INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2013; 3(1): 23-36.

CHAOS EMBEDDED CHARGED SYSTEM SEARCH FOR PRACTICAL OPTIMIZATION PROBLEMS

B. Nouhi[1], S. Talatahari[2,*,†] and H. Kheiri[1]
[1]Department of Mathematical Sciences, University of Tabriz, Tabriz, Iran
[2]Marand Faculty of Engineering, University of Tabriz, Tabriz, Iran

ABSTRACT

Chaos is embedded to the he Charged System Search (CSS) to solve practical optimization problems. To improve the ability of global search, different chaotic maps are introduced and three chaotic-CSS methods are developed. A comparison of these variants and the standard CSS demonstrates the superiority and suitability of the selected variants for practical civil optimization problems.

Received: 12 August 2012; Accepted: 20 December 2012

KEY WORDS: Charged system search; chaos; optimization; chaos-based charged system search algorithm, practical civil optimization problems.

1. INTRODUCTION

The Charged System Search (CSS), introduced by Kaveh and Talatahari [1], is a meta-heuristic optimization technique. This algorithm utilizes governing laws of electrostatics in physics and the governing laws of motion from the Newtonian mechanics [2]. This algorithm is growing and its application is extending to various optimization problems such as discrete optimum design of truss structures [2], design of skeletal structures [3], grillage system design [4], optimization of geodesic domes [5] and configuration optimization [6] etc. The results of the CSS show a better performance of the CSS comparing to those of the other heuristics [5].

Similar to many other meta-heuristics, the CSS needs to use some random generators. Recently, the idea of using chaotic systems instead of random processes has been noticed in

* Corresponding author: S. Talatahari, Marand Faculty of Engineering, University of Tabriz, Tabriz, Iran
†E-mail address: siamak.talat@gmail.com (S. Talatahari)

optimization algorithms where the role of randomness can be played by a chaotic dynamics. Experimental studies show the benefits of using chaotic signals instead of random signals [7,8]. For examples, chaos is added to genetic algorithms [9], harmony search [10], simulated annealing [11], accelerated particle swarm optimization [12], imperialist competitive algorithm [13], firefly algorithm [14] and charged system search [8].

This paper develops Chaos embedded CSS (CCSS) methods for solving practical optimization problems. In these algorithms, we use different chaotic systems to replace the parameters of the CSS. Thus different methods that use chaotic maps as efficient alternatives to pseudorandom sequences have been proposed.

The remaining of this paper is organized as follows. Review of the standard CSS is presented in Section 2. In Sections 3 and 4 different CCSS methods as well as the utilized chaotic maps are proposed, respectively. In Section 5, the suggested methods are evaluated through practical optimization problems, and the results are compared to designate the most efficient approach. Finally, the conclusion is drawn in Section 6 based on the reported comparison analyses.

2. STANDARD CHARGED SYSTEM SEARCH ALGORITHM

The Charged System Search (CSS) algorithm is based on the Coulomb and Gauss laws from electrical physics and the governing laws of motion from the Newtonian mechanics. This algorithm can be considered as a multi-agent approach, where each agent is a Charged Particle (CP). Each CP is considered as a charged sphere with radius a, having a uniform volume charge density and is equal to

$$q_i = \frac{fit(i) - fitworst}{fitbest - fitworst} \ , \ \ i = 1,2,...,N \tag{1}$$

where *fitbest* and *fitworst* are the best and the worst fitness of all the particles; *fit(i)* represents the fitness of the agent *i*, and *N* is the total number of CPs. The initial positions of CPs are determined randomly in the search space using

$$x_{i,j}^{(o)} = x_{i,\min} + rand_{ij} \cdot (x_{i,\max} - x_{i,\min}) , \ \ i = 1,2,...,N \tag{2}$$

where $x_{i,j}^{(o)}$ determines the initial value of the *i*th variable for the *j*th CP; $x_{i,\min}$ and $x_{i,\max}$ are the minimum and the maximum allowable values for the *i*th variable; $rand_{ij}$ is a random number in the interval [0,1]. The initial velocities of charged particles are taken as:

$$v_{i,j}^{(o)} = 0 , \ i = 1,2,...,N \tag{3}$$

CPs can impose electric forces on the others, and its magnitude for the CP located inside

the sphere is proportional to the separation distance between the CPs, and for a CP located outside the sphere is inversely proportional to the square of the separation distance between the particles. The kind of the forces can be attractive or repelling determined by using a force parameter ar_{ij} defined as

$$ar_{ij} = \begin{cases} +1 & k_t < \mathrm{r}and_{ij} \\ -1 & k_t > \mathrm{r}and_{ij} \end{cases} \tag{4}$$

where ar_{ij} determines the type of the force, in which +1 represents the attractive force and −1 denotes the repelling force, and k_t is a parameter to control the effect of the kind of the force. In general the attractive force collects the agents in a part of search space and the repelling force strives to disperse the agents. The resultant force is redefined as

$$\mathbf{F}_j = q_j \sum_{i,i\neq j} \left(\frac{q_i}{a^3} r_{ij} \cdot i_1 + \frac{q_i}{r_{ij}^2} \cdot i_2 \right) ar_{ij}\, p_{ij} (\mathbf{X}_i - \mathbf{X}_j)$$

$$\begin{cases} j = 1,2,...,N \\ i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a \\ i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a \end{cases} \tag{5}$$

where F_j is the resultant force acting on the jth CP; r_{ij} is the separation distance between two charged particles defined as

$$r_{ij} = \frac{\| \mathbf{X}_i - \mathbf{X}_j \|}{\| (\mathbf{X}_i + \mathbf{X}_j)/2 - \mathbf{X}_{best} \| + \varepsilon} \tag{6}$$

Here $\mathbf{X}_i$ and $\mathbf{X}_j$ are the positions of the ith and jth CPs, respectively; $\mathbf{X}_{best}$ is the position of the best current CP, and ε is a small positive number to avoid singularity. The p_{ij} determines the probability of moving each CP toward the others as

$$p_{ij} = \begin{cases} 1 & \dfrac{fit(i) - fitbest}{fit(j) - fit(i)} > rand \vee fit(j) > fit(i) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The resultant forces and the laws of the motion determine the new location of the CPs. At this stage, each CP moves towards its new position under the action of the resultant

forces and its previous velocity as

$$\mathbf{X}_{j,new} = rand_{j1} \cdot k_a \cdot \frac{\mathbf{F}_j}{m_j} \cdot \Delta t^2 + rand_{j2} \cdot k_v \cdot \mathbf{V}_{j,old} \cdot \Delta t + \mathbf{X}_{j,old} \tag{8}$$

$$\mathbf{V}_{j,new} = \frac{\mathbf{X}_{j,new} - \mathbf{X}_{j,old}}{\Delta t} \tag{9}$$

where k_a is the acceleration coefficient; k_v is the velocity coefficient to control the influence of the previous velocity; and $rand_{j1}$ and $rand_{j2}$ are two random numbers uniformly distributed in the range (0,1). If each CP moves out of the search space, its position is corrected using the harmony search-based handling approach [1]. In addition, to save the best results, a memory, known as the Charged Memory, is utilized. The flowchart of the standard CSS is presented in Figure 1.

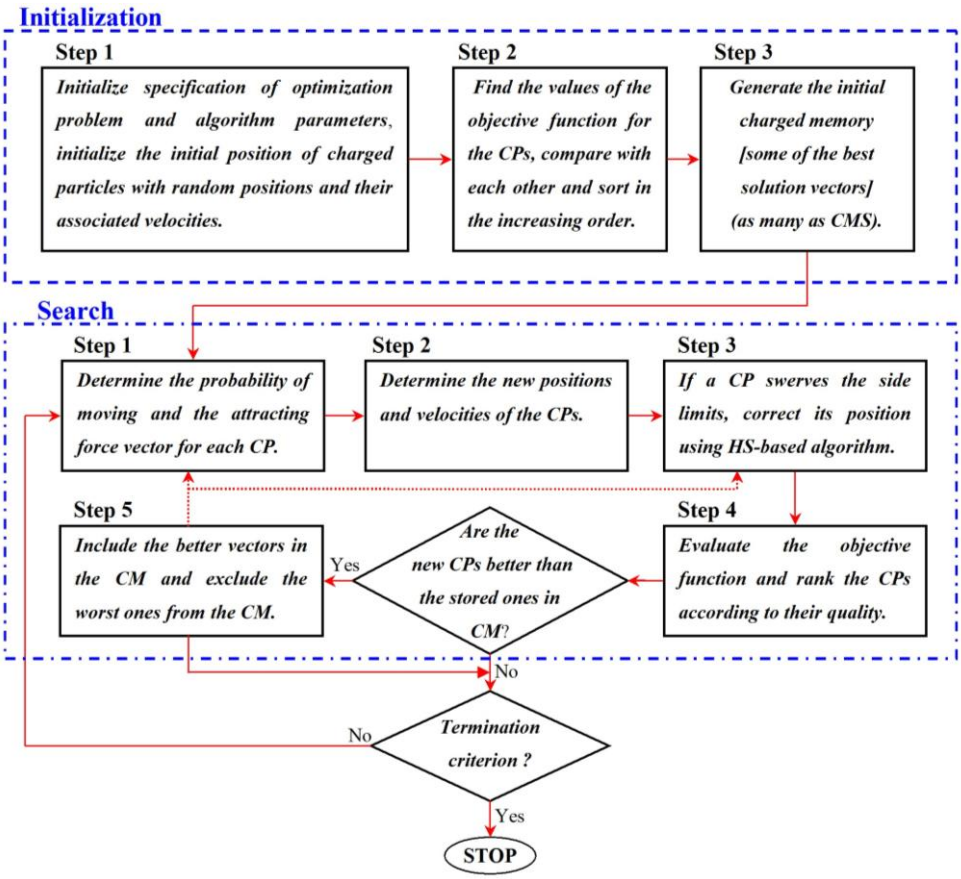

Figure 1. The flowchart of the standard CSS [2]

3. CHAOS EMBEDDED CHARGED SYSTEM SEARCH ALGORITHM

The standard CSS utilizes the fixed parameters during subsequent iterations while adjusted limit parameters may affect the performance of the algorithm and reduce/increase its convergence speed. k_t, k_a and k_v are three fixed predefined parameters for the CSS. Though these values are the key factors to control the balance of the exploration and exploitation of the algorithm, however there are two problems in using of them; first, when these are multiplied to random numbers, the resultant values will have randomized nature and therefore, their changes are limited to effects of the related random numbers during the subsequent iterations. Second, there is no deterministic approach for determining suitable values for these parameters. As a result, due to the importance of these parameters on the performance of the algorithm in one hand, and having no definite and reliable approach to determine these parameters on the other hand; their coefficients may be selected chaotically by using chaotic maps [8].

In this paper, sequences generated from chaotic systems substitute the random parameters utilized in the CSS algorithm, to improve the global convergence and to prevent being trapped in a local solution. Chaos is a deterministic, random-like process found in nonlinear, dynamical system, which is non-period, non-converging and bounded. The nature of chaos looks to be random and unpredictable, possessing an element of regularity. Mathematically, chaos is randomness of a simple deterministic dynamical system, and chaotic system may be considered as the sources of randomness [15, 16]. The new Chaos embedded CSS algorithms, denoted by CCSS, can be classified and described as follows:

3.1. CCSS-1

In this algorithm the kind of the forces (attracting or repelling) is determined by using chaotic maps, defined as

$$ar_{ij} = \begin{cases} +1 & k_t < cm_{ij} \\ -1 & k_t > cm_{ij} \end{cases} \tag{10}$$

where cm_{ij} is a chaotic variable according to the selected map. Also, the probability of moving each CP toward the others is determined as

$$p_{ij} = \begin{cases} 1 & \dfrac{fit(i) - fitbest}{fit(j) - fit(i)} > cm_{ij} \vee fit(j) > fit(i) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

3.2. CCSS-2

The coefficients of the force and velocity terms in Eq. (8) are modified by the selected

chaotic maps and position update equation is modified as

$$\mathbf{X}_{j,new} = cm_{j1} \cdot \frac{\mathbf{F}_j}{m_j} \cdot \Delta t^2 + cm_{j2} \cdot \mathbf{V}_{j,old} \cdot \Delta t + \mathbf{X}_{j,old} \tag{12}$$

3.2. CCSS-3

CCSS1 and CCSS2 are combined, that is the kind of the forces and the moving probability function are determined by using Eqs. