Struct Multidisc Optim
DOI 10.1007/s00158-011-0754-4

RESEARCH PAPER

Chaotic imperialist competitive algorithm for optimum design of truss structures

Siamak Talatahari · Ali Kaveh · Razi Sheikholeslami

Received: 6 March 2011 / Revised: 10 November 2011 / Accepted: 17 December 2011

Abstract The imperialist competitive algorithm is a new socio-politically motivated optimization algorithm which recently is applied for structural problems. This paper utilizes the idea of using chaotic systems instead of random processes in the imperialist competitive algorithm. The resulting method is called chaotic imperialist competitive algorithm (CICA) in which chaotic maps are utilized to improve the movement step of the algorithm. Some well-studied truss structures are chosen to evaluate the efficiency of the new algorithm.

Keywords Imperialist competitive algorithm · Chaos · Meta-heuristic algorithms · Truss structures · Optimum design

1 Introduction

Structural optimization has become one of the most active branches of structural engineering in the last decades (Kaveh et al. 2008) and meta-heuristic optimization techniques provide efficient tools to reach the optimum design of structures. Genetic algorithms, particle swarm optimization (Eberhart and Kennedy 1995), ant colony optimization (Dorigo et al. 1996), harmony search algorithm (Geem et al. 2001), charged system search (Kaveh and Talatahari 2010a), and imperialist competitive algorithm (Atashpaz-Gargari and Lucas 2007) are some familiar examples of meta-heuristic algorithms.

Characterizing the irregular behavior that can be caused either by deterministic chaos or by stochastic processes is not an easy task to perform and it is still an open problem to distinguish among these two types of phenomena. However, the interest in studying the use of chaotic systems instead of random ones arises when the theme of chaos reaches a high interdisciplinary level involving not only mathematicians, physicians and engineers but also biologists, economists and scientists from different areas (Chen and Dong 1998; Antoniou et al. 2003; Harb and Abdel-Jabbar 2003). One of these fields is based on the idea of using chaotic systems for stochastic optimization algorithms.

Although chaos and random signals share the property of long term unpredictable irregular behavior and many of random generators in programming softwares as well as the chaotic maps are deterministic; however chaos can help order to arise from disorder. Similarly, nature-inspired optimization algorithms are inspired from biological systems where order arises from disorder. In these cases disorder often indicates both non-organized patterns and irregular behavior, whereas order is the result of self-organization and evolution and often arises from a disorder condition or from the presence of dissymmetries. Self-organization and evolution are two key factors of many stochastic optimization techniques. Due to these common properties between chaos and optimization algorithms, simultaneous use of these concepts may improve the performance. Experimental studies show the benefits of such combination; although, this is not mathematically proved yet (Tavazoei and Haeri 2007).

S. Talatahari
Marand Faculty of Engineering, University of Tabriz, Tabriz, Iran

A. Kaveh (✉)
Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran
e-mail: alikaveh@iust.ac.ir

R. Sheikholeslami
Department of Civil Engineering, Amirkabir University of Technology, Tehran, Iran

Published online: 31 January 2012

Chaotic sequences have been shown to be easy and fast to generate and store, and there is no need for storing long sequences. Merely a few functions (chaotic maps) and few parameters (initial conditions) are needed even for very long sequences. In addition, an enormous number of different sequences can be generated simply by changing its initial condition. Moreover these sequences are deterministic and reproducible (Alatas 2010).

This paper presents chaotic imperialist competitive algorithm (CICA) to determine optimum design of truss structures. Original imperialist competitive algorithm (ICA) is a socio-politically motivated optimization algorithm. Each individual agent of an empire is called a country, and the countries are categorized into colony and imperialist states that collectively form empires. Imperialistic competitions among these empires form the basis of the ICA which directs the search process toward the powerful imperialist or the optimum points. Kaveh and Talatahari improved the ICA by defining two new movement steps and investigated the performance of this algorithm to optimize the design of skeletal structures (Kaveh and Talatahari 2010b, c). This algorithm is called orthogonal imperialist competitive algorithm (OICA). In the proposed algorithm (CICA), we use different chaotic systems with substitute random number generators for different parameters of the ICA.

Several design examples are tested using the new method, and the results reveal that the improvement of the present algorithm is due to the application of deterministic chaotic signals in place of random sequences. The remaining sections of this paper are organized as follows:

Review of ICA is presented in Section 2. In Section 3, we introduce the proposed method which is called chaotic imperialist competitive algorithm (CICA). The CICA method for optimal design of trusses is provided in Section 4. Section 5 contains several illustrative examples, and Section 6 concludes the paper.

2 Imperialist competitive algorithm

2.1 Standard imperialist competitive algorithm

Meta-heuristics are designed to tackle complex optimization problems where other optimization methods have failed to be either effective or efficient. These methods have come to be recognized as one of the most practical approaches for solving many complex problems, and this is particularly true for the many real-world problems that are combinatorial in nature. Although some algorithms perform better on some design problems than others and no one algorithm performs best on all problems, the practical advantage of meta-heuristics lies in both their effectiveness and more or less general applicability (Ólafsson 2006). The ICA is one of these meta-heuristic optimization techniques which simulates the social political process of imperialism and imperialistic competition. This algorithm is a population based process in which each individual of the population is called a "country". Some of the best countries (in optimization terminology, countries with lower cost) are selected to be the "imperialist" states and the remaining countries form the "colonies" of these imperialists. All the colonies of initial countries are divided among the imperialists based on their "power".

In the ICA, the assimilation policy, pursued by some of former imperialist states, is modeled by moving all the colonies toward the imperialist. The total power of an empire depends on both the power of the imperialist country and the power of its colonies. Then the imperialistic competition begins among all the empires. Any empire that is not able to succeed in this competition and cannot increase its power (or at least prevent losing its power) will be eliminated from the competition. The imperialistic competition will gradually result in an increase in the power of the powerful empires and a decrease in the power of weaker ones. Weak empires will loose their power and ultimately they will collapse. The movement of colonies, competition among empires and the collapse mechanism direct the optimization process toward an optimum point.

The agents or countries (with the number of $N_{country}$) are divided into two types; the best ones or imperialist states (with the number of N_{imp}) and the colonies (with the number of N_{col}). The initial countries are generated randomly as:

$$\{x\}_i = \{x_{\min}\} + \{rand\} \otimes \{x_{\max} - x_{\min}\} \quad (1)$$

in which *min* and *max* represent the lower and upper bounds for variable vector $\{x\}$, respectively. The sign "$\otimes$" denotes an element-by-element multiplication. $\{rand\}$ is a random vector.

In this paper, 10 percent of countries are considered as empires and the remaining is used as colonies. In order to divide the colonies of initial countries among the imperialists, the power of each country is defined inversely proportional to its cost which is calculated considering the related objective function. The imperialist states together with their colonies form some empires. To form the initial empires, the normalized cost of an imperialist is defined as:

$$C_n = f_{\cos t}^{(imp,n)} - \max_i \left(f_{\cos t}^{(imp,i)} \right) \quad (2)$$

where $f_{\cos t}^{(imp,n)}$ is the cost of the nth imperialist and C_n is its normalized cost. The initial colonies are divided among empires based on normalized cost, and for the nth empire it is as follows:

$$NC_n = Round\left(\left|\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}\right| \cdot N_{col}\right) \tag{3}$$

where NC_n is the initial number of colonies associated to the nth empire. In order to select the colonies of nth empire, the order of colonies is changed using a random permutation and then NC_n ones are chosen. These colonies along with the nth imperialist form the nth empire.

After forming the initial empires, the colonies in each empire start moving toward their relevant imperialist country. This movement is shown in Fig. 1 in which a colony moves toward the imperialist by using a random generator that is uniformly distributed between 0 and $\beta \times d$:

$$\{x\}_{new} = \{x\}_{old} + U(0, \beta \times d) \times \{V_1\} \tag{4}$$

where β is a control parameter and d is the distance between colony and imperialist. $\{V_1\}$ is a vector which its start point is the previous location of the colony and its direction is toward the imperialist locations. The length of this vector is set to unity.

In order to increase the searching around the imperialist, a random amount of deviation is added to the direction of movement in the original ICA. Figure 1 shows the new direction which is obtained by deviating the previous location of the country by the amount θ which is a random number generator with uniform distribution.

At the end of each iteration, if the new position of a colony is better than that of the corresponding imperialist (considering the cost function), the imperialist and the colony change their positions and the new location with lower cost becomes the imperialist.

The colonies are loyal to their empires; however all empires try to take the possession of the colonies of other empires and control them according to the imperialistic competition strategy. The imperialistic competition gradually reduces the power of the weaker empires and increases the power of more powerful ones by picking some (usually one) of the weakest colonies of the weakest empires and making a competition among all empires to possess these (this) colonies. Based on their total power, in this competition, each of empires will have a likelihood of taking possession of the above mentioned colonies.

Total power of an empire is affected by the power of imperialist country and the colonies of an empire as

$$TC_n = f_{\cos t}^{(imp,n)} + \xi \cdot \frac{\sum_{i=1}^{NC_n} f_{\cos t}^{(col,i)}}{NC_n} \tag{5}$$

where TC_n is the total cost of the nth empire and ξ is a positive number. Similar to (2), the normalized total cost is defined as

$$NTC_n = TC_n - \max_i (TC_i) \tag{6}$$

where NTC_n is the normalized total cost of the nth empire. Having the normalized total cost, the possession probability of each empire is evaluated by

$$P_n = \left|\frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i}\right| \tag{7}$$

When an empire loses all its colonies, it is assumed to be collapsed. In this model when the powerless empires collapse in the imperialistic competition, the corresponding imperialist will be added to an empire as a colony. This step is known as "implementation".

Moving colonies toward imperialists are continued and imperialistic competition and implementations are performed during the search process. When the number of iterations reaches a pre-defined value, the search process is stopped (Kaveh and Talatahari 2010b, c).

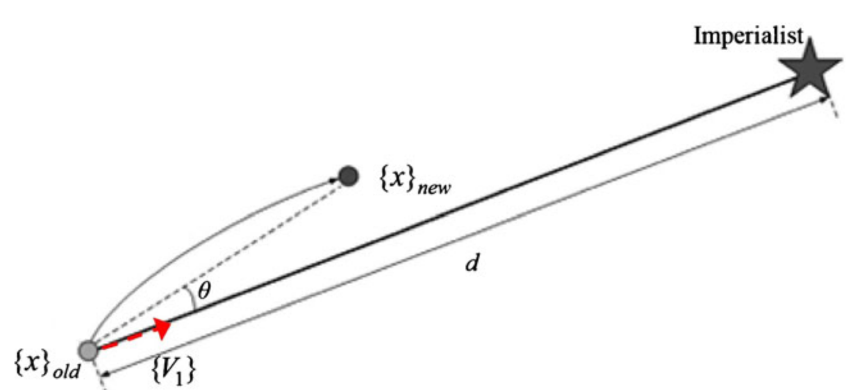

Fig. 1 Movement of colonies to its new location in the original ICA

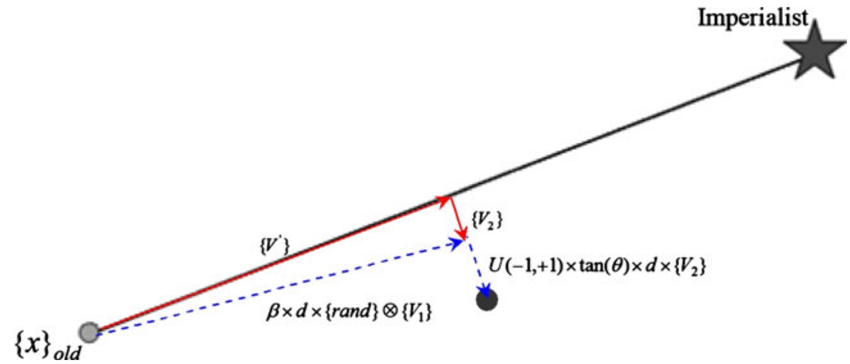

Fig. 2 Movement of colonies to the new location in the improved ICA

Table 1 The chaotic maps

Chaotic map name	Description
Sinusoidal map (May 1976)	$x_{k+1} = \sin(\pi x_k)$
Logistic map (May 1976)	$x_{k+1} = 4x_k\,(1 - x_k)$
Zaslavskii map (Zaslavskii 1978)	$x_{k+1} = (x_k + 400 + 12y_{k+1}) - \left[x_k + 400 + 12y_{k+1}\right]^*$ $y_{k+1} = \cos(2\pi x_k) + e^{-3}y_k$
Tent map (Peitgen et al. 1992)	$x_{k+1} = \begin{cases} x_k/0.7 & x_k < 0.7 \\ 10/3x_k(1 - x_k) & otherwise \end{cases}$

*Here, $[a]$ denotes the largest integer less than a

Fig. 3 Pseudo-code for the CICA methods

1) Initialize the algorithm parameters.

2) Select a chaotic map and generate chaotic variable according to the selected map.

3) Move the colonies toward their relevant imperialist chaotically (Chaotically assimilating).

3) If there is a colony in an empire which has lower cost than that of imperialist, exchange the positions of that colony and the imperialist.

4) Compute the total cost of all empires.

5) Use imperialistic competition and pick the weakest colony from the weakest empire.

6) Eliminate the powerless empires.

7) If there is just one empire, stop, if not go to 2.

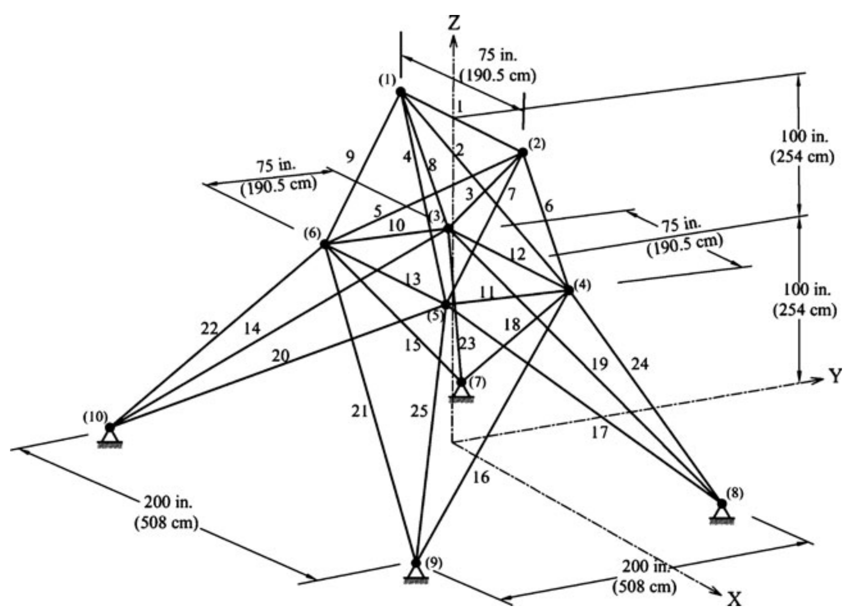

Fig. 4 A 25-bar spatial truss structure

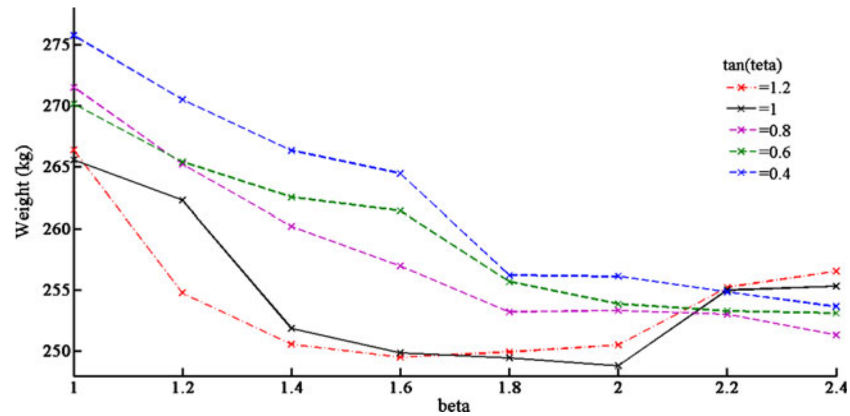

Fig. 5 Effect of the CICA-1 parameters on the average weight of the 25-bar truss

Table 2 Optimal design comparison for the 25-bar spatial truss

Element group		Optimal cross-sectional areas (cm^2)					
		ICA	OICA	CICA-1	CICA-2	CICA-3	CICA-4
1	A_1	0.0645	0.0645	0.0645	0.0645	0.0645	0.0645
2	$A_2 \sim A_5$	14.148	14.219	12.523	14.394	14.877	12.342
3	$A_6 \sim A_9$	17.903	18.768	19.580	18.935	18.245	21.277
4	$A_{10} \sim A_{11}$	0.0645	0.0645	0.0645	0.0645	0.0645	0.0645
5	$A_{12} \sim A_{13}$	0.0645	0.0645	0.0645	0.0645	0.0645	0.0645
6	$A_{14} \sim A_{17}$	4.3806	4.7871	4.2645	4.2387	4.3548	4.5677
7	$A_{18} \sim A_{21}$	10.419	9.9225	10.968	9.7483	9.7226	10.355
8	$A_{22} \sim A_{25}$	17.652	17.039	17.264	17.684	17.684	16.613
Best weight (kg)		247.68	247.63	247.38	247.55	247.54	247.75
Average Weight (kg)		257.22	249.44	248.81	253.58	249.29	249.33
SD (kg)		9.378	1.699	1.225	5.603	2.361	2.506

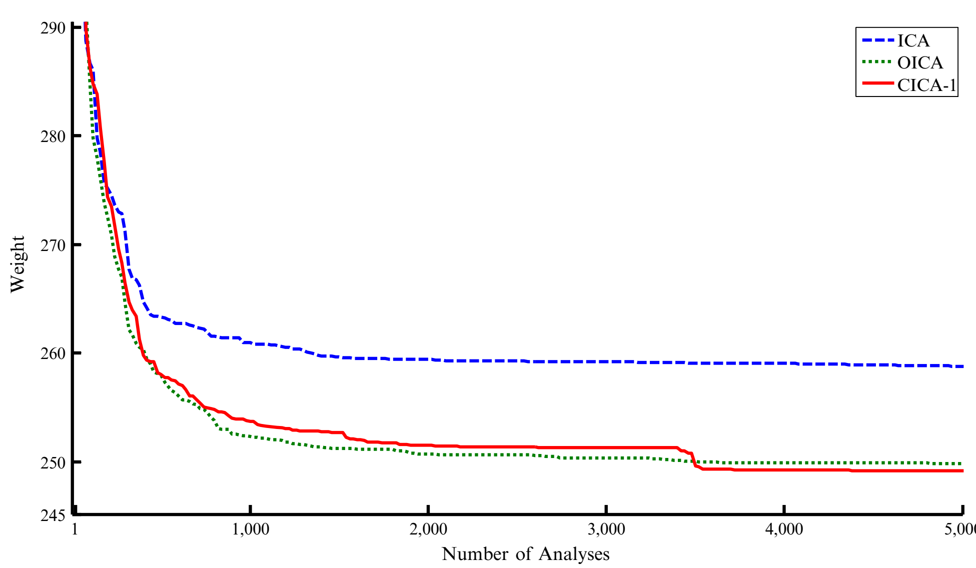

Fig. 6 Convergence rate comparison between the three algorithms for the 25-bar spatial truss structure (average of 30 different runs)

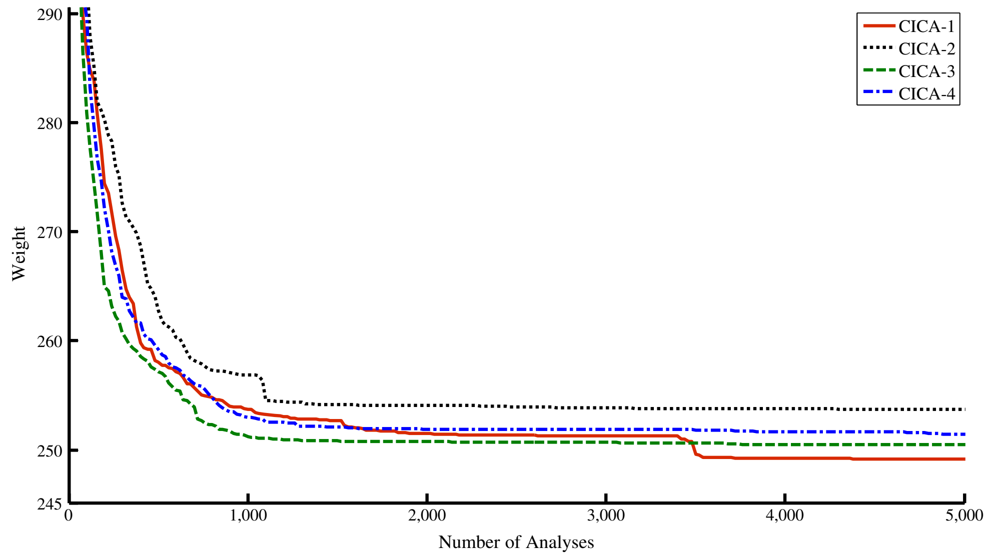

Fig. 7 Convergence rate comparison between the different CICA algorithms for the 25-bar spatial truss structure (average of 30 different runs)

2.2 Improved imperialist competitive algorithm

Recently, Kaveh and Talatahari (2010b) presented an improved ICA. This algorithm is obtained by modifying the movement stage of the original algorithm. Considering the movement process of the ICA, a point out of the colony-imperialistic contacting line can be obtained as indicated in Fig. 2. In this algorithm, not only different random values are used, but also the orthogonal colony-imperialistic contacting line is utilized for deviating the colony as follows:

$$\{x\}_{new} = \{x\}_{old} + \beta \times d \times \{rand\} \otimes \{V_1\} + U(-1, +1) \times \tan(\theta) \times d \times \{V_2\}, \{V_1\} \cdot \{V_2\} = 0, \|\{V_2\}\| = 1 \quad (8)$$

where $\{V_2\}$ is perpendicular to $\{V_1\}$, and therefore from now on this algorithm will be called orthogonal imperialist competitive algorithm (OICA). To implement this equation, we must determine the vector $\{V_2\}$. After obtaining $\{V_1\}$ (with start point of the location of the colony and the direction toward the imperialist location), similarly it is possible to find the image of the vector $\beta \times d \times \{rand\} \otimes \{V_1\}$ on $\{V_1\}$. This is shown by $\{V'\}$ in the figure. Now $\beta \times d \times \{rand\} \otimes \{V_1\} - \{V'\}$ results in the direction of $\{V_2\}$. Since vector must be crossed the point obtained from the two first terms, we use a random generator shown by $U(-1, +1)$ for the third term of the (8) which changes its value in addition to its direction by using the negative values.

3 Chaotic imperialist competitive algorithm

Chaos theory as a new emerging theory has been considered in various scientific in recent decades. A chaotic map is a deterministic pseudo-randomness. Hence, chaos could be considered as a serious alternative of randomness for those systems which their behaviors appear strange (Yousefpoor et al. 2008). Chaos theory is based on two principles. The first principle is that simple systems will exhibit complex behavior which cannot be explained using conventional theories. The second principle is that complex systems will exhibit behavior which will seem random and unstructured, but it has an underlying order. In other word, chaos is a bounded unstable dynamic behavior that includes infinite unstable periodic motions in nonlinear systems. Therefore, currently chaos as a kind of dynamic behavior of nonlinear systems has raised enormous interest in optimization theory (He et al. 2009). In random-based optimization

Table 3 Performance comparison for the 25-bar spatial truss

	Rajeev & Krishnamoorthy GA (1992)	Schutte & Groenwold PSO (2003)	Kaveh & Talatahari PSACO (2009a)	Kaveh & Talatahari HPSACO (2009a)	Kaveh & Talatahari HBB–BC (2009b)	Present study CICA-1
Best Weight (kg)	247.66	247.30	247.23	247.20	247.28	247.38
Average Weight (kg)	N/A	248.04	N/A	247.44	247.50	248.81
No. of analyses	N/A	9,596	28,850	9,875	12,500	5,000

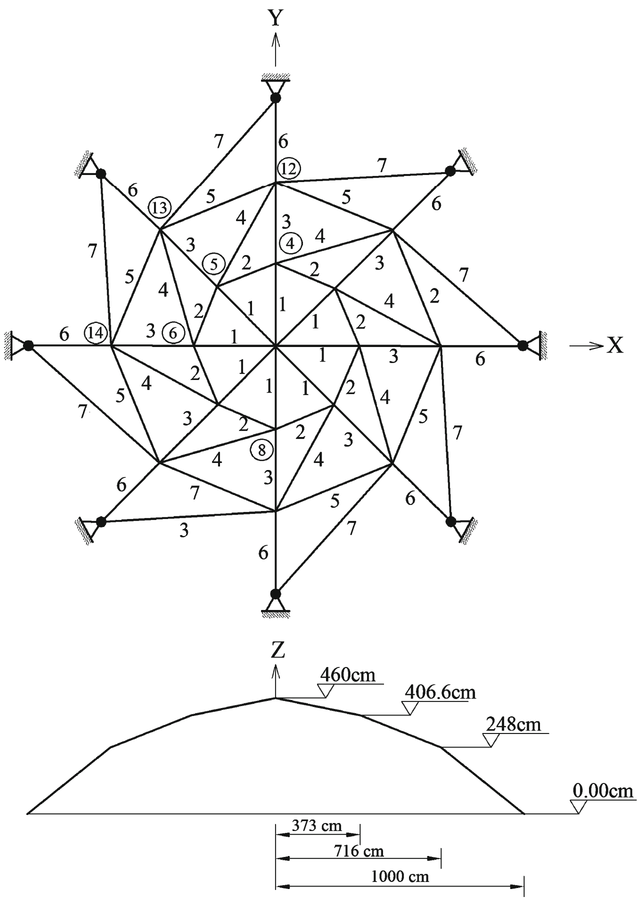

Fig. 8 A 56-bar dome spatial truss structure

algorithms, the methods using chaotic variables instead of random variables are called chaotic optimization algorithm (COA). Optimization algorithms based on the chaos theory are stochastic search methodologies that differ from any of the existing evolutionary computation and swarm intelligence methods. Due to the non-repetition of chaos, it can carry out overall searches at higher speeds than stochastic searches that depend on probabilities (Coelho and Mariani 2008).

Here we present a chaotic imperialist competitive algorithm. When a random number is needed by the CICA algorithm, it can be generated by iterating one step of the chosen chaotic map (cm) being started from a random initial condition at the first iteration of the CICA. One-dimensional noninvertible maps are the simplest systems with capability of generating chaotic motion. The choice of chaotic sequences can justified theoretically by their unpredictability, corresponding to their spread-spectrum characteristic and ergodic properties. The chaotic maps that generate chaotic sequences in CICA steps used in the experiments are listed in the Table 1. New chaotic ICA (CICA) algorithm may simply be described as follows:

Parameter $\{rand\}$ and $U(-1, +1)$ of (8) are modified by the selected chaotic maps and the assimilation (moving the colonies of an empire toward the imperialist) equation is modified by:

$$\{x\}_{new} = \{x\}_{old} + \beta \times d \times \{cm\} \otimes \{V_1\} + cm \times \tan(\theta) \times d \times \{V_2\}, \{V_1\} \cdot \{V_2\} = 0, \|\{V_2\}\| = 1 \quad (9)$$

where *cm* is a chaotic variable based on the sinusoidal map for CICA-1, logistic map for CICA-2, zaslavskii map for CICA-3 and tent map for CICA-4. Figure 3 presents a pseudo-code for the CICA methods.

In designing a meta-heuristic, two contradictory criteria must be taken into account: exploration of the search space (diversification) and exploitation of the best solutions found (intensification). Promising regions are determined by the obtained "good" solutions. In intensification, the promising regions are explored more thoroughly in the hope to find better solutions. In diversification, non explored regions must be visited to be sure that all regions of the search space

Table 4 Optimal design comparison for the 56-bar dome truss

Element group	Optimal cross-sectional areas (mm^2)					
	ICA	OICA	CICA-1	CICA-2	CICA-3	CICA-4
1	200.00	200.00	200.00	200.00	200.00	200.00
2	799.78	805.03	798.54	789.41	807.65	790.76
3	1425.84	1423.39	1424.38	1403.87	1419.77	1405.94
4	588.11	588.35	588.91	591.10	587.07	591.24
5	1081.78	1075.07	1082.49	1088.69	1075.56	1084.50
6	892.09	900.44	894.64	913.02	899.34	916.42
7	502.14	499.41	501.43	508.13	501.40	506.56
Best weight (kg)	546.14	546.15	546.13	546.16	546.15	546.15
Average Weight (kg)	547.91	546.24	546.21	546.31	546.24	546.34
SD (kg)	5.791	0.85	0.49	0.62	0.56	0.59

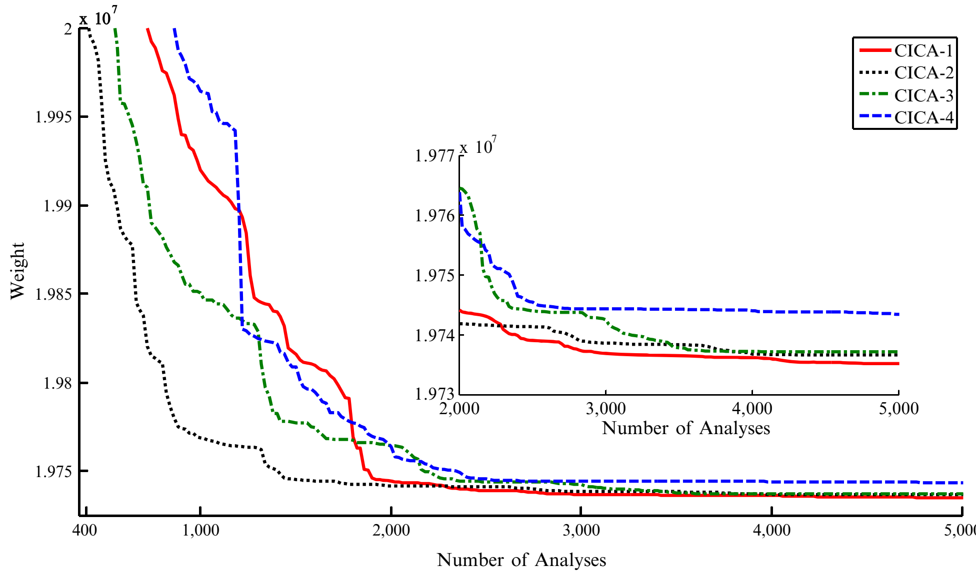

Fig. 9 Convergence rate comparison between the different CICA algorithms for the 56-bar spatial truss structure (average of 30 different runs)

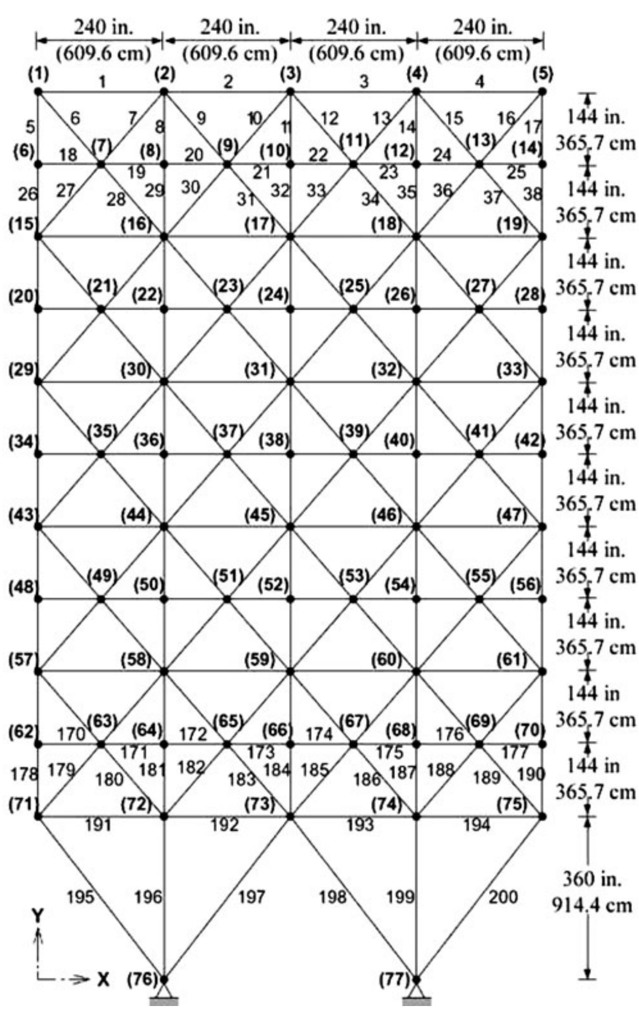

Fig. 10 A 200-bar spatial truss structure

are evenly explored and that the search is not confined to only a reduced number of regions (Talbi 2009).

Imperialist countries started to improve their colonies by moving all the colonies toward the imperialist using (8). In order to improve the searching abilities of the algorithm around the imperialist, we use chaotic variables instead of random variables in (9). In fact, however, random parameters of the ICA may affect the algorithm performance and cannot ensure the optimization's ergodicity entirely in phase space, because they are random in original ICA. That is why; these parameters may be selected chaotically by using chaotic maps because of the ergodic property of chaotic variables.

4 The CICA method for optimal design of truss structures

Unlike exact methods, meta-heuristics allow to tackle large-size problem instances by delivering satisfactory solutions in a reasonable time. There is no guarantee to find global optimal solutions or even bounded solutions. Meta-heuristics have received more and more popularity in the past 20 years. Application of meta-heuristics falls into a large number of areas; one of them is size optimization of truss

Table 5 Performance comparison for the 200-bar planer truss problem

	ICA	OICA	CICA-1
Best weight (kg)	12,082.5	11,802.2	11,486.3
Average Weight (kg)	12,553.6	12,203.8	11,828.8
SD (kg)	1,870.5	1,025.6	544.7

structures. Size optimization of truss structures involves determining optimum values for member cross-sectional areas, A_i, that minimizes the structural weight W. This minimum design should also satisfy the inequality constraints that limit design variable sizes and structural responses. The optimal design of a truss can be formulated as:

$$\text{minimize} \quad W(\{x\}) = \sum_{i=1}^{n} \gamma_i . A_i . L_i \tag{10}$$

$$\text{subject to} \quad \begin{array}{ll} \delta_{\min} \le \delta_i \le \delta_{\max} & i = 1, 2, \ldots, m \\ \sigma_{\min} \le \sigma_i \le \sigma_{\max} & i = 1, 2, \ldots, n \\ A_{\min} \le A_i \le A_{\max} & i = 1, 2, \ldots, ng \end{array} \tag{11}$$

where $W(\{x\})$ = weight of the structure; n = number of members making up the structure; m = number of nodes; ng = number of groups (number of design variables); γ_i = material density of member i; L_i = length of member i; A_i = cross-sectional area of member i chosen between $A_{\min}$ and $A_{\max}$; min = lower bound and max = upper bound; σ_i and δ_i = the stress and nodal deflection.

Here, an appropriate penalty function is utilized to handle the constraints. In utilizing penalty functions, if the constraints are between the allowable limits, the penalty is zero; otherwise the amount of penalty is obtained by dividing the violation of allowable limit to the limit itself. After analyzing a structure, the deflection of each node and the stress in each member are obtained. These values are compared

Table 6 Optimal design comparison for the 200-bar planar truss

Element group	HS	SA	AL	GA	Present study		
					ICA	OICA	CICA
$A_{1\sim4}$	0.806	0.941	0.954	2.238	0.665	0.645	0.677
$A_{5, 8, 11, 14, 17}$	6.548	6.064	6.096	6.974	6.107	6.102	6.956
$A_{19\sim24}$	0.684	0.645	0.645	0.645	0.664	0.645	1.126
$A_{18, 25, 56, 63, 94, 101, 132, 139, 170, 177}$	0.703	0.645	0.645	0.645	0.665	0.645	0.680
$A_{26, 29, 32, 35, 38}$	12.49	12.52	12.55	13.82	19.54	12.21	12.64
$A_{6, 7, 9, 10, 12, 13, 15, 16,27,28, 30, 31, 33, 34, 36, 37}$	1.729	1.910	1.923	2.239	1.461	1.740	2.171
$A_{39\sim42}$	0.671	0.645	0.645	0.645	0.645	0.645	0.677
$A_{43, 46, 49, 52, 55}$	19.18	20.03	20.15	23.00	27.27	31.19	19.08
$A_{57\sim62}$	0.839	0.645	0.645	2.239	0.664	3.281	0.677
$A_{64, 67, 70, 73, 76}$	26.99	26.48	26.60	31.00	33.92	25.69	29.68
$A_{44, 45, 47, 48, 50, 51, 53, 54,65,66, 68, 69, 71, 72, 74, 75}$	2.555	2.600	2.574	2.839	2.073	2.984	2.582
$A_{77\sim80}$	2.845	1.232	0.645	2.839	2.280	0.646	1.298
$A_{81, 84, 8790, 93}$	33.46	35.02	34.79	38.40	32.21	34.38	33.82
$A_{95\sim100}$	1.232	0.645	0.645	2.239	3.479	4.674	0.677
$A_{102, 105, 108, 111, 114}$	40.26	41.47	41.25	42.40	38.59	39.88	49.23
$A_{82, 83, 85, 86, 88, 89,91, 92,103,104, 106, 107,109, 110, 112, 113}$	4.510	3.697	3.394	6.155	4.560	4.986	2.946
$A_{115\sim118}$	0.742	0.852	2.806	2.239	0.665	0.645	0.677
$A_{119, 122, 125, 128, 131}$	50.09	51.43	51.29	55.00	48.42	64.16	48.69
$A_{133\sim138}$	0.645	0.645	0.645	0.645	0.645	0.645	3.758
$A_{140, 143, 146, 149, 152}$	56.94	57.87	57.74	60.00	54.80	58.48	54.61
$A_{120, 121, 123, 124, 126, 127,129,130, 141, 142, 144, 145, 147, 148,150, 151}$	4.503	4.542	5.541	6.155	3.676	5.356	5.426
$A_{135\sim156}$	10.04	2.703	0.968	11.38	9.433	1.635	0.677
$A_{157, 160, 163, 166, 169}$	70.84	70.06	70.90	85.81	65.54	71.81	67.47
$A_{171\sim176}$	0.845	0.645	0.645	2.239	0.665	1.497	3.360
$A_{178, 181, 184, 187, 190}$	78.32	76.52	77.35	85.81	72.17	78.75	73.57
$A_{158, 159, 161, 162, 164, 165, 167, 168, 179, 180, 182, 183, 185, 186,188, 189}$	10.56	6.671	5.890	13.82	8.183	7.216	7.067
$A_{191\sim194}$	32.28	43.10	42.98	31.00	46.34	35.62	44.34
$A_{195, 197, 198, 200}$	60.35	69.74	69.68	60.00	78.49	63.08	66.46
$A_{196, 199}$	97.35	89.29	89.16	110.8	85.75	91.83	86.98
Best weight (kg)	11531.3	11531.7	11542.4	12947.3	12082.5	11802.2	11486.3
Number of analyses	48,000	9,650	N/A	51,360		15,000	

to the allowable limits to calculate the penalty functions as (Kaveh et al. 2008)

$$\begin{cases} \delta_i^{\min} < \delta_i < \delta_i^{\max} & \Rightarrow \Phi_\delta^{(i)} = 0 \\ \delta_i^{\min} > \delta_i \ \text{or} \ \delta_i^{\max} < \delta_i & \Rightarrow \Phi_\delta^{(i)} = \frac{\delta_i - \delta_i^{\min/\max}}{\delta_i^{\min/\max}} \\ & \quad i = 1, 2, \ldots, m \end{cases} \quad (12)$$

$$\begin{cases} \sigma_i^{\min} < \sigma_i < \sigma_i^{\max} & \Rightarrow \Phi_\sigma^{(i)} = 0 \\ \sigma_i^{\min} > \sigma_i \ \text{or} \ \sigma_i^{\max} < \sigma_i & \Rightarrow \Phi_\sigma^{(i)} = \frac{\sigma_i - \sigma_i^{\min/\max}}{\sigma_i^{\min/\max}} \\ & \quad i = 1, 2, \ldots, n \end{cases} \quad (13)$$

In this method, the aim of the optimization is redefined by introducing the cost function as

$$f_{cost}(\{x\}) = \left(1 + \varepsilon_1 \cdot \sum \Phi\right)^{\varepsilon_2} \times W(\{x\}) \quad (14)$$

where Φ_σ and Φ_δ = the value of stress penalty and the nodal deflection penalty, respectively. The constant ε_1 and ε_2 are selected considering the exploration and the exploitation rate of the search space. Here, ε_1 is set to unity, ε_2 is selected in the way that it decreases the penalties and reduces the cross-sectional areas. Thus, in the first steps of the search process ε_2 is set to 1.5, and ultimately increased to 3 (Kaveh et al. 2008).

From the structural design point of view, the CICA determines the appropriate sections for each group of elements so that with these set of sections the response of the truss is within the limitations imposed by the design condition when it has the minimum weight. The chaotically movement of colonies towards their relevant imperialist states along with competition among empires and also the collapse mechanism will hopefully cause all the countries to converge to a state in which there exist just one empire in the world and all the other countries are colonies of that empire. In this ideal new world, colonies will have the same position and power as the imperialist.

5 Design examples

In this section, some truss structures are optimized utilizing the present method. The optimization examples include:

- A 25-bar spatial truss;
- A 56-bar dome truss;
- A 200-bar planar truss;
- A 244-bar transformation tower.

The examples are solved by the standard ICA, improved ICA (OICA) and the present ICA (CICA) and the results are compared. For two first examples all variants of the CICA are utilized while for the other ones only the best one is selected to evaluate. $N_{country}$ is set to 20 and 30 for the first two examples and for the later examples, respectively. The algorithms are coded in Matlab and a direct stiffness method is utilized to analyze the structures.

5.1 25-bar spatial truss

The topology and nodal numbers of a 25-bar spatial truss structure are shown in Fig. 4. The material density is considered as 2767.990 kg/m^3 and the modulus of elasticity is taken as 68,950 MPa. Twenty five members are categorized into eight groups, as: (1) A_1, (2) $A_{2\sim5}$, (3) $A_{6\sim9}$, (4) $A_{10\sim11}$,

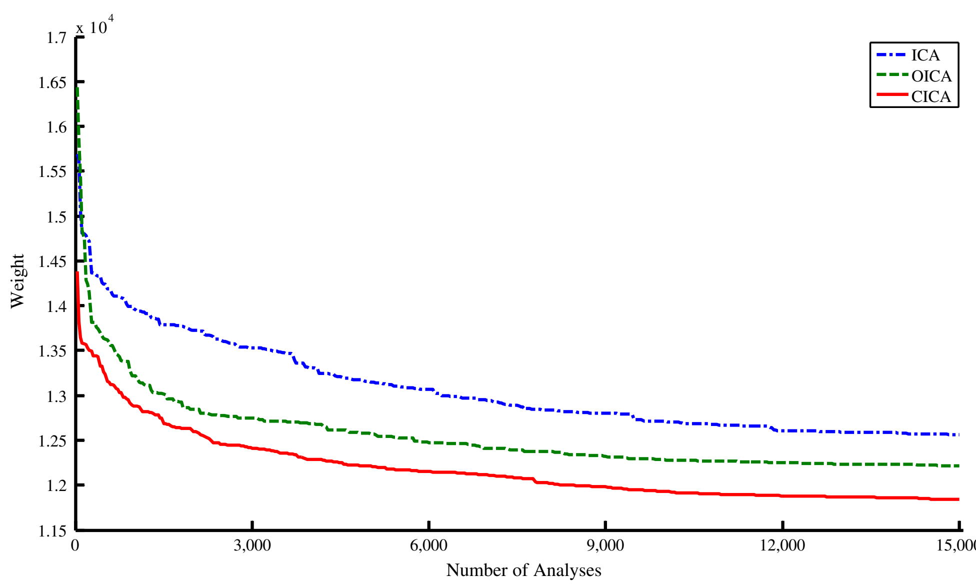

Fig. 11 Convergence rate comparison between the three algorithms for the 200-bar spatial truss structure (average of 30 different runs)

(5) $A_{12\sim13}$, (6) $A_{14\sim17}$, (7) $A_{18\sim21}$, and (8) $A_{22\sim25}$. The detailed information related to the loading condition and constraints can be found in Kaveh and Talatahari (2010a).

Tuning the utilized parameters for a meta-heuristic algorithm is a very important issue. In order to fulfill this, herein a sensitive study on two parameters of the algorithm is performed utilizing the 25-bar spatial truss. For various values of β and $tan(\theta)$, this example is solved several times (20 times for each value of β and $tan(\theta)$) and the average weight of designs is shown in Fig. 5. This figure shows that $\beta > 1$ make the colonies to move closer to the imperialist state from both sides while a very close value to 1 for β reduces the search ability of the algorithm. As shown in the figure, $\beta = 2$ and $tan(\theta) = 1$ are suitable values for the CICA-1 algorithm. These parameter values are used for all other presented examples.

In Table 2, the optimum cross sectional areas and statistical information of the solutions obtained by the ICA, OICA and the different variants of the CICA are presented. Figures 6 and 7 provide a comparison of the convergence rates of the ICA, OICA, CICA-1, CICA-2, CICA-3 and CICA-4 algorithms. Comparing the results of different chaotic maps, it can be concluded that CICA have somewhat shown better performance when Sinusoidal and Zaslavskii maps have been used for generating chaotic signals. The best weight obtained by the CICA-1 is 247.38 kg, and this shows that the CICA-1 has the best performance than other algorithms. As another investigation and for testing the degree of consistency from the Table 2, it can be seen that the standard deviation of the results by CICA-1 in 30 independent runs is the smallest one. Table 3 compares the optimum results obtained by the CICA-1 and the GA (Rajeev and Krishnamoorthy 1992), the PSO (Schutte and Groenwold 2003), the PSACO and HPSACO (Kaveh and Talatahari 2009a) and HBB–BC (Kaveh and Talatahari 2009b). The differences between the results are very small, however, the present method needs small number of analyses to find the optimum result.

5.2 56-bar dome truss

A 56-bar dome truss structure is shown in Fig. 8. Members of the dome are initially collected into 3 groups as given by Kelesoglu (2007), but in this study all members are re-grouped into 7 groups (see Fig. 8). The value of the modulus of elasticity is taken as 210 kN/mm^2 and the material density is 2767.990 kg/m^3. The displacement limits as well as the considered loading cases are taken from Kelesoglu (2007). The minimum cross-sectional area of all members is 200mm^2 and the maximum cross sectional area is 2000 mm^2.

Table 4 shows the statistical results and the optimum cross sectional areas for the 56-bar dome truss using the proposed method. The CICA-1 has achieved the best solution after 5,000 analyses and found an optimum weight of 546.13 kg. Although the differences between the best results of these ICA-based algorithms are small, however the new CICA methods can reduce the value of standard deviation and in this way the reliability of the algorithm is increased considerably. As it can be seen from Table 4, the CICA methods reduce the standard deviation almost ten and two times in comparison with the standard and improved

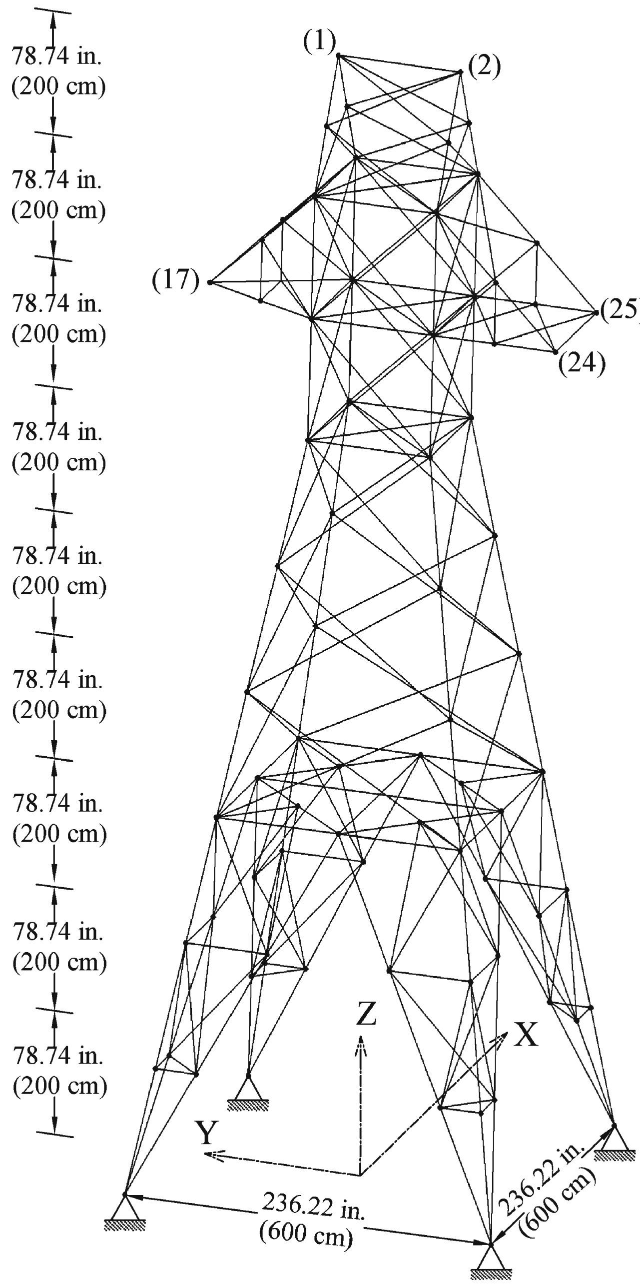

Fig. 12 A 244 -bar transformation tower truss

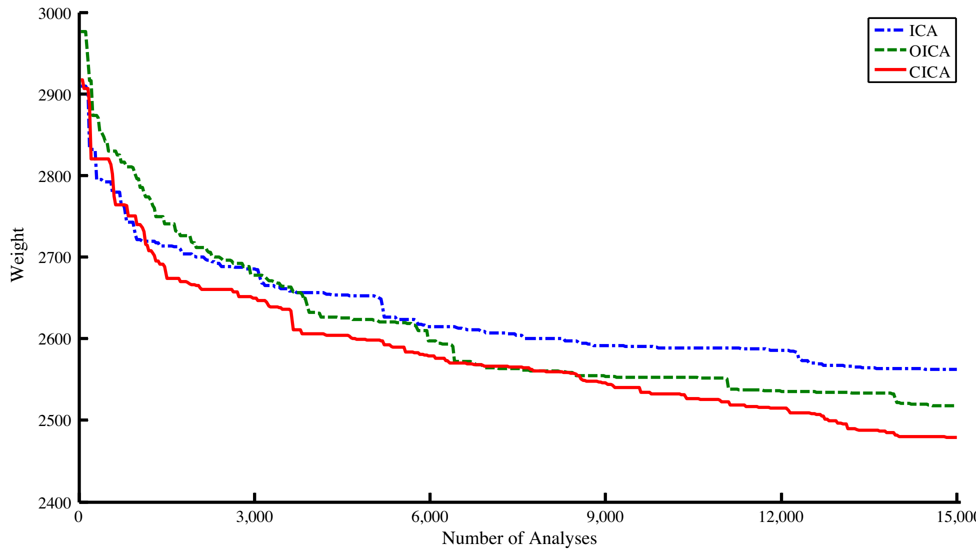

Fig. 13 Convergence rate comparison between the best results of three algorithms for the 244-bar truss structure

ICA algorithms, respectively. Figure 9 compares the convergence characteristic curve of four different CICA methods.

5.3 200-bar planar truss

The 200-bar planar truss structure is shown in Fig. 10. The 200 structural members of this planar truss are categorized as 29 groups (design variable) using symmetry. The paper of Lee and Geem (2004) presents the constraints and loading conditions.

The minimum weight and the statistical values of the best solution obtained by CICA methods are reported in Table 5 in which the standard deviation of the CICA-1 is 1.9 and 3.4 times less than the ones obtained by the standard ICA and OICA. The minimum weight and the values of the cross sectional area obtained by the standard ICA, OICA, CICA-1 as well as some other previous studies reported in the literature such as harmony search (HS) (Lee and Geem 2004), a modified simulated annealing algorithm (Lamberti 2008) an augmented Lagrangian method (Coster and Stander 1996) and an improved genetic algorithm (Togan and Daloglu 2008) are presented in Table 6. As shown in the table, the proposed CICA method can find the best design among the other existing studies and the best weight of the CICA is 11,486.3 kg. It is worth pointing out that the CICA-1 method requires 15,000 searches to reach the optimum design. The convergence characteristic curve for this case using the ICA, OICA and CICA is shown in Fig. 11.

5.4 A 244-bar transformation tower

The final example is a 244-bar transmission tower shown in Fig. 12, (Kaveh and Talatahari 2009a). Members of the transmission tower are linked into 32 groups (design variables). Other information related to this example is presented in Kaveh and Talatahari (2009a).

The maximum number of analyses is 15,000 for the ICA, OICA and CICA-1. The CICA achieves the best solution 2,478.95 kg while the OICA and ICA algorithm achieves 2,517.29 kg and 2,562.09 kg respectively. The HPSACO and PSOPC algorithms achieved 2,415.02 kg and 2,652.56kg, respectively (Kaveh and Talatahari 2009a). Although, the HPSACO finds a 2.5% lighter design, however, it is worth to note that the HPSACO utilizes the PSO with two auxiliary tools (ACO and HS) and if one add these tools to CICA, obviously the resultant method will be improved. Figure 13 compares the convergence history for the minimum weight of 244-bar transformation tower solved by different ICA-based methods.

6 Conclusion

Recently, chaos is found to have a great potential in the theory of optimization. An irregular motion and unpredictable random behavior exhibited by a deterministic nonlinear system are the major positive properties of chaotic systems, and therefore they can be utilized instead of different random number generators available in of the stochastic optimization algorithms.

This paper combines the benefits of chaotic and the imperialist competitive algorithm to determine optimum design of truss structures. Imperialist competitive algorithm, a socio-politically motivated algorithm, contains some agents or countries and movement of the colonies and imperialistic competition are the two main steps of this algorithm. Here we modified the movement step by using chaotic maps. To fulfill this aim, the orthogonal imperialist competitive algorithm, as an improved ICA with two movement steps, and four different chaotic maps containing Sinusoidal, Logistic, Zaslavskii and Tent maps are utilized.

These different chaotic maps are investigated by solving two benchmark truss examples involving 25- and 56-bar trusses to recognize the most suitable one for the present algorithm. The results show that the use of Sinusoidal map results in a better performance for the CICA than others. Two other larger examples are also considered to obtain more clear details about the performance of the new algorithm. These are 200- and 244-bar trusses with 29 and 32 groups (design variables), respectively. Almost for all examples, the performance of the new algorithm is far better than the original ICA and OICA; especially when the standard deviations of the results are compared. The standard deviation of the new algorithm is much better than other ICA-based algorithms and this illustrates the high ability of the new algorithm. Due to the simplicity and potency of the present method, it seems that it can easily be utilized for many engineering problems to find the optimum designs.

Acknowledgments The second author is grateful to the Iran National Science Foundation for the support.

References

Alatas B (2010) Chaotic harmony search algorithms. Appl Math Comput 216:2687–2699

Antoniou I, Akishina EP, Ivanov VV, Kostenko BF, Stalios AD (2003) Cellular automata study of high burn-up structures. Chaos, Solitons Fractals 28:1111–1128

Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: Proceedings of IEE congress on evolutionary computation, Singapore, pp 4661–4667

Chen G, Dong X (1998) From chaos to order: methodologies, perspectives and applications. World Scientific, Singapore

Coelho L, Mariani V (2008) Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. Expert Syst Appl 34:1905–1913

Coster IE, Stander N (1996) Structural optimization using augmented Lagrangian methods with secant Hessian updating. Struct Optim 12:113–119

Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B 26(1):29–41

Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan, pp 1942–1948

Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm; harmony search. Simulation 76:60–68

Harb AM, Abdel-Jabbar N (2003) Controlling hopf bifurcation and chaos in a small power system. Chaos, Solitons Fractals 18:1055–1063

He Y, Zhou J, Xiang X, Chen H, Qin H (2009) Comparison of different chaotic maps in particle swarm optimization algorithm for long-term cascaded hydroelectric system scheduling. Chaos, Solitons Fractals 42:3169–3176

Kaveh A, Talatahari S (2009a) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 87(5–6):267–283

Kaveh A, Talatahari S (2009b) Size optimization of space trusses using Big Bang–Big Crunch algorithm. Comput Struct 87:1129–1140

Kaveh A, Talatahari S (2010a) Optimal design of skeletal structures via the charged system search algorithm. Struct Multidiscipl Optim 41(6):893–911

Kaveh A, Talatahari S (2010b) Optimum design of skeletal structures using imperialist competitive algorithm. Comput Struct 88:1220–1229

Kaveh A, Talatahari S (2010c) Imperialist competitive algorithm for engineering design problems. Asian J Civil Eng 11(6):675–697

Kaveh A, Farahmand Azar B, Talatahari S (2008) Ant colony optimization for design of space trusses. Int J Space Struct 23(3):167–181

Kelesoglu O (2007) Fuzzy multiobjective optimization of truss-structures using genetic algorithm. Adv Eng Softw 38:717–721

Lamberti L (2008) An efficient simulated annealing algorithm for design optimization of truss structures. Comput Struct 86:1936–1953

Lee, KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82:781–798

May R (1976) Mathematical models with very complicated dynamics. Nature 261:459

Ólafsson S (2006) Metaheuristics. Handbook on Simulation, Handbooks in Operations Research and Management Science VII, Elsevier, pp 633–654

Peitgen H, Jurgens H, Saupe D (1992) Chaos and fractals. Springer-Verlag, Berlin, Germany

Rajeev S, Krishnamoorthy CS (1992) Discrete optimization of structures using genetic algorithms. J Struct Eng ASCE 118(5):1233–1250

Schutte JJ, Groenwold AA (2003) Sizing design of truss structures using particle swarms. Struct Multidiscipl Optim 25:261–269

Talbi EG (2009) Metaheuristics: from design to implementation. John Wiley & Sons, New Jersey

Tavazoei M, Haeri M (2007) Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. J Appl Math Comput 187:1076–1085

Togan V, Daloglu AT (2008) An improved genetic algorithm with initial population strategy and self-adaptive member grouping. Comput Struct 86:1204–1218

Yousefpoor P, Esfahani MS, Nojumi H (2008) Looking for systematic approach to select chaos tests. Appl Math Comput 198:73–91

Zaslavskii GM (1978) The simplest case of a strange attractor. Phys Lett A 69:145–147