Truss Damage Detection Using Modified Differential Evolution Algorithm with Comparative Studies

Sumin Kim, Nam Il Kim, Hyunjoo Kim, T. N. Nguyen, Q. X. Lieu and Jaehong Lee

Abstract In this paper, an efficient numerical algorithm is developed for the damage detection of planar and space truss structures based on the modified differential evolution algorithm (mDE) and vibration data. For this purpose, the mathematical programming of the finite element based on the force method and the singular value decomposition technique is presented. The general equilibrium equations in which unknown member forces and reaction forces are taken into account are formulated. The compatibility equations in terms of forces are explicitly presented by using the singular value decomposition method. The modified differential evolution algorithm (mDE) is used as an optimization algorithm of damage detection. The objective function for damage detection is based on vibration data such as natural frequencies and mode shapes. The feasibility and efficiency of the present method are compared with a genetic algorithm (GA) and a particle swarm optimization (PSO) for example. The numerical results show that the proposed strategy based on force method using mDE and vibration data can provide a reliable tool on determining the sites and extents of multiple damages of truss structures.

Keywords Damage detection · Force method · Free vibration analysis
Modified differential evolution algorithm (mDE)

1 Introduction

The identification of structural damage is an important part of the monitoring and repairing of structural systems during their functional age. In order to improve the safety and life expectancy of structures, it is necessary to detect the damage extents and sites. The damage assessment technique has drawn wide attention from various

S. Kim · N. I. Kim · H. Kim · T. N. Nguyen · Q. X. Lieu · J. Lee (✉)
Department of Architectural Engineering, Sejong University, Seoul, Republic of Korea
e-mail: jhlee@sejong.ac.kr

S. Kim
e-mail: sumkim9355@gmail.com

H. Nguyen-Xuan et al. (eds.), *Proceedings of the International Conference on Advances in Computational Mechanics 2017*, Lecture Notes in Mechanical Engineering, https://doi.org/10.1007/978-981-10-7149-2_1

engineering fields such as aerospace, mechanical, and civil engineering applications. The proposed identification of structural damage prevents the deterioration in a structural system from the impact loads and the responses of the structure. For damage detection and structural health monitoring, techniques based on vibration data have been extensively used. Thus, the damage evaluation of structures has derived wide consideration from engineering fields.

The various computational techniques have been used to solve the damage detection problem such as genetic algorithm (GA), differential evolution algorithm (DE), and particle swarm optimization (PSO). Among non-gradient-based methods, GA has been used widely from damage detection field. The GA is a well-known global optimizer based on the Darwin's theory of evolution and survival of the fittest and the natural process through reproduction [1]. It makes the mechanisms of biological evolution to perform optimization without information regarding the objective function. Generally, the GA has been shown to be solved various optimization problems through some basic concepts and operators. However, some research work has appeared that it is complex for the simple GA to solve the damage quantification problems properly. To achieve a better detection result, the various optimization algorithms were improved [2–4]. Krishnakumar [5] proposed the micro-genetic algorithm based on serial genetic algorithms [6], and Koh and Dyke [2] proposed the use of correlation-based damage detection methods for long-span cable-stayed bridge. Nobahari and Seyedpoor [7] proposed a modified genetic algorithm (MGA) to identify multiple damage and two new tools to recognize the actual damage correctly and rapidly. Also, he used the DE algorithm to solve the optimization problem for detecting actual sites and elements which was demonstrated by truss structures. The GA has discrete design variables, and new attempts based on continuous domain have been proposed such as PSO, DE, and ant colony optimization (ACO).

Turning to the problems of global optimization method, Storn and Price [8] proposed the differential evolution (DE) algorithm as a simple and powerful population-based stochastic method, which has an operation process similar to GA. In order to overcome the shortcomings of the non-gradient-based methods regarding computational time as well as of the gradient-based methods concerning solution accuracy, several variations of DE algorithm have been suggested. Pampara et al. [9] suggested a binary DE using trigonometric functions, in which experimental results indicate the effectiveness of the technique and the viability for the DE to operate in binary space. Zou et al. [10] proposed to solve constrained optimization problems using a modified differential evolution algorithm. Its own crossover rate and scale factor were adjusted using uniform distribution, and mutation phase was modified to enhance the convergence for unconstrained optimization problems. Concerning the damage detection, Vo-Duy et al. [11] used a strain energy-based method and an improved differential evolution algorithm to locate and quantify damage on the laminated composite plate. Also, Dinh-Cong et al. [12] proposed a two-stage damage identification method to identify the location and extent of multiple damages at individual layers in laminated composite beams using DE algorithm. Jena et al. [13] presented a damage detection technique combining analytical and experimental investigations on a cantilever aluminum alloy beam with a transverse surface crack. The damage location and assessment is the third stage and is formulated as a constrained optimization problem and solved using the proposed differential evolution algorithm.

Based on the mentioned researches, the mDE has not yet been used as optimizer for 2D and 3D truss structures based on force method and vibration data until now. In this study, a mDE combined with the free vibration analysis is proposed as a highly efficient algorithm for performing the damage identification of truss structures. The introduced algorithm resolves efficiently the limitation of GA by randomly continuous design variables instead of discrete domain as in the GA. In order to solve the problem, the number of design variables and member types of truss structures are first defined. Design variables are randomly created in the range of lower and upper bounds based on the number of design variables. Consequently, based on the force method with the present mDE in the damage detection process, the results satisfying the required deficiencies of the force method and equilibrium matrices are selected.

2 Theoretical Modeling

In order to build a model for truss structure, the general equations and relations are briefly presented. Several approaches that are similar to those can be found in the literature [14, 15]. For a d-dimensional ($d = 2$ or 3) truss structure with m members, n free nodes, and n_c constrained nodes, its topology can be expressed by a connectivity matrix $\mathbf{C}_s$ [16]. Let r_c be the unknown vector of reaction forces employed to remove all boundary constraints. Therefore, the constrained nodes can now be treated as free nodes. In the context, the general equilibrium equations for all nodes (including constrained nodes) of the discrete truss can be written as follows:

$$\bar{\mathbf{A}}\bar{\mathbf{f}} = \bar{\mathbf{p}} \tag{1}$$

where $\bar{\mathbf{A}}$ is called the general equilibrium matrix of all nodes, which transforms the vector of member and reaction forces $\bar{\mathbf{f}}$ to the vector of external loads $\bar{\mathbf{p}}$, $\bar{\mathbf{f}}$ is the unknown member and reaction force vector, and $\bar{\mathbf{p}}$ is the external load vector of all nodes.

In this study, the SVD method is used as an effective and numerically stable algorithm in order to extract r compatibility conditions. The SVD is performed on the general equilibrium matrix $\bar{\mathbf{A}}$ instead of the kinematic matrix $\mathbf{B}$ (i.e., the transpose of the general equilibrium matrix $\bar{\mathbf{A}}$). The compatibility equations expressed in terms of forces are obtained as follows:

$$\mathbf{D}\bar{\mathbf{f}} = \mathbf{p}_{e_o} \tag{2}$$

where $\mathbf{D}$ is the compatibility matrix in terms of forces of all members and $\mathbf{p}_{e_o}$ is the virtual force vector in the indeterminate truss caused by member initial elongations $\mathbf{e}_o$.

For the free vibration analysis, the extended equation of motion based on the force method can be written as

$$\bar{\mathbf{M}}\ddot{\bar{\mathbf{d}}} + \tilde{\mathbf{A}}\bar{\mathbf{f}} = 0 \tag{3}$$

where $\bar{\mathbf{M}}$ is the extended mass matrix, $\ddot{\bar{\mathbf{d}}}$ is the extended acceleration vector, and $\tilde{\mathbf{A}}$ is the square full-rank matrix coupling the general equilibrium matrix and the compatibility matrix. For this analysis case, it is assumed that element forces are harmonics in time, $\bar{\mathbf{f}} = \bar{\mathbf{F}}\sin(\omega t)$, where ω and $\bar{\mathbf{F}}$ are the circular frequency and corresponding force mode shape, respectively. Therefore, equation [14] can be written as the following eigenvalue problem

$$(\tilde{\mathbf{A}} + \omega^2\mathbf{B})\bar{\mathbf{F}} = 0 \tag{4}$$

where

$$\mathbf{B} = \bar{\mathbf{M}}\tilde{\mathbf{A}}^{-T}\mathbf{G} \tag{5}$$

and $\bar{\mathbf{M}}$ is the extended mass matrix.

3 Structural Damage Detection

To determine the location and extent of the damages more accurately, the natural frequency and the mode shape changes are demanded $messina_structural_1998, wang_structural_2001$. In this study, the objective function based on the sum of two formulations is given as follows:

$$f(\mathrm{X}) = f_1(\mathrm{X}) + f_2(\mathrm{X}) \tag{6}$$

where

$$f_1(\mathrm{X}) = \sum_{i=1}^{n} \frac{|\omega_{di} - \omega_i(\mathrm{X})|}{\omega_{di}} \tag{7}$$

and

$$f_2(\mathrm{X}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \left\| \frac{\Phi_{dj} - \Phi_j(\mathrm{X})}{\Phi_{dj}} \right\| \tag{8}$$

where ω_i is the corresponding hypothesis natural frequency which is can be expected from analytic model and ω_{di} is the ith natural frequency of the damaged truss. Φ_{dj} and $\Phi_j(X)$ are the jth component of the ith damaged force eigenvector and the jth component of the ith hypothesis force eigenvector, respectively, in $f_2(\mathrm{X})$. Then, the optimal design problem for damage identification of the truss structure is formulated as follows:

$$\text{Minimize} \quad f(X) + \gamma g(X) \tag{9}$$

in which

$$g(X) = (n_d - n_h)^2 \quad \text{for} \ \ n_d > n_h \tag{10}$$

where x_i is the damaged elastic modulus of the ith member, γ is the penalty parameter, n_d and n_h are the number of the damaged elements and the number of the healthy elements, respectively. In real life, the healthy members are more than the damaged members in structures. Thus, both algorithms in this study can be accelerated to find better solution using the penalty function $g(X)$. In many researches related to the damage detection problem, the damage has been simulated by decreasing one of the stiffness parameters of the element. In this study, the damage of truss structures is simulated by the reduction in the elasticity modulus in each element as

$$x_i = \frac{E - E_i}{E}, \qquad i = 1, \ldots, n \tag{11}$$

where E is the initial elastic modulus and E_i is the final elastic modulus of nth element.

4 Optimization Algorithm

4.1 Differential Evolution Algorithm

Differential evolution (DE) has been first proposed by Storn and Price [8] as a vector-based metaheuristic algorithm. The DE has similarity to genetic algorithms, its use of selection, crossover, and mutation. The DE is a stochastic search algorithm with self-organizing tendency and does not use the information of derivatives [17]. Thus, it is a population-based, derivative-free method. In addition, DE uses real numbers as solution strings, so no encoding and decoding is needed. The main procedure of DE includes mutation, crossover, selection, and initialization of design variables.

4.1.1 Initialization

In the DE algorithm, d-dimensional real number of design variable vectors is randomly generated as much as population size for global optimum point. The ith vector of the population at any iteration can be expressed by conventional notation as

$$X_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,d}\} \tag{12}$$

where X_i is a vector at ith population and d is a number of design variables in X_i vector. It is noted that this vector, called target vector, can be considered as chromosomes. A d-dimensional vector represents a candidate result to the problem and is randomly made in the domain as follows:

$$x_{i,j} = x_{\min,j} + rand \times (x_{\max,j} - x_{\min,j}),\ \ i = 1, \overline{\mathrm{NP}},\ \ j = 1, \bar{n} \tag{13}$$

where NP is the population size, n is the number of design variables or dimensions, *rand* is a uniformly distributed random number in the interval [0,1], $x_{i,j}$ is the jth component of individual $\mathbf{x}_i$, and $x_{\min,j}$ and $x_{\max,j}$ are the lower and upper bounds of x_j, respectively.

4.1.2 Mutation

In the mutation step, DE generates a mutant vector $\mathbf{v}_i$ based on the target vector $\mathbf{x}_{r_1}$ by using mutation operation at the current iteration. The frequently used some mutation schemes are introduced as below:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \times (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{14}$$

where $\mathbf{v}_i$ is a mutation vector and known as a new candidate solution; $\mathbf{x}_{r_1}$, $\mathbf{x}_{r_2}$, and $\mathbf{x}_{r_3}$ are candidate solutions; r_1, r_2, and r_3 are randomly determined from [1, NP] to satisfy the constraints as $r_1 \neq r_2 \neq r_3 \neq i$; the scale factor F can be selected within the range $F \in [0, 2]$.

4.1.3 Crossover

In the crossover process, trial vector $\mathbf{u}_i$ is created by combining the target vector $\mathbf{x}_i$ and the mutant vector $\mathbf{v}_i$. The combination of these vectors in this process is controlled by a crossover ratio (CR) chosen in the range [0,1]. The CR controls how likely it is that each component of $\mathbf{u}_i$ comes from the mutant vector $\mathbf{v}_i$ and is defined by user. The range [0.1,0.8] or 0.5 can get good result at the beginning.

The crossover phase is generally performed two schemes: binomial and exponential. Due to its simplicity, the binomial crossover is employed and defined as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } (rand_{i,j} < CR) \text{ or } (j = j_{rand}) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{15}$$

in which $rand_{i,j}$ is a uniformly distributed random number [0,1] and j_{rand} is the integer chosen from [0,1].

4.1.4 Selection

Finally, the $\mathbf{u}_i$ and $\mathbf{x}_i$ vectors are compared so that the most fit vector in each pair is kept for the next generation and the least fit is discarded. This selection process is essentially the same as that used in GA and expressed as

$$\mathbf{x}_i^{new} = \begin{cases} \mathbf{u}_i, & \text{if} f(\mathbf{u}_i) \leq f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases} \tag{16}$$

where $f(\mathbf{u}_i)$ and $f(\mathbf{x}_i)$ are the objective functions.

4.2 A Modified Differential Evolution Algorithm

In this section, mutation and selection phases are modified to overcome the drawbacks of the non-gradient- and gradient-based algorithms.

4.2.1 Modification of the Mutation Phase

Two mutation operators are adaptively chosen based on the absolute value of deviation of objective function between the best individual and the entire population in the previous generation (denoted as *delta*). More specifically, the value of delta is defined by

$$delta = \left| \frac{f_{mean} - f_{best}}{f_{best}} \right| \tag{17}$$

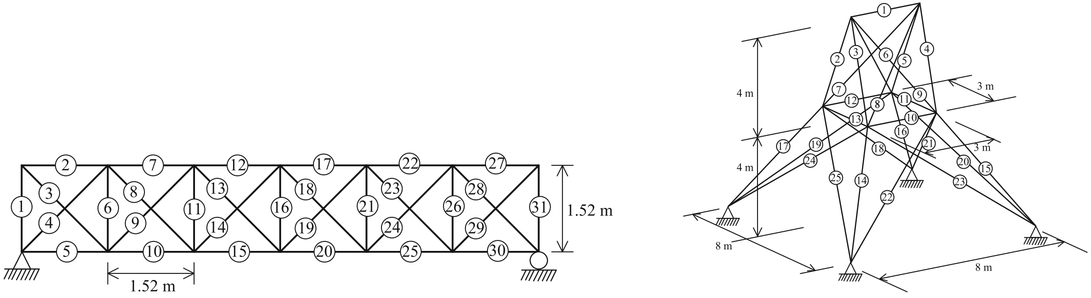
where f_{mean} is the mean objective function value of the whole population and f_{best} is the objective function value of the best individual. The new mutation scheme is described as follows:

$$\mathbf{v}_i = \begin{cases} \mathbf{x}_{best} + F^k \times (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}), & \text{if } delta > threshold \\ \mathbf{x}_{r_1} + F^k \times (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), & \text{otherwise} \end{cases} \tag{18}$$

where F^k is a number randomly chosen in [0,1] at the *k*th iteration; *threshold* is a criterion value which is chosen based on the stopping criterion of the algorithm.

4.2.2 Modification of the Selection Phase

In the DE's selection, the vectors are compared and unselected individual is worse than its target individual in the pair, but it can be still better than other individuals in the entire population. Consequently, several good information of unselected indi-



(a) The 31-bar planar truss

(b) The 25-bar space truss

Fig. 1 Two- and three-dimensional truss structures

viduals can be omitted and the convergence speed of the DE algorithm is slow. The good individuals are kept for the next iteration as follows:

$$\mathbf{Q} = \mathbf{P} \cup \mathbf{U} \tag{19}$$

Table 1 Induced elastic modulus for 31-bar planar truss

Member	x_i	GA (dq)				PSO	This study
		0.1	0.05	0.01	0.001		
1	0	0.0	0.00	0.00	0.008	0.027	0.073
2	0	0.0	0.00	0.00	0.000	0.050	0.034
3	0	0.0	0.00	0.02	0.152	0.065	0
4	0	0.0	0.00	0.00	0.000	0.040	0.006
5	0	0.0	0.00	0.00	0.078	0.060	0.022
6	0	0.0	0.00	0.15	0.000	0.062	0.01
7	0	0.0	0.00	0.00	0.000	0.008	0.006
8	0.475	0.5	0.50	0.42	0.407	0.475	0.475
9	0	0.0	0.00	0.00	0.000	0.009	0.016
10	0	0.0	0.00	0.00	0.000	0.011	0
11	0	0.2	0.00	0.11	0.011	0.021	0.099
12	0	0.0	0.00	0.00	0.000	0.018	0.085
13	0	0.0	0.00	0.05	0.038	0.033	0.088
14	0	0.0	0.00	0.00	0.047	0.065	0.076
15	0	0.0	0.00	0.00	0.012	0.006	0.048
16	0.319	0.1	0.30	0.17	0.000	0.319	0.319
17	0	0.0	0.00	0.00	0.008	0.045	0.058
18	0	0.0	0.00	0.06	0.000	0.073	0.021
19	0	0.0	0.00	0.04	0.000	0.081	0.01
20	0	0.0	0.00	0.00	0.088	0.010	0.005
21	0	0.2	0.00	0.06	0.000	0.008	0.036
22	0	0.0	0.00	0.00	0.002	0.078	0
23	0	0.0	0.00	0.00	0.148	0.035	0.013
24	0	0.0	0.00	0.07	0.000	0.015	0.002
25	0	0.0	0.00	0.00	0.000	0.005	0.018
26	0	0.0	0.00	0.05	0.096	0.019	0.03
27	0	0.0	0.00	0.00	0.000	0.026	0.029
28	0	0.0	0.00	0.00	0.010	0.048	0.003
29	0	0.0	0.00	0.00	0.082	0.022	0.013
30	0	0.0	0.00	0.00	0.000	0.050	0.088
31	0	0.0	0.00	0.00	0.000	0.024	0
$f(X)$		1.86E-2	6.62E-3	4.19E-2	1.41E-1	4.37E-8	3.86E-11

where **P** and **U** are populations consisting of NP individuals $\mathbf{x}_i$ and $\mathbf{u}_i$ with $i = 1, \overline{\mathrm{NP}}$, respectively; **Q** includes all individuals of **P** and **U**. Then, NP best individuals in **Q** are chosen to create a new population for the next iteration. Thus, the best solutions of the whole population are always stored for the next generation and the convergence rate is significantly improved.

5 Numerical Examples

In this section, two- and three-dimensional truss structures are investigated using GA, PSO, and proposed mDE algorithm. The searching process of these algorithms will be stopped when the maximum number of iterations is reached. In all examples, the crossover ratio (CR) for the proposed mDE and GA is set to be 0.6 and 1.0, respectively. Accordingly, all populations have to perform a crossover operation at all generation and there is no use of mutation process. The first three natural frequencies and mode shapes are used for detection. mDE algorithm and PSO find solution in a continuous domain, while results gained by GA are done in a discrete domain. The numerical results obtained by mDE are compared to GA and PSO to illustrate the accuracy and effectiveness of the proposed algorithm.

5.1 31-Bar Planar Truss

The first example is a 31-bar planar truss as shown in Fig. 1 and is modeled using 28 consistent finite elements without internal nodes leading to 25 degrees of freedom. The elasticity modulus $E = 70$ GPa, the mass density $\rho = 2{,}770$ kg/m^3, and the area of cross section $A = 0.01$ m^2. For every algorithms, the maximum number of

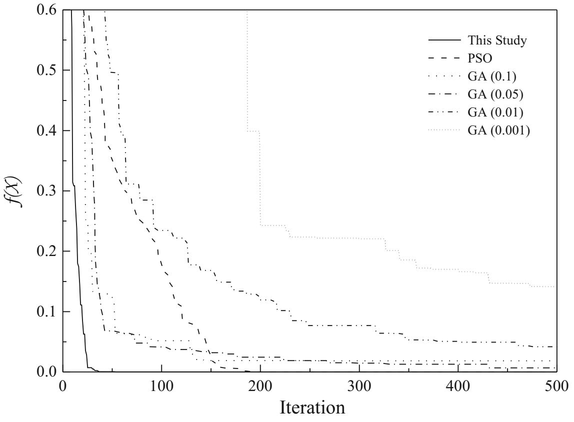


Fig. 2 Convergence history of the 31-bar planar truss

iterations and population size are respectively set to be 500 and 300. In damage case, as damage ratios, the stiffness reduction in elasticity modulus 0.475 and 0.319 is induced at elements 8 and 16, respectively. The detected damages are shown by comparing the ratios of elasticity modulus reduction in GA, PSO, and mDE. Here, GA is considered with four different step points $dq = 0.1$, 0.05, 0.01, and 0.001. The obtained results are expressed in Table 1 in comparison with each step points of GA and PSO. From this table, the damage location and extent of proposed algorithm were successfully determined than other cases of GA and agree well with this acquired by PSO. In a narrow discrete domain of search space, GA cannot find exact damage. Thus, GA have some difficulties about the accuracy of the solution if real solutions have many digits after the decimal point. Figure 2 shows the comparison

Table 2 Induced elastic modulus for 25-bar space truss

Member	x_i	GA (dq)				PSO	This study
		0.1	0.05	0.01	0.001		
1	0	0.0	0.40	0.00	0.596	0.022	0
2	0	0.0	0.00	0.00	0.006	0.011	0.008
3	0	0.0	0.00	0.03	0.022	0.008	0.006
4	0	0.0	0.05	0.00	0.066	0.029	0.052
5	0	0.0	0.00	0.00	0.276	0.064	0.056
6	0.383	0.4	0.40	0.45	0.328	0.383	0.383
7	0	0.0	0.00	0.04	0.194	0.048	0.061
8	0	0.0	0.15	0.00	0.004	0.069	0.02
9	0	0.0	0.00	0.00	0.069	0.040	0
10	0	0.1	0.25	0.24	0.055	0.052	0.01
11	0	0.1	0.00	0.08	0.295	0.092	0
12	0	0.0	0.00	0.27	0.089	0.077	0.064
13	0	0.0	0.00	0.00	0.232	0.041	0
14	0	0.0	0.00	0.00	0.137	0.030	0.024
15	0	0.0	0.00	0.00	0.047	0.071	0.039
16	0.294	0.3	0.40	0.32	0.522	0.294	0.294
17	0	0.0	0.05	0.01	0.002	0.019	0.04
18	0	0.0	0.00	0.00	0.097	0.051	0.022
19	0	0.0	0.05	0.00	0.019	0.087	0.058
20	0	0.0	0.00	0.03	0.070	0.047	0.066
21	0	0.0	0.00	0.00	0.006	0.085	0.055
22	0	0.0	0.00	0.00	0.045	0.062	0.02
23	0	0.0	0.00	0.04	0.018	0.082	0.001
24	0	0.0	0.05	0.00	0.165	0.088	0.025
25	0	0.0	0.00	0.00	0.106	0.063	0.096
$f(X)$		3.37E-02	2.05E-01	1.34E-01	8.65E-01	9.76E-7	2.80E-11

of the convergence of each cases of GA, PSO and mDE at the whole iterations. As shown in the figure, mDE converges faster and lower than GA. For more detail, the mDE converges in 00 iterations with an objective function $f(\mathrm{X}) = 3.86 \times 10^{-11}$ in the first damage case, while each step point of GA converges 1.86×10^{-2}, 6.62×10^{-3}, 4.19×10^{-2}, and 1.41×10^{-1} corresponding step points 0.1, 0.05, 0.01 and 0.001, respectively.

5.2 25-Bar Space Truss

The 25-bar space truss is shown in Fig. 1, with 10 nodes leading to 18 degrees of freedom. The area of cross section $A = 0.25$ m^2, the material density $\rho = 7{,}830$ kg/m.^{3}, and the modulus of elasticity $E = 210$ GPa. Likewise, the maximum number of iterations and population size are, respectively, set to be 500 and 300.

As a case of damage, the stiffness reductions in elasticity modulus 0.383 and 0.294 are induced at elements 6 and 16, respectively. The results gained by this example in comparison with the GA and PSO are presented in Table. 2. It can be seen that the present algorithm has shown excellent performance for damage detection than all step points of GA and agrees well with the result of PSO. When the step point increases, the gained results of GA are less accurate. Moreover, the convergence history of GA, PSO, and mDE are shown in Fig. 3. It demonstrates that mDE converges even faster than GA. Specially, mDE converges as 2.80×10^{-11}, while each step point of GA converges as 3.37×10^{-2}, 2.05×10^{-1}, 1.34×10^{-1}, and 8.65×10^{-1} corresponding to step points $dq = 0.1$, 0.05, 0.01, and 0.001, respectively, and the value of converged objective function from PSO is 9.76×10^{-7}.

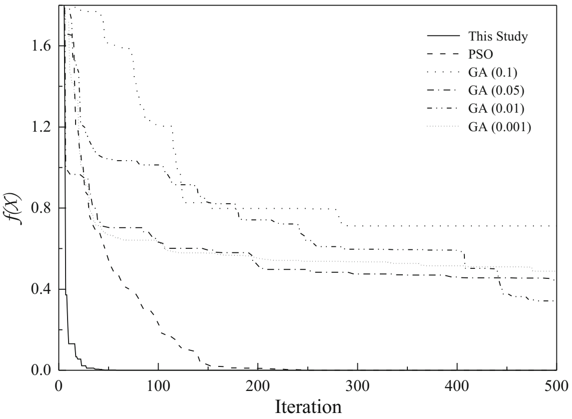


Fig. 3 Convergence history of the 25-bar space truss

6 Conclusion

This study proposed the optimization technique which properly determines the locations and extents of multiple damages of planar and space trusses based on the modified differential evolution (mDE) algorithm with comparative studies. To generate the compatibility conditions for indeterminate trusses, the singular value decomposition technique is employed on the general equilibrium equations. The force mode vectors are introduced as eigenvectors in the objective function. The optimization problem has been solved using mDE which has continuous design variables to identify the actual damages. Three illustrated test examples such as planar and space trusses are considered in order to assess the performance of the proposed method. Throughout the numerical examples, the relative performance of mDE, GA, and PSO in the damage detection of trusses is studied. Numerical results show that the combination of the present force method and mDE is far more efficient than PSO and GA with discrete design variables for identifying multiple damages of trusses.

References

1. Michalewicz Z (1994) GAs: What are they? In: Genetic algorithms + data structures = evolution programs. Springer, Heidelberg, p. 13–30. https://doi.org/10.1007/978-3-662-07418-3_2
2. Bertalmio M, Sapiro G, Caselles V, Ballester C (2000) Image inpainting. In: Proceedings of the 27th annual conference computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp 417–424
3. Chan T, Vese L (2001) Active contours without edges. IEEE Trans Image Process 10(2):266–277
4. Criminisi A, Pérez P, Toyama K (2004) Region filling and object removal by exemplar-based image inpainting. IEEE Trans Image Process 13:1200–1212
5. Emerson C, Lam N, Quattrochi D (1999) Multi-scale fractal analysis of image texture and patterns. Photogramm Eng Remote Sens 65(1):51–62
6. Farias M, Mitra S (2005) No-reference video quality metric based on artifact measurements. In: Proceedings of the international conference on image processing, vol 3, pp III–141–4
7. Faugeras O, Lustman F (1988) Motion and structure from motion in a piecewise planar environment. Technical report RR-0856, INRIA
8. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
9. Google images (2012). http://www.images.google.com
10. Guillemot C, Meur OL (2014) Image inpainting: overview and recent advances. IEEE Signal Process Mag 31(1):127–144
11. Hartley R, Zisserman A (2003) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, New York, NY, USA
12. Jobson DJ, Rahman Z, Woodell GA (1997) Properties and performance of a center/surround retinex. IEEE Trans Image Process 6(3):451–462
13. Katahara S, Aoki M (1999) Face parts extraction windows based on bilateral symmetry of gradient direction. Computer analysis of images and patterns, vol 1689. Springer, Heidelberg, pp 834–834
14. Kenney JF (1954) Mathematicals of statistics. Van Nostrand

15. Kovesi PD (2005) MATLAB and Octave: functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. http://www.csse.uwa.edu.au/pk/research/matlabfns/robust/ransacfithomography.m
16. Legrand P (2009) Local regularity and multifractal methods for image and signal analysis. In: Scaling, fractals and wavelets. Wiley
17. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110