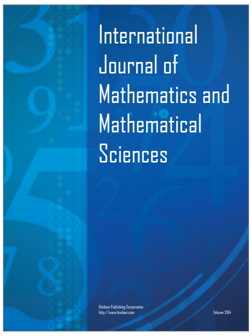

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

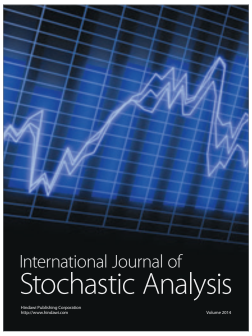

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization