Cooperative Coevolution with Dynamic Species-Size Strategy for Vibration-Based Damage Detection in Plates

Jitti Pattavanitch
Dept. of Mechanical Engineering,
Faculty of Engineering,
Burapha University,
Chonburi, Thailand.
jitti@eng.buu.ac.th

Puttha Jeenkour
Dept. of Mechanical Engineering,
Faculty of Engineering,
Burapha University,
Chonburi, Thailand.
puttha@eng.buu.ac.th

Kittipong Boonlong
Dept. of Mechanical Engineering,
Faculty of Engineering,
Burapha University,
Chonburi, Thailand.
kittipong@eng.buu.ac.th

***Abstract*—Vibration-based damage detection is based on the fact that vibration characteristics such as natural frequencies and mode shapes of structures are changed when the damage occurs. This paper proposes dynamic species-size strategy in cooperative coevolution concept. The resulting algorithm, cooperative coevolutionary genetic algorithm with dynamics species-size (CCGADSS), is used as the optimization algorithm in the vibration-based damage detection in plates. The objective function is numerically calculated from the difference between experimentally vibration characteristics and numerically evaluated vibration characteristics of the predicted damage. In finite element model for objective calculation, the plates are equally divided into 64 elements. There are 2 different cases with dissimilar occurred damage in plates are considered. In first case, the plate hase only one region consisting of 4 elements which are together connected and have same damage. In second case, there are 5 separated elements which are damaged differently. In order to demonstrate the performance of the dynamic species-size strategy, 3 optimization algorithms, which are genetic algorithm (GA), cooperative coevolutionary genetic algorithm (CCGA), and CCGADSS. The results indicate that CCGADSS is superior to GA and CCGA. Moreover solutions obtained using CCGADSS are quite close the actual damage. These results show that the dynamic species-size strategy can enhance performance of cooperative coevolution concept.**

***Keyworkds*—dynamic species-size strategy; vibration-based damage detection; genetic Algorithm; co-operative co-evolution; plate structure; finite element method.**

I. INTRODUCTION

The vibration-based damage detection methods are non-destructive methods, which are important for structural integrity testing. When there is some damage in the structure, vibration characteristics such as natural frequencies and mode shapes are changed. These vibration characteristics are actually related to structural physical parameters such as mass, stiffness, and damping of the structure. The structural damage causes a reduction of structural stiffness so that vibration characteristics are consequently changed. Many applications in civil engineer and mechanical engineer [1-5] employed the vibration-based damage detection methods.

This method formulates structural damage identification to an optimization problem, the optimization algorithm is then to be used in this method. In optimization process, there are two main optimization approaches, derivative-based and derivative-free methods. Compared to the derivative-based schemes, the derivative-free methods do not need a functional derivative of a given objective. They, instead, rely on the repeated evaluations of the objective functions and perform the search direction under the nature-inspired heuristic guidelines. In the vibration-based damage detection methods, since it is impossible to formulate the objective function as the explicit function of predicted damage, the derivative-free methods should be considered.

Genetic algorithm (GA) [6, 7] is a derivative-free population-based optimization method of which search mechanisms are based on the Darwinian concept of survival of the fittest. There are a number of works [3, 4] used GAs as the optimization algorithms in the structural damage detection. Co-operative co-evolutionary genetic algorithm (CCGA) originally developed by Potter and De Jong [8, 9] is an improved version of GA, it is capable for a large-scale optimization problem. Many researches effectively applied the CCGA to the particular optimization problems [10-12]. According to Potter and De Jong's framework, a problem consisting of n variables is decomposed into *n* subcomponents which are subsequently assigned to *n* corresponding species. Evolution of these species is performed more or less independently of one another. Since any given species only represents subcomponent of a complete solution, collaborators from the other species are selected to exhibit the remaining subcomponents. Potter and De Jong [8] applied this framework for solving the optimization problems having decision variables up to 30. The experimented results revealed that CCGA outperformed the conventional GA on the separable problems, however, its performance turned to be worse on the non-separable problems. Instead of using static decomposition strategy, the species-size and the species member that can be dynamically adjusted during run time using pre-specified rules by algorithm settings could make a good sense for alleviating problems mentioned earlier. Yang et al. [13] proposed a dynamic decomposition strategy, the so-called "random grouping", based on probability assignment and integrated this idea with cooperative coevolutionary differential evolution (CCDE) for solving non-separable

IJARE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

problem with up to 1,000 variables. This decomposition method allows group size, i.e. species-size, to be varied throughout the course of optimization. Boonlong et al. [14] employed dynamic species-size strategy in cooperative coevolution for large-scale multi-objective portfolio optimization. In this paper, co-operative co-evolutionary genetic algorithm with dynamic species-size (CCGADSS) is utilized for solving damage detection problems in plates. In order to illustrate the search performance of CCGADSS, CCGADSS will be benchmarked against GA and CCGA.

II. OBJECTIVE CALCULATION

For free vibration of undamped structures, the equation of motion is given by the following equation.

$$[m]\ddot{\vec{x}} + [k]\vec{x} = \vec{0} \tag{1}$$

where $[m]$ and $[k]$ are mass and stiffness matrices respectively. The corresponding eigenvalue equation for vibration mode j is given by

$$[k]\{v_j\} - \lambda_j[m]\{v_j\} = \{0\} \tag{2}$$

where λ_j and $\vec{v}$ are eigenvalue which is the square of nature frequency and eigen vector or mode shape of j^{th} mode of vibration.

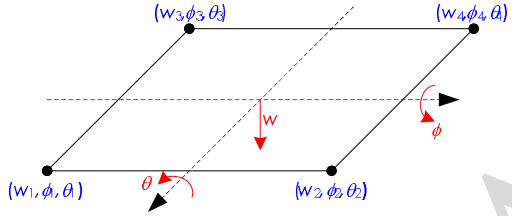


Fig 1. A rectangular divided element for finite element model.

His paper uses the finite element method for plates [15]. Fig. 1 shows a divided rectangular element. At each nodal point, there are 3 degrees of freedom which are deflection in z direction (w), twist angle about x-axis (ϕ) and twist angle about y-axis (θ). From the element, there are 4 nodal points so that 12 degrees of motion for computing local stiffness matrix $[k]_i$ and local mass matrix $[m]_i$.

In finite element model, the stiffness matrix $[k]$ and mass matrix $[m]$ can be calculated by the sum of their local matrices of all divided elements as the following equations.

$$[m] = \sum_{i=1}^{N}[m]_i \tag{3}$$

$$[k] = \sum_{i=1}^{N}[k]_i \tag{4}$$

Once the damage occurs in an element i of a structure, local damaged matrix $[k]_{di}$ is reduced from its local undamaged matrix $[k]_i$ according to damage factor (γ_i) of the element. The damaged local matrix can be computed by the following equation.

$$[k]_{di} = (1-\gamma_i)[k]_i \tag{5}$$

The values of the parameters γ_i fall in the range 0 to 1. The damage factor γ_i = 1 indicates that an completely damaged element and γ_i = 0 or less than 1 implies undamaged or partially damaged elements respectively.

Similar to equation (5) the stiffness matrix of the damaged structure is the sum of their local damaged matrices of all elements.

$$[k] = \sum_{i=1}^{N}[k]_{di} = \sum_{i=1}^{N}(1-\gamma_i)[k]_i \tag{6}$$

Moreover, it is assumed that the mass matrix is unchanged due to the occurred damage. In the optimization process, the decision variables are the predicted damage factors β_i of all divided elements so that the number of decision variables is equal to the number of divided elements. The objective function f is numerically calculated from the difference between the experimentally vibration characteristics - natural frequencies and mode shape - of the actual damage and those of the predicted damage [16] as shown in equation (7).

$$f = \sum_{i=1}^{NF} w_{\omega i}\Delta\omega_i^2 + \sum_{i=1}^{NM} w_{\bar{v}i}(1-MAC_i) \tag{7}$$

where $w_{\omega i}$ is a weight factor corresponding to the i^{th} natural frequency, while $w_{\Phi i}$ correspondingto i^{th} MAC. $\Delta\omega_i$ and MAC_i are the numerical indicators of the difference of i^{th} natural frequency and that of i^{th} mode shape as shown in equations (8) and (9) respectively. NF and NM are the numbers of natural frequencies and mode shapes used in the calculation.

$$\Delta\omega_i = \left|\frac{\omega_{ip} - \omega_{ie}}{\omega_{ie}}\right| \tag{8}$$

$$MAC_i = \frac{(\mathbf{v}_{ip}\cdot\mathbf{v}_{ie})}{|\mathbf{v}_{ip}|^2|\mathbf{v}_{ie}|^2} \tag{9}$$

where ω_{ip} and ω_{ie} are natural frequencies of predicted damage and actual damage of vibration mode i, while $\mathbf{v}_{ip}$ and $\mathbf{v}_{ie}$ are mode shapes of measured points of the predicted and actual damage of the vibration mode i. It can be noted that ω_{ip} and $\mathbf{v}_{ip}$ are related to the predicted damage, while ω_{ie} and $\mathbf{v}_{ie}$ are measured natural frequency and mode shape of the actual damaged occurred in the structure. If the damage is correctly

IJARE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

predicted, for all vibration mode i $\Delta\omega_i$ and MAC_i are equal to 0 and 1 respectively, the objective function f is then equal to 0.

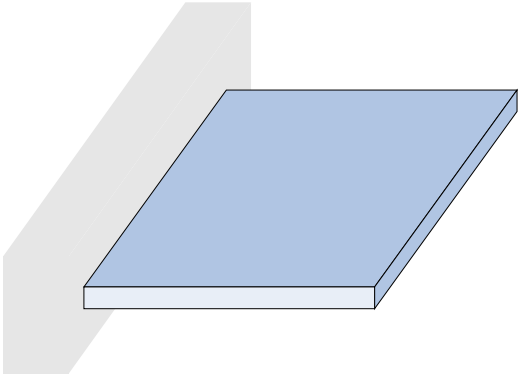


Fig 2. A test problem.

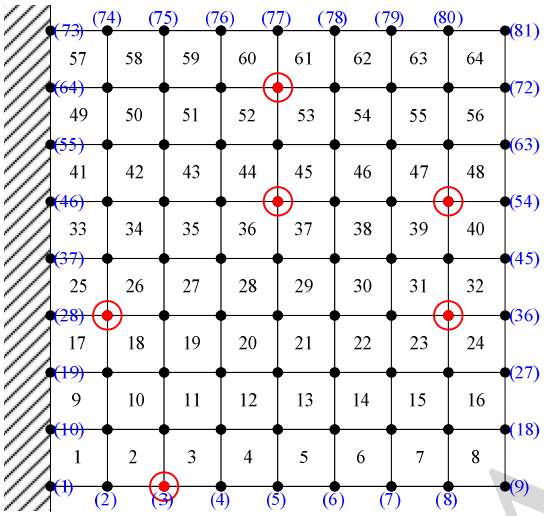
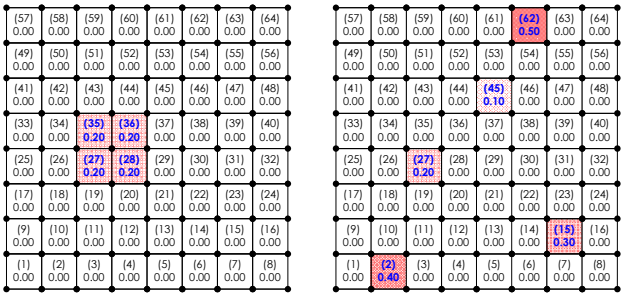


Fig 3. Divided elements and 6 mesured points in circles

III. Test Problem

The damage detection in a cantilever plate (Fig. 2) used as the test problem. The plate has the dimensions of 100 cm × 100 cm × 5 cm. The properties of the plate are as follows: modulus of elasticity (E) = 210 GPa, Poisson'ratio (ν) = 0.3, and density (ρ) = 7,800 kg/m^3. In finite element model, the plate is divide into 64 square elements of which each has 4 nodal points. In experimental approximation, there are 6 measured points as shown in Fig. 3 where deflection in z direction (w) can be measured. Two different cases of the test problem are considered – (1) the plate having only one region consisting of elements 27, 28, 35, and 36 which are together connected damaged partially to an extent of 20%, and (2) the plate having elements 2, 15, 27, 45, and 62 damaged partially 40%, 30%, 50%, 60%, and 20% respectively. shows the damage occurred in each elements of both cases of the test problem.



(a) Case 1 (b) Case 2

Fig 4. Damage factors of each element.

IV. Optimization Algorithm

A. Genetic Algorithm

Genetic algorithm (GA) [7, 8] is a derivative-free population-based optimization method of which search mechanisms are based on the Darwinian concept of survival of the fittest. At the beginning of GA, individuals of an initial population are randomly generated. The individuals could be encoded by binary-coded chromosomes or real-coded chromosomes. The chromosome of every individual is decoded in order to obtain particular solutions of which objectives can be determined. The fitness value of each individual is then evaluated using the obtained objective value. By using the fitness values, a parent population is selected from the current population. The selected population is performed a transformation using genetic operators, crossover and mutation, to obtain the resulting updated population. The individuals in the population is decoded to be solutions of a problem again. The search mechanism is repeated until termination condition is satisfied. Once the search is terminated, the latest best solution is the output of the algorithm

B. Cooperative Coevolution

The co-operative co-evolutionary genetic algorithm (CCGA) explores the search space by utilizing a population which contains a number of species or sub-populations. Each species is independently evolved as the procedure of employed algorithm. In each species, an individual i represents only a decision variable or part of a solution to a problem. A combination of an individual i with corresponding other variables or solution parts will lead to a complete solution to the problem where the objective value of the complete solution can then be determined. By partitioning the solution into species, the search space that each species has to cover is significantly reduced compared to the full solution searches. The CCGA produces best performances when there is completely no coupling between different species.

For instance, a solution is encoded into real-coded chromosome of length 10 in which each bit is represented by the number with 2 decimal places whose value is from 0.00 to 1.00. The number of possible solutions is $101^{10} = 1.05\times10^{20}$. By dividing the binary chromosome into 10 un-coupled species and each species is represented by a 1-bit real string, the number of possible solutions is then reduced to only 10×101 = 1,010 that makes the algorithm searches solutions

IJARE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

much easier. However, if there is coupling between species, search performances will be deteriorated with an increasing of the coupling strength.

C. Cooperative Coevolution with Dynamic Species-size Strategy

With the use of static decomposition strategy in CCGA, it disreputable for solving a non-separable problem. The presence of dynamic decomposition strategy that allows CCGA to alter species-size and species member during the algorithm runs could be the efficient way the handle the non-separable problem [13]. Li & Yao [17] demonstrated that by changing species-size frequently (the so-called "grouping size" in their work), the probability of two interacting variables that will be placed into the same subcomponent is significantly increased.

This paper developed a CCGA with dynamic species-size method. In this method, the set of dynamic species-size (*SS*) is initially defined. The minimum *SS* is 1 and the next *SS* is 2 times the previous one, i.e. {1, 2, 4, 8, 16, ...}. The algorithm starts with the randomly selected and then removed *SS* from the set, and defined species indices on *SS* groups of decision variables at random. From the initial random population, the best solution is obtained. The species-size set **SS** is initially identified so that $\mathbf{SS} = \mathbf{SS}_0$ = {1, 2, 4, 8, 16, ...}. }. The species size (*SS*) is randomly picked and removed from **SS**. After the implementation of one species, if the best solution is not changed which mean that this *SS* is converged. The other SS is therefore randomly selected and removed from the current **SS** and so on. Once **SS** is empty, it is re-initialized to be $\mathbf{SS}_0$. The process is repeated again until it meet a termination condition.

V. RESULTS AND DISCUSSIONS

In order to demonstrate the performance of the dynamic species-size strategy, 3 optimization algorithms, which are genetic algorithm (GA), cooperative coevolutionary genetic algorithm (CCGA), and CCGA with dynamics species-size (CCGADSS) will be used in the test problem previously described. The parameter setting for solution search of the algorithms is showed in Table I.

Table I. Parameter Setting.

Parameter	**Setting and values**
Chromosome coding	Real-valued chromosome with 64 decision variables
Number of species species size (***SS***)	**SS** ={1} for CCGA **SS** ={1, 2, 4, 8, 16, 32, 64} for CCGA with dynamic species-size
Crossover method	Simulated-binary crossover with probability = 1.0
Mutation method	Variable-wise polynomial mutation
Mutation probability	1/chromosome length for GA 1/species-size for CCGA and CCGADSS
Population size	20
Number of iterations	3200

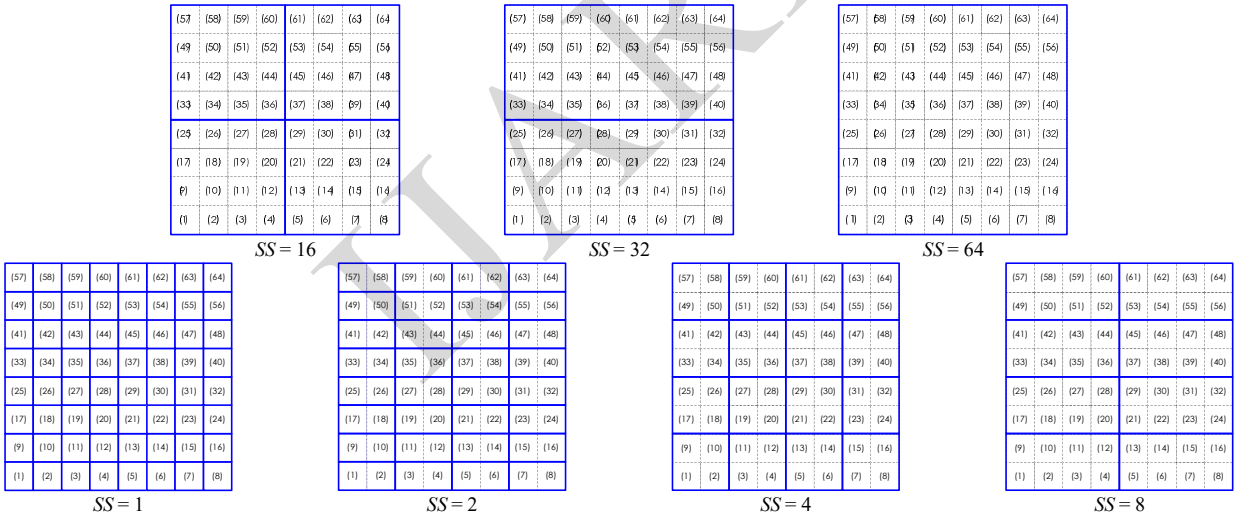
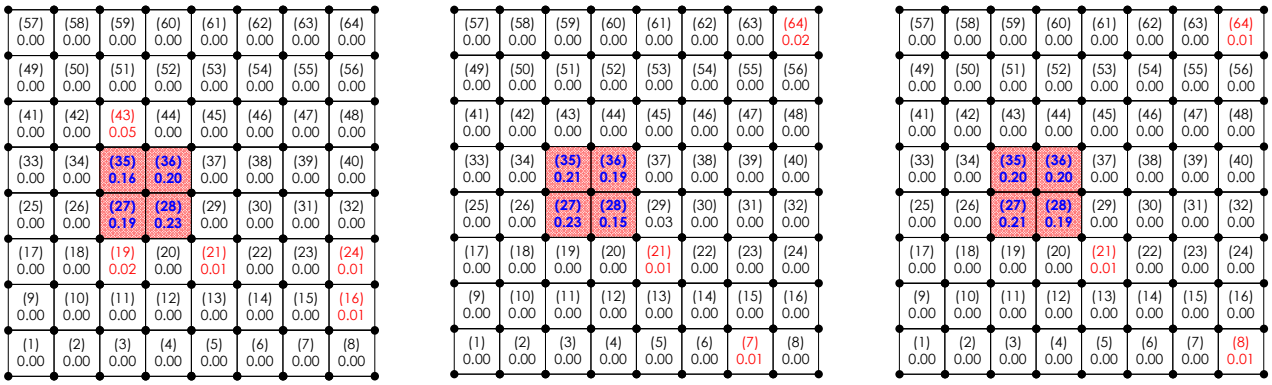


Fig 5. Species size used in CCGADSS.

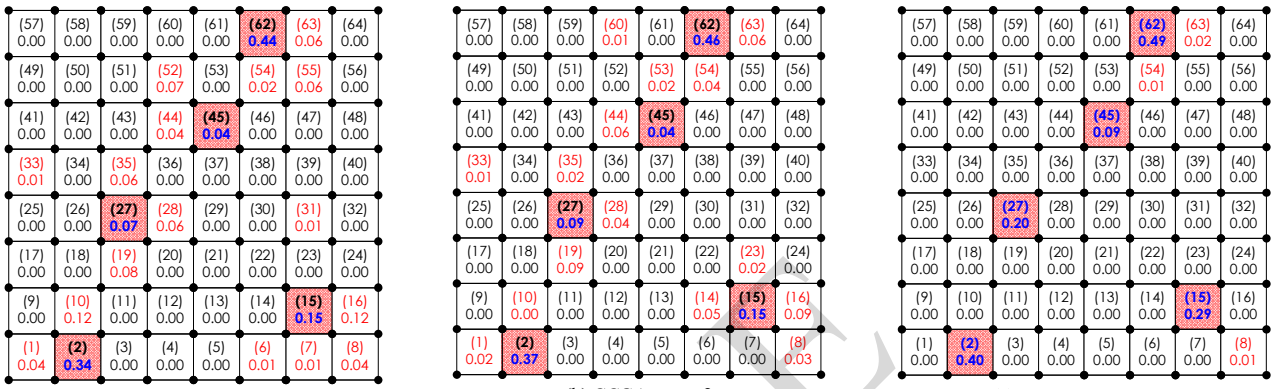
IJARE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)



(a) GA - case 1 (b) CCGA - case 1 (c) CCGADSS - case 1

Fig 6. Solutions obtained in case 1



(a) GA - case 2 (b) CCGA - case 2 (c) CCGADSS - case 2

Fig 7. Solutions obtained in case 2

Since CCGA employs the static decomposition strategy, only one species size used in this paper which is 1. On the other hand CCGA with dynamic species-size (CCGADSS) uses a number of species size so that species size set **SS** = {1, 2, 4, 8, 16, 32, 64}. Fig. 5 shows division of elements in each of the species size used in CCGADSS. The crossover method used in this paper is Simulated-binary crossover (SBX) [18] with the probability of 1. This paper employs the variable-wise polynomial mutation [19]. The probability of mutation is equal to the result of one divide the number decision variables on the chromosome to be performed in each optimization algorithm. The population size is set to 20 and Number of iterations used in each algorithm is 3200.

Error! Reference source not found. and **Error! Reference source not found.** show optimized solutions in case 1 and case 2 respectively. In case 1, the solutions by GA and CCGA in **Error! Reference source not found.** (a) and **Error! Reference source not found.** (b) are quite different from to the actual damage in **Error! Reference source not found.** (a). On the other hand, the solution obtained using CCGADSS in **Error! Reference source not found.** (c) is close to the actual damage occurred. For damaged elements 27, 28, 35, 36, which are all partially damaged 20%, damage factors of these elements by GA are 19%, 23%, 16%, 20%, the damage factors by CCGA are 23%, 15%, 21%, 19%, and the damage factors by CCGADSS are 21%, 19%, 20%, 20%. There is little errors in prediction of CCGADSS in only 3 other undamaged elements of which predicted damage factors are 1%. For the undamaged elements, the solution of CCGADSS is also more accurate than those of GA and CCGA. In case 2, the damage factors in the damaged elements 2, 15, 27, 45, 62 are 40%, 30%, 20%, 10%, 50% respectively as shown in **Error! Reference source not found.** (b). The solution obtained using GA in **Error! Reference source not found.** (a) shows that the damage occurred in the damaged elements are 34%, 15%, 7%, 4%, 44% which are quite different from the actual damage. It is found that there is incorrectly high damage in neighbor undamaged elements of the damaged elements such as elements 10, 16, 19, 28, 35, 52, 55, 63 in the solution by GA. In this case, solution by CCGA in **Error! Reference source not found.** (b) outperforms solution by GA, the damaged factors in the damaged elements are 37%, 15%, 9%, 4%, 46%. Similar to the solution obtained using GA, there is incorrectly high damage in some neighbor undamaged elements such as 14, 16, 19, 28, 44, 54, 63. In the contrast, the solution by CCGADSS in **Error! Reference source not found.** (c) shows that the damage factors in the damaged elements are 40%, 29%, 20%, 9%, 49% which are very close the actual damage.

IJARE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

VI. CONCLUSIONS

Vibration-based damage detection is based on the fact that vibration characteristics such as natural frequencies and mode shapes of structures are changed when the damage occurs. This paper employs cooperative coevolutionary genetic algorithm with dynamics species-size (CCGADSS) as the optimization algorithm in the vibration-based damage detection in plates.

The objective function is numerically calculated from the difference between experimentally vibration characteristics and numerically evaluated vibration characteristics of the predicted damage. There are 2 different cases with dissimilar occurred damage in plates are considered. In order to demonstrate the performance of the dynamic species-size strategy, 3 optimization algorithms, which are genetic algorithm (GA), cooperative coevolutionary genetic algorithm (CCGA), and CCGADSS. The results indicate that CCGADSS is superior to GA and CCGA. In addition, solutions obtained using CCGADSS are quite close the actual damage. The results show that the dynamic species-size strategy can enhance performance of cooperative coevolution concept.

REFERENCES

[1] J. E. Mottershead, M. I. Friswell, and C. Mares, "A method for determining model‐structure errors and for locating damage in vibrating systems," Meccanica. vol. 34, issue 3, pp. 153-166, 1999.

[2] Q. W. Yang, and J. K. Liu, "Structural damage identification based on residual force vector," Journal of Sound and Vibration. vol. 305, issue 1-2, pp. 298-307, 2007.

[3] S. K. Panigrahi, S. Chakraverty, and B. K. Mishra, "Vibration based damage detection in a uniform strength beam using genetic algorithm," Meccanica, vol. 44, issue 6, pp. 697-710, 2009.

[4] A. C. Altunışık, F. Y. Okur, and V. Kahya, "Vibration based damage detection in a uniform strength beam using genetic algorithm," Engineering Failure Analysis. vol. 79, pp. 154-170, 2017.

[5] F. L. M. dos Santos, B. Peeters, H. V. D. Auweraer, L. C. S. Góes, and W. Desmet, "Vibration-based damage detection for a composite helicopter main rotor blade," Case Studies in Mechanical Systems and Signal Processing. vol. 3, pp. 22-27, 2016.

[6] J. H. Holland, "Adaptation in natural and artificial systems," Ann Arbour, MI: University of Michigan Press, USA, 1975.

[7] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," Addison-Wesley, USA, 1989.

[8] M. A. Potter, and K. A. de Jong, "A cooperative coevolutionary approach to function optimization," Lecture Notes in Computer Science. vol. 866, pp. 249-257, 1994.

[9] M. A. Potter, and K. A. de Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," Evolutionary Computation. vol. 8, issue 1, pp. 1-29, 2000.

[10] K. Boonlong, N. Chaiyaratana, and S. Kuntanapreeda, "Optimal control of a hysteresis system by means of co-operative co-evolution," International Journal of Computational Intelligence and Applications. vol. 4, issue 4, pp. 321-336, 2004.

[11] R. Chandra, M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction," Neurocomputing. vol. 86, issue 1, pp. 116-123, 2012.

[12] O. Ibáñez, O. Cordón, S. Damas, "A cooperative coevolutionary approach dealing with the skull–face overlay uncertainty in forensic identification by craniofacial superimposition," Soft Computing. vol. 16, issue 5, pp. 797-808, 2012.

[13] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," Information Sciences. vol. 178, pp. 2985-2999, 2008.

[14] K. Boonlong, S. Jiamworanunkul, and K. Suksonghong, "Cooperative coevolutionary with dynamic species-size strategy for large-scale optimization problem," 2017 International Conference on Advances in Artificial Intelligence (ICAAI 2017), Bangkok, Thailand, April 3-5, 2017.

[15] D. J. Dawe, "Matrix and finite element displacement analysis and structures," Oxford engineering science series. pp. 410-541, 1984.

[16] Z. H. Ding, M. Huang, and Z. R. Lu, "Structural damage detection using artificial bee colony algorithm and hybrid search strategy," Swarm and Evolutionary Computation. vol. 28, pp. 1-13, 2016.

[17] X. Li, and X. Yao, "Cooperatively Coevolving Particle Swarms for Large Scale Optimization," IEEE Transactions on Evolutionary Computation. vol. 16, pp. 210-224, 2012.

[18] K. Deb, and R. B. Agrawal, "Simulated binary crossover for continuous search space," Complex Systems. vol. 9, issue 2, pp. 115-148, 1995.

[19] K. Deb, "Mechanical component design using genetic algorithms," Evolutionary Algorithms in Engineering Applications. pp. 495-512, 1997.

.

IJARE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)