International Journal of Control and Automation
Vol.9, No.1 (2016), pp.423-434
http://dx.doi.org/10.14257/ijca.2016.9.1.36

Chaotic Monkey Algorithm Based Optimal Sensor Placement

Zhen-rui Peng[*1], Hong Yin[1], An Pan[1] and Yu Zhao[2]

[1]*School of Mechatronics Engineering,*
Lanzhou Jiaotong University, China
[2]*School of Electronic Information and Electrical Engineering,*
Tianshui Normal University, China
**pengzr@mail.lzjtu.cn*

Abstract

Optimal sensor placement (OSP) techniques are of vital importance for the monitoring of large-scale structures or complex mechanical systems. In order to overcome the defects of slow convergence speed and easily falling into local optimum in monkey algorithm (MA), a new MA, called chaotic monkey algorithm (CMA), is proposed by introducing chaos search strategy to implement the OSP. In this algorithm, the initial monkeys are generated by using chaos variable and binary coding to enhance the global search capability, and a greedy strategy is adopted to improve the efficiency of local search. Numerical experiments are conducted on the sensor placement of a suspension bridge. The results verify that the new CMA can solve the OSP problem well and has better search capability than MA.

Keywords*: optimal sensor placement; monkey algorithm; chaos; modal assurance criterion*

1. Introduction

Due to an increasing number of complex mechanical systems or large structures for operating state monitoring or structural health monitoring, optimal sensor placement (OSP) in mechanical systems or structures has become a popular and common issue in the last decade. However, the performance of dynamic characteristics of these systems and structures depends heavily on the quantity and quality of the measured data, which in turn relies on the number of sensors and locations placed. Because of economic reasons, high cost of data acquisition systems and other reasons, the sensors installed in these structures or systems are always sparse, in fact, far less than available positions. The OSP techniques can reduce the chance of measuring and processing a large volume of redundant sensor data. Consequently, how to optimally place limited number of sensors for better structural identification and feature extraction is a challenging task [1-3].

A great deal of research has been conducted over the last decade on optimal sensor placement using a variety of placement techniques and criteria. It is not easy to classify all the OSP techniques without missing. Here, we categorize OSP techniques into two classifications — traditional techniques and non-traditional ones.

As for the traditional techniques, Kammer [4] developed the effective independence (EFI) method, which maximizes a combination of target mode signal strength and linear independence. EFI method classifies sensor locations based on the quantified information by observing target modes, and eliminates less significant locations from the candidates. Meo and Zumpano modified the EFI method, and proposed the effective independence driving-point residue (EFI-DPR) method for OSP to identify the vibration characteristics of a bridge [5]. Heo *et al*. proposed the modal kinetic energy (MKE) method to determine a sensor set that maximizes the KE of the system [6]. Li *et al*. combined EFI method with MKE method, raised a

ISSN: 2005-4297 IJCA

quick EFI method by QR decomposition, and demonstrated the connection between EFI and MKE on the I-40 Bridge [7].Carne and Dohrmann used the correlation of target mode shapes and defined the sensor set that minimizes the off-diagonal term of correlation matrix [8]. Papadimitriou *et al*. introduced the information entropy norm as a measure that best corresponds to the objective of structural testing which is to minimize the uncertainty in the model parameter estimates [9]. Lim employed the method based on a given rank for the system observability matrix that satisfies modal test constraints to determine sensor locations [10].

The non-traditional techniques mainly include some powerful heuristic and meta-heuristic techniques motivated by physics and biology, e.g. genetic algorithm (GA), simulated annealing, tabu search, monkey algorithm, particle swarm optimization, etc. [11]. Liu *et al*. introduced an improved genetic algorithm to find the optimal sensors placement for spatial lattice structure [12]. Yi *et al*. developed an improved genetic algorithm, called generalized genetic algorithm (GGA), to explore optimal placement of sensors [13]. Feng *et al*. [14] explored the use of GA in optimizing both the deployment and the modulated field of view (FOV) of the PIR sensors for improving the localization performance. Zhan *et al*. introduced Tabu search (TS) algorithm to solve the OSP problem in the field of the structural health monitoring and moving force identification [15]. Tong *et al*. [16] presented an improved simulated annealing (SA) algorithm, which can increase SA's random search performance while minimizing the computation efforts, to solve the sensor placement problem. Yi *et al*. developed a niching monkey algorithm (NMA) by combining the monkey algorithm (MA) with the niching techniques for sensor placement optimization [2]. Zhang *et al*. proposed an improved particle swarm optimization (IPSO) algorithm for the optimal sensor placement of latticed shell structure [17].

The monkey algorithm (MA), inspired by the mountain-climbing processes of monkeys, was firstly introduced by Zhao and Tang in [18]. Yi *et al*. adopted MA to solve the OSP for structural health monitoring [19, 20]. Zheng proposed an improved MA with dynamic adaptation [21]. MA consists of three main processes, namely, climb process, watch-jump process and somersault process. Climb process is used to explore the local optimal solution. Watch-jump process is designed to find out whether there are higher mountains around it when its own mountaintop is arrived. Somersault process makes the monkeys transfer to new search domains rapidly.

In this paper, the initial monkeys are generated by using chaos variable and binary coding. Because there is one-to-one correspondence between the binary coding and the entire solution set of OSP problem, the ergodicity of chaotic variable is put to full use. And a greedy strategy is adopted to improve the efficiency of local search.

The rest of the paper is organized as follows. In Section 2, the finite element model of a suspension bridge structure is presented and the mathematical model of the structure is also modeled. Section 3 is concerned with the implementation of CMA in details. And Section 4 analyzes the results by comparing CMA and MA methods for OSP. Finally, a conclusion is made for this paper.

2. Mathematical Model of A Suspension Bridge

The OSP is a typical combinatorial optimization problem. The objective of the optimization problem is to minimize the number of sensors and to locate them properly for the quality estimation of target dynamic modes. The optimal number of sensors and their locations are expected to simultaneously produce the minimum sensor management cost as well as accurate estimation of structural modal parameters [3]. Essentially, the first task of this optimization problem is to analyze the structure and determine the objective function for achieving the optimal solution.

2.1. Finite Element Model

As one of the typical complex structures, a suspension bridge is selected for the example of the OSP. Reinforced concrete stiffening truss system is adopted in the suspension bridge. Reinforced concrete is the material of the main tower. H tower is employed in the transverse bridge. Reinforced concrete truss is applied in stiffening girder. ANSYS13. 0 is used to establish finite element model (FEM) of the bridge. SHELL63 element is used in the bridge panel. BEAM4 unit is used in stiffening girder and bridge tower. LINK10 unit is used in main cable and sling. The FEM of the bridge is shown in Figure 1.

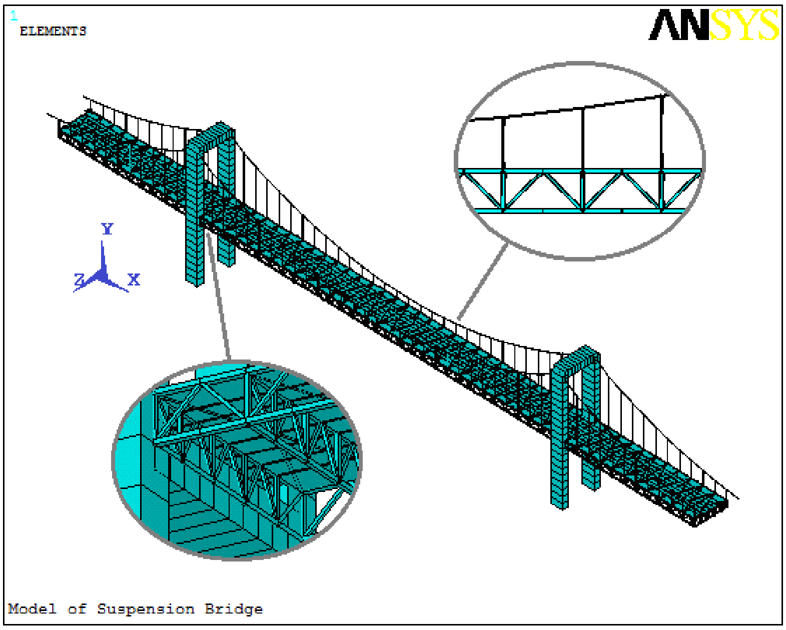
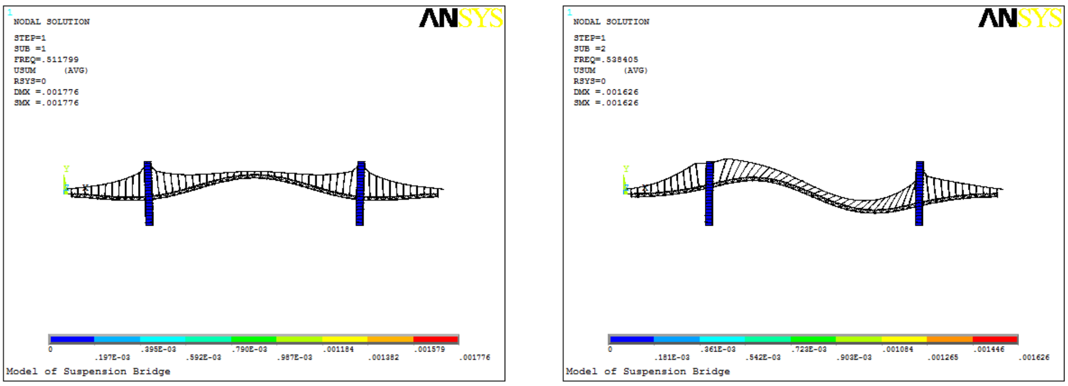


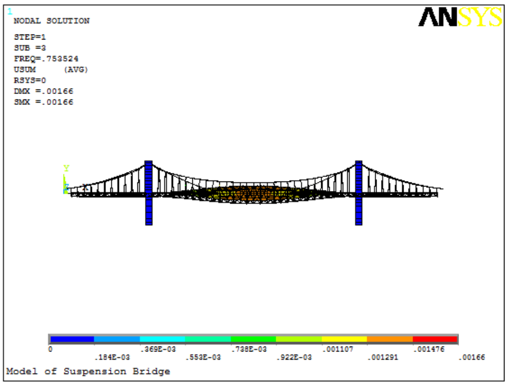
Figure 1. FEM of the Suspension Bridge

To obtain the mode shapes data of the suspension bridge, modal analysis is conducted on the FEM of the bridge. Considering the low order modes having larger coefficients, the first 9 mode shapes are extracted, which are shown in Figure 2. And the first 9 modal frequencies are calculated. The first 9 modal frequencies and mode shape characteristics are given in Table 1.

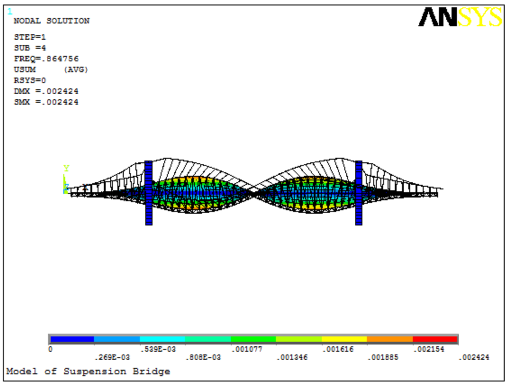


(a) The first mode shape

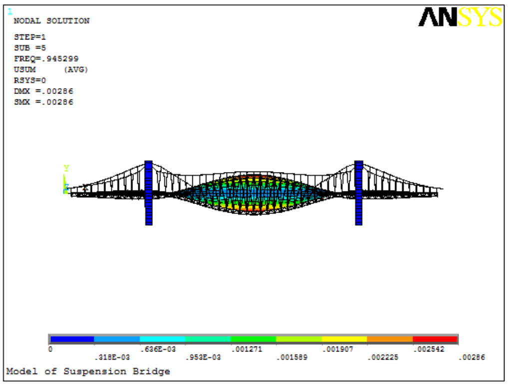
(b) The second mode shape



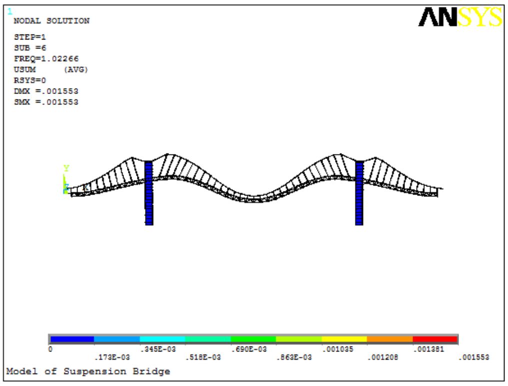
(c) The third mode shape



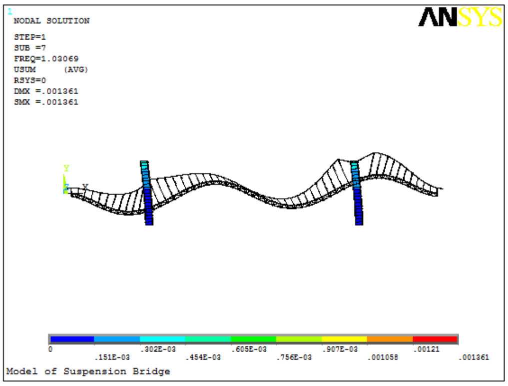
(d) The fourth mode shape



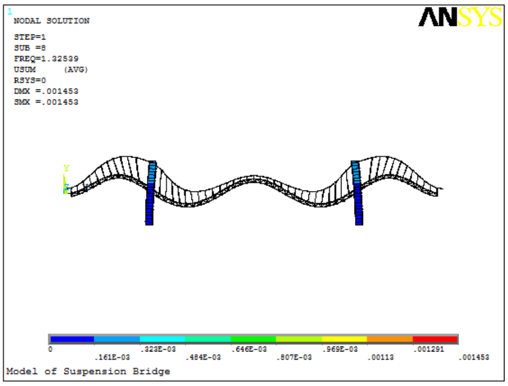
(e) The fifth mode shape



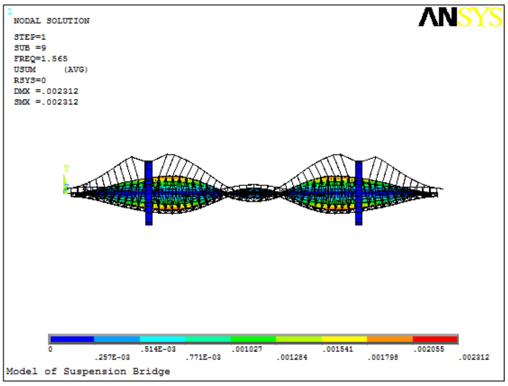
(f) The sixth mode shape



(g) The seventh mode shape



(h) The eighth mode shape



(i) The ninth mode shape

Figure 2. The First 9 Mode Shapes of the Bridge

Table 1. The First 9 Modal Frequencies and Mode Shape Characteristics

Order number	Frequency /Hz	Mode shape characteristic
1	0.5118	Symmetric lateral bending of main beam
2	0.5384	Anti-symmetric lateral bending of main beam
3	0.7535	Symmetric torsion of main beam
4	0.8647	Symmetric torsion of main beam
5	0.9456	Symmetric torsion of main beam
6	1.0227	Symmetric lateral bending of main beam
7	1.0307	Anti-symmetric lateral bending of main beam
8	1.3254	Symmetric lateral bending of main beam
9	1.5650	Symmetric torsion of main beam

2.2. Mathematical Model

Denote $\boldsymbol{\Phi}_{n\times m}$ the modal matrix, which consists of the data from modal analysis, where n is the number of degrees of freedom (DOFs); m is the mode order. We need to select s DOFs from n DOFs as the final sensor locations. Modal assurance criterion (MAC), which is a commonly used criterion [3], is applied to evaluate the correlation between the modal vectors so that the measured modal vectors can be easily distinguished. The maximum off-diagonal element of ***MAC*** is selected as the objective function of the OSP. The mathematical model can be expressed as follows.

$$f(\boldsymbol{\Phi}_{s\times m}) = \max\left\{ \underset{i\neq j}{\boldsymbol{MAC}_{ij}} = \frac{(\boldsymbol{\phi}_i^T\boldsymbol{\phi}_j)^2}{(\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i)(\boldsymbol{\phi}_j^T\boldsymbol{\phi}_j)} \right\} \qquad (1)$$

where $0 \le f(\boldsymbol{\Phi}_{s\times m}) \le 1$; $\boldsymbol{\Phi}_{s\times m} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ \cdots\ \boldsymbol{\phi}_m]$; $\boldsymbol{\Phi}_{s\times m}$ denotes the modal matrix of a solution with s DOFs in placement problem; $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ represent the i th and j th column vectors in matrix $\boldsymbol{\Phi}_{s\times m}$, respectively; and the superscript T denotes the transpose of the vector. In equation (1), if the off-diagonal element $\boldsymbol{MAC}_{ij}\ (i \ne j)$ tends to zero, it indicates that there is little correlation between the modal vector $\boldsymbol{\phi}_i$ and the modal vector $\boldsymbol{\phi}_j$, that is to say, the modal vector can be distinguished easily.

3. Chaos Monkey Algorithm

Although the MA can successfully find optimal or near-optimal solutions to the optimization problems with large dimensions and a huge number of local optima, the values of random variable in initialization process, climb process, watch-jump process and somersault process cannot guarantee the ergodicity [2-21]. Inspired by the chaos optimization algorithm [22], similar to those in Refs. [2-21], the idea of chaotic search strategy is introduced to monkey algorithm, which is called chaotic monkey algorithm (CMA), to avoid repeating search in the same domains and falling into local optimum.

In this study, the chaotic variable is used to replace the random variable in CMA, and the binary coding is adopted for the OSP multivariable problems. The details of CMA are described in the following sections.

3.1. Binary Coding

Binary coding is intuitive and convenient for the OSP, so it is adopted for the optimal sensor placement problem. Let n denote the initial number of candidate sensors and s denote the final number of sensors determined. The binary coding for a solution can be expressed as $\boldsymbol{X} = (x_0, x_1, ..., x_j, ..., x_n)^T$, $j \in \{1, 2, \cdots, n\}$. If x_j is 1, there is a sensor on that DOF. If x_j is 0, there is no sensor on that DOF. And the condition $\sum_{j=1}^{n} x_j = s$ is satisfied.

The conversion between binary coding $\boldsymbol{X}$ and modal matrix $\boldsymbol{\Phi}_{s\times m}$ is the process of removing the rows of $\boldsymbol{\Phi}_{n\times m}$. In this process, remove those rows of $\boldsymbol{\Phi}_{n\times m}$ corresponding to those $\{x_j \mid x_j = 0, j \in [1, 2, \cdots, n]\}$ in $\boldsymbol{X}$. For convenience, this process is represented by equation (2).

$$\boldsymbol{\Phi}_{s\times m} = g(\boldsymbol{\Phi}_{n\times m}, \boldsymbol{X}_{n\times 1}) \qquad (2)$$

where $\boldsymbol{\Phi}_{s\times m}$ is the modal matrix of a solution with s sensors in OSP; $\boldsymbol{X}_{n\times 1}$ is the binary coding corresponding to a solution.

For a simple example, substituting $n = 4$, $m = 2$, $s = 3$ and $\boldsymbol{X}_{4\times 1} = (1, 0, 1, 1)^T$ into equation (2) , and take $\boldsymbol{\Phi}_{4\times 2} = \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$ as the input modal matrix of function g , the output modal matrix of g is $\boldsymbol{\Phi}_{3\times 2} = g\left(\begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 & 5 \\ \not{2} & \not{6} \\ 3 & 7 \\ 4 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$ can be easily obtained by deleting the 2nd row of $\boldsymbol{\Phi}_{4\times 2}$.

3.2. Initialization

The characteristics of ergodicity and randomness of chaotic variable can make the chaos optimization algorithm jump out of local optimum, and speed up the search process. In order to avoid falling into local optimum, chaotic search strategy, instead of random process, is introduced to determine the values of random variable of monkey algorithm.

Here, the well-known logistic map is adopted:

$$c_{i+1} = 4 \cdot c_i \cdot (1 - c_i) \quad (3)$$

where c_i is chaotic variable; $0 < c_i < 1$; $c_0 \notin \{0, 0.25, 0.75, 1\}$.

The binary coding of monkey is generated by using Algorithm 1. This algorithm is executed N times when the size of monkeys' population is N . In the implementation of Algorithm 1, the order of bits is changed, but the number of 0 and the number of 1 are not changed. So there is a one-to-one correspondence between the positions of monkeys (binary coding set) and the solution set of OSP.

3.3 Climb Process

Climb process is used to explore the local optimal solution. In other words, climb process can generate a better position than previous one by changing some bits of current position $\boldsymbol{X}$.

Algorithm 1 Chaotic Shuffle

Input: The number of candidate DOFs *n* and the number of sensors *s*.

Set $\boldsymbol{X} = (\underbrace{1,1,\ldots,1}_{s}, \underbrace{0,0,\ldots,0}_{n-s})^T$

For i from n to 1:

Set $j = 1 + c \times i$, where c is chaotic variable, and $c \in (0,1)$.

Set j is the *nearest* integer not greater than i .

Swap ($\boldsymbol{X}(i)$, $\boldsymbol{X}(j)$).

Return $\boldsymbol{X}$

Output: The chaotic coding of a solution $\boldsymbol{X}$.

In order to raise the efficiency of climb process, a kind of greedy local search strategy is adopted. The best DOF is selected in each step. The entire process is executed N_c times for each monkey as depicted in Algorithm 2.

Algorithm 2 Climb Process

Input: The position of a monkey $\boldsymbol{X} = (x_0, x_1, \ldots, x_j, \ldots, x_n)^T$.

Set $\boldsymbol{X}_{\text{best}} = \text{null}$

For j from 1 to n :

If $x_j = 0$ then

Set $x_j = 1$

If $\boldsymbol{X}_{\text{best}} = \text{null}$ or $f(g(\boldsymbol{\Phi}_{n\times m}, \boldsymbol{X})) < f(g(\boldsymbol{\Phi}_{n\times m}, \boldsymbol{X}_{\text{best}}))$ then

Set $\boldsymbol{X}_{\text{best}} = \boldsymbol{X}$

Set $x_j = 0$

If $\boldsymbol{X}_{\text{best}} = \text{null}$ then

Return null

Set $\boldsymbol{X} = \boldsymbol{X}_{\text{best}}$ (Note: there are $s + 1$ $x_j = 1$ in current $\boldsymbol{X}$)

Set $\boldsymbol{X}_{\text{best}} = \text{null}$

For j from 1 to n :

If $x_j = 1$ then

Set $x_j = 0$

If $\boldsymbol{X}_{\text{best}} = \text{null}$ or $f(g(\boldsymbol{\Phi}_{n\times m}, \boldsymbol{X})) < f(g(\boldsymbol{\Phi}_{n\times m}, \boldsymbol{X}_{\text{best}}))$ then

Set $\boldsymbol{X}_{\text{best}} = \boldsymbol{X}$

Set $x_j = 1$
If $\boldsymbol{X}_{\text{best}} = \text{null}$ then
Return null
Return $\boldsymbol{X}_{\text{best}}$
Output: A better position $\boldsymbol{X}_{\text{best}}$ than previous one.

Step 1: A DOF from the $n - s$ DOFs with no sensor is determined by the value of objective function. For example, the DOF leading to the smallest value of objective function is chosen. After a sensor is placed on the DOF, the number of sensors is increased form s to $s + 1$, and the modal matrix is changed from $\Phi_{s\times m}$ to $\boldsymbol{\Phi}_{(s+1)\times m}$.

Step 2: A DOF from the $s + 1$ DOFs with sensor is determined by the value of objective function. For example, the DOF leading to the smallest value of objective function is chosen. After a sensor is removed from the DOF, the number of sensors is decreased form $s + 1$ to s, and the modal matrix is changed from $\boldsymbol{\Phi}_{(s+1)\times m}$ to $\boldsymbol{\Phi}_{s\times m}$.

In the end of Algorithm 2, the number of 0 and the number of 1 are not changed. So, there is a one-to-one correspondence between the positions of monkeys (binary coding set) and the solution set of OSP.

Applying the method by Carne and Dohrmann [8], the maximum off-diagonal element of ***MAC*** is calculated efficiently by only adding or deleting one DOF. So the efficiency of Algorithm 2 is significantly improved. The procedure of the method, depicted in formula (4), is used to add one sensor at a time.

$$\left(\boldsymbol{MAC}_{ij}\right)_k = \frac{(a_{ij} + \phi_{ki}\phi_{kj})(a_{ij} + \phi_{ki}\phi_{kj})}{(a_{ii} + \phi_{ki}\phi_{ki})(a_{jj} + \phi_{kj}\phi_{kj})} \tag{4}$$

where a_{ij} are the elements of A=$\boldsymbol{\Phi}^T_{s\times m}\boldsymbol{\Phi}_{s\times m}$; ϕ_{ki} and ϕ_{kj} are the elements of $\boldsymbol{\Phi}_{n\times m}$.

3.4 Watch-Jump Process

The purpose of watch-jump process is designed to find out whether there are higher mountains around it and when its own mountaintop is arrived. For a CMA with binary representation, the watch-jump process is generally performed by independently and chaotically swapping two bits with different values (e.g. one is 1, another is 0). The entire process is executed N_w times for each monkey as depicted in Algorithm 3. In the implementation of Algorithm 3, the number of 0 and the number of 1 are not changed. So there is a one-to-one correspondence between the positions of monkeys (binary coding set) and the solution set of OSP.

Algorithm 3 Watch-jump Process

Input: The position of a monkey $\boldsymbol{X} = (x_0, x_1, \ldots, x_j, \ldots, x_n)^T$.
Set $\boldsymbol{X}_{\text{best}} = \boldsymbol{X}$
Random select x_a from $\{x_j \mid x_j = 1, j \in [1, n]\}$
Random select x_b from $\{x_j \mid x_j = 0, j \in [1, n]\}$
Set $x_a = 0$ and $x_b = 1$
If $f(g(\boldsymbol{\Phi}_{n\times m}, \boldsymbol{X})) < f(g(\boldsymbol{\Phi}_{n\times m}, \boldsymbol{X}_{\text{best}}))$ is better than previous one then
Set $\boldsymbol{X}_{\text{best}} = \boldsymbol{X}$
Output: A better position $\boldsymbol{X}_{\text{best}}$ than previous one.

3.5 Somersault Process

Somersault process makes the monkeys transfer to new search domains rapidly. If, after a monkey exhausts its neighbor position, no one is better than its current position, then the somersault process is executed to generate a new position.

For all monkeys $\boldsymbol{X}_i = (x_{i0}, x_{i1}, \ldots, x_{ij}, \ldots, x_{in})^T$, $i \in \{1, 2, \cdots, N\}$, the somersault process is as follows:

(1) The coordinate of monkeys' center $\boldsymbol{X}_{\text{center}}$ is calculated, and the top 10 bits of $\boldsymbol{X}_{\text{center}}$ are found.

(2) A bit is chaotically selected from the top 10 bits, which is always set to 1 for monkeys who need to execute the somersault process.

(3) The process similar to Algorithm 1 is performed for each monkey to chaotically generate other $s-1$ bits.

The entire process is depicted in Algorithm 4. At the end of Algorithm 4, the number of 1 is s in binary coding. So there is a one-to-one correspondence between the positions of monkeys (binary coding set) and the solution set of OSP.

Algorithm 4 Somersault Process

Input: The positions of all monkeys $\boldsymbol{X}_i = (x_{i0}, x_{i1}, \ldots, x_{ij}, \ldots, x_{in})^T$, $i \in \{1, 2, \cdots, N\}$.

Set $\boldsymbol{X}_{\text{center}} = \frac{1}{N}\left(\sum_{i=1}^{N} x_{i1}, \sum_{i=1}^{N} x_{i2}, \ldots, \sum_{i=1}^{N} x_{ij}, \ldots, \sum_{i=1}^{N} x_{in}\right)^T$

Chaotically select a k from $\left\{k \mid k \in \underset{j\in\{1,2,\cdots,n\}}{\arg\text{top}10}(\boldsymbol{X}_{\text{center}}(j))\right\}$

For each monkey:

Chaotically generate a new position and guarantee $\boldsymbol{X}_{\text{new}}(k) = 1$

Set $\boldsymbol{X} = \boldsymbol{X}_{\text{new}}$

Output: The positions of all monkeys $\boldsymbol{X}_i = (x_{i0}, x_{i1}, \ldots, x_{ij}, \ldots, x_{in})^T$, $i \in \{1, 2, \cdots, N\}$.

The termination condition of CMA is determined by $N_{\max}$ (the maximum number of iterations). If the termination condition is satisfied after the somersault process, then the CMA outputs the best solution that appears in the whole process of CMA; otherwise, go to the climb process. It is also important to note that the best solution should be updated at each step of CMA.

The flow chart about above processes is shown in Figure 3.

4. Analysis of the Results

In order to show the effectiveness of CMA for OSP problem, the result of sensor placements by CMA is compared with that by MA. The original data is generated by the FEM of the suspension bridge as described in section 2.1. The modal matrix is obtained by processing the original data. The objective function of OSP problem is $f(\boldsymbol{\Phi}_{s\times m})$ which is defined in section 2.2.

The problem of parameters selection in CMA for OSP is studied. After trial and error, a group of optimal parameters are determined. The maximum number of iteration $N_{\max} = 50$, the maximum number of allowed climb process $N_c = 30$, the maximum number of allowed watch-jump process $N_w = 30$, the size of monkeys' population $N = 5$. In order to guarantee the effectiveness of comparison between CMA and MA, the same parameters are set to both of the two algorithms.

For the purpose of test convenience, the nodes are selected only from the nodes on the edges of bridge. There are 107 nodes on each edge. The total number of nodes is 214. The displacements of nodes on x-axis are so weak that it is difficult to measure, so the displacements of nodes on y-axis and z-axis are considered as the candidates of DOFs. In other words, the number of the candidate DOFs is $214 \times 2 = 428$. Suppose there are 18 sensors being used to measure the displacements, the specific problem now is that how to minimize the objective function $f(\boldsymbol{\Phi}_{s \times m})$. Here, $\boldsymbol{\Phi}_{s \times m} = \boldsymbol{\Phi}_{18 \times 9}$, where the rows are selected from the rows of $\boldsymbol{\Phi}_{428 \times 9}$; 9 represents the number of mode shapes needed to measure.

The difference between CMA and MA is search strategy and coding format. In CMA, the initial monkeys are generated by using chaos variable and binary coding to enhance the global search capability, and a greedy strategy is adopted to improve the efficiency of local search. In MA, the initial monkeys are generated by using random variable and decimal coding reduces the global search capability, and random search strategy is adopted in climb process.

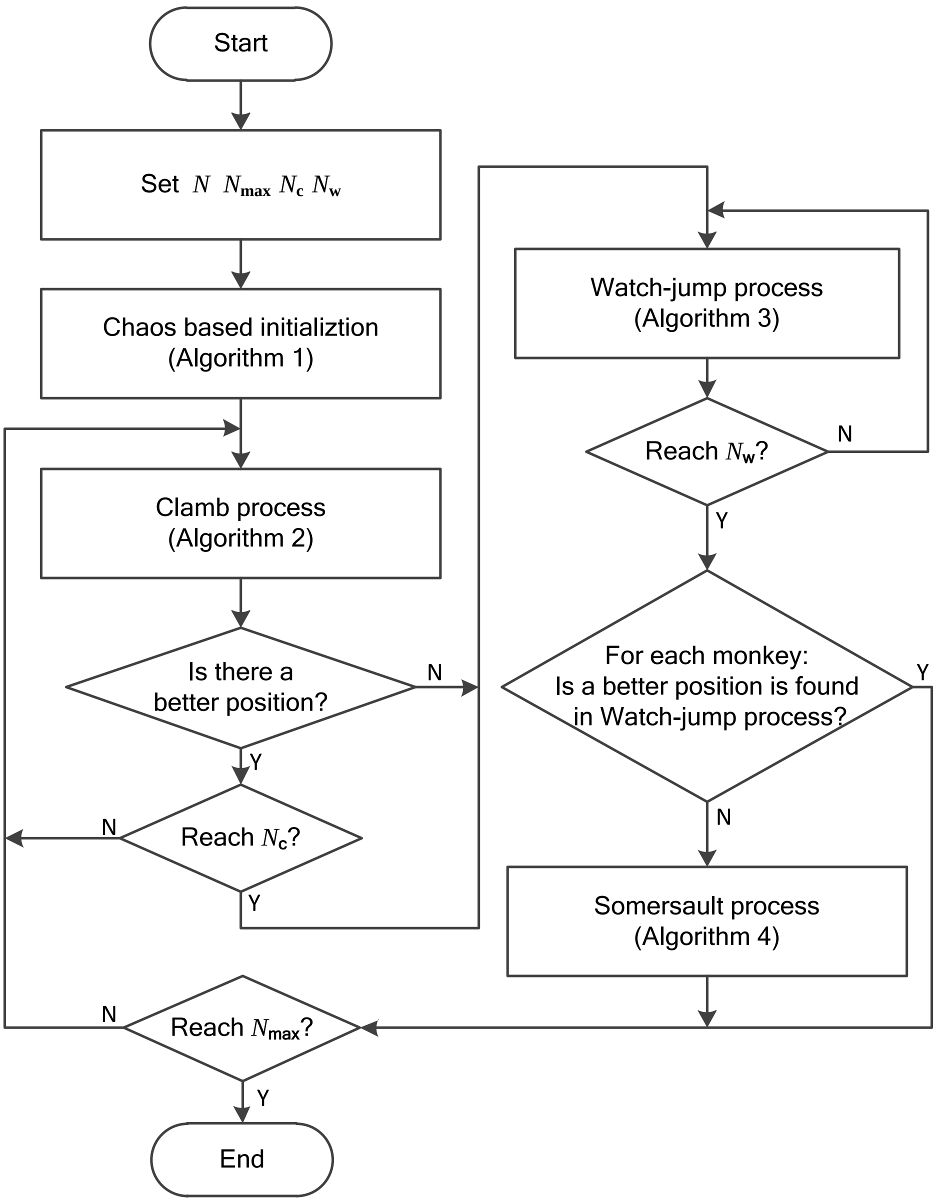


Figure 3. Flow Chart of CMA for OSP Problem

MA and CMA for OSP are implemented by using g++ 4.8.1 compiler. The statistical results are listed in Table 2. As illustrated in Table 2, the mean, standard deviation, minimum and maximum by CMA are less than those by MA, and the average run time of CMA is slightly longer than that of MA. Figure 4 shows that the sensors' placement by CMA.

Table 2. Statistical results for OSP by using CMA and MA methods

Method	Minimum	Mean	Maximum	Std. dev.	Ave.Time (s)
MA	0.032394	0.044347	0.069189	0.011050	1.318000
CMA	0.021615	0.034692	0.043806	0.006668	1.466400

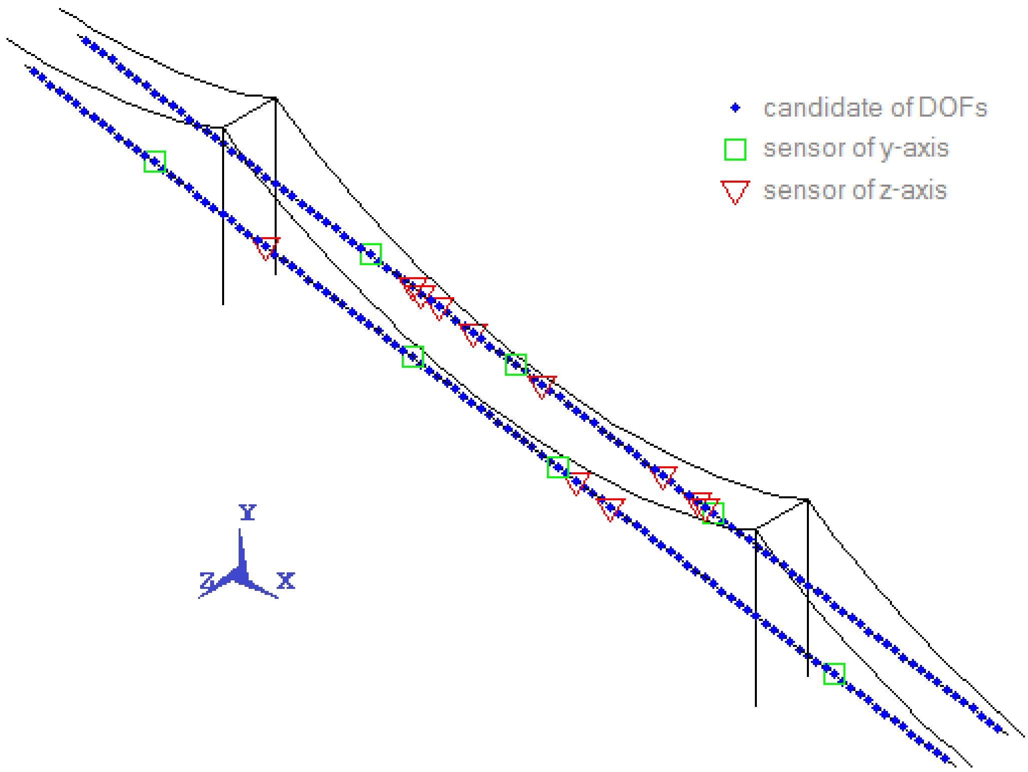


Figure 4. Placement of Sensors by CMA

5. Conclusions

In this paper, a new MA, called chaotic monkey algorithm (CMA) is presented. And then this CMA is adopted to solve the optimal sensor placement problem. Unlike the traditional MA, this method is improved by introducing chaotic searching strategy and using binary encoding. And this study takes a suspension bridge to verify the proposed approach. Modal analysis is carried out to extract the first 9 order mode shapes. The maximum off-diagonal element of ***MAC*** is selected as the objective function of OSP problem. Initialization, climb process, watch-jump process and somersault process are described as means for searching the optimal solution. The sensors' placement result is influenced by the parameters of the algorithm which are determined after trial and error. The advantage of CMA in the OSP problem is shown by comparing the statistical results of CMA and MA.

Acknowledgements

This research is supported by National Natural Science Foundation of China (No. 61463028), Fundamental Research Funds for Colleges and Universities in Gansu Province (No. 213054), and Research Project of Gansu Education Department (No. 213027). The authors would like to thank the reviewers for their helpful suggestions. The authors are grateful to Dr. Yong-Kui CHANG for his kind suggestions.

References

[1] D.S. Li, H.N. Li and C.P. Fritzen, "Load dependent sensor placement method: Theory and experimental validation", Mechanical Systems and Signal Processing, vol. 31, **(2012)**, pp. 217-227.
[2] T.H. Yi, H.N. Li and M. Gu, "Sensor Placement Optimization in Structural Health Monitoring Using Niching Monkey Algorithm ", International Journal of Structural Stability and Dynamics, **(2014)**.
[3] M. Chang and S.N. Pakzad, "Optimal sensor placement for modal identification of bridge systems considering number of sensing nodes", Journal of Bridge Engineering, vol. 9, no. 6, **(2014)**.
[4] D.C. Kammer, "Sensor placement for on-orbit modal identification and correlation of large space structures [J]", Journal of Guidance, Control, and Dynamics, vol. 14, no. 2, **(1991)**, pp. 251-259.
[5] M. Meo and G. Zumpano, "On the optimal sensor placement techniques for a bridge structure", Engineering Structures, vol. 27, no. 10, **(2005),** pp. 1488-1497.
[6] G. Heo, M.L. Wang and D. Satpathi, "Optimal transducer placement for health monitoring of long span bridge [J]", Soil dynamics and earthquake engineering, vol. 16, no. 7, **(1997)** pp. 495-502.
[7] D.S. Li, H.N. Li and C.P. Fritzen, "The connection between effective independence and modal kinetic energy methods for sensor placement", Journal of sound and vibration, vol. 305, no. 4, **(2007)**, pp. 945-955.
[8] T.G. Carne and C.R. Dohrmann, "A modal test design strategy for model correlation", Proceedings of the 13th International Modal Analysis Conference, **(1995)**, pp. 927-927.
[9] C. Papadimitriou and G. Lombaert, "The effect of prediction error correlation on optimal sensor placement in structural dynamics", Mechanical Systems and Signal Processing, vol. 28, **(2012)**, pp. 105-127.
[10] K.B. Lim, "Method for optimal actuator and sensor placement for large flexible structures", Journal of Guidance, Control, and Dynamics, vol. 15, no. 1, **(1992)**, pp. 49-57.
[11] L. Yao, W.A. Sethares and D.C. Kammer, "Sensor placement for on-orbit modal identification via a genetic algorithm [J]", AIAA journal, vol. 31, no. 10, **(1993)**, pp. 1922-1928.
[12] W. Liu, W. Gao and Y. Sun, "Optimal sensor placement for spatial lattice structure based on genetic algorithms [J]", Journal of Sound and Vibration, vol. 317, no. 1, **(2008)**, pp. 175-189.
[13] T.H. Yi, H.N. Li and M. Gu, "Optimal sensor placement for health monitoring of high-rise structure based on genetic algorithm", Mathematical Problems in Engineering, **(2011)**.
[14] G. Feng, M. Liu and G. Wang, "Genetic algorithm based optimal placement of PIR sensors for human motion localization", Optimization and Engineering, **(2013)**, pp. 1-14.
[15] J.Z. Zhan and L. Yu, "Optimal Sensor Placement Based on Tabu Search Algorithms", Applied Mechanics and Materials, vol. 578, **(2014)**, pp. 1069-1072.
[16] K.H. Tong, N. Bakhary, A.B.H. Kueh and A.Y. Mohd Yassin, "Optimal sensor placement for mode shapes using improved simulated annealing", Smart Structures and Systems, vol. 3, **(2014)**pp. 389-406.
[17] X. Zhang, J. Li and J. Xing, "Optimal Sensor Placement for Latticed Shell Structure Based on an Improved Particle Swarm Optimization Algorithm", Mathematical Problems in Engineering, **(2014)**.
[18] R. Zhao and W. Tang, "Monkey algorithm for global numerical optimization", Journal of Uncertain Systems, vol. 2, no. 3, **(2008)**, pp. 165-176.
[19] T.H. Yi, H.N. Li and X.D. Zhang, "Sensor placement on Canton Tower for health monitoring using asynchronous-climb monkey algorithm", Smart Materials and Structures, vol. 21, no. 12, **(2012)**, pp. 125-023.
[20] T.H. Yi, H.N. Li and X.D, Zhang, "A modified monkey algorithm for optimal sensor placement in structural health monitoring", Smart Materials and Structures, vol. 21, no. 10, **(2012)**, pp. 105-033.
[21] L. Zheng, "An improved monkey algorithm with dynamic adaptation", Applied Mathematics and Computation, vol. 222, **(2013)**, pp. 645-657.
[22] B.L.I.W. JIANG, "Optimizing complex functions by chaos search", Cybernetics & Systems, vol. 29, no. 4, **(1998)**, pp. 409-419.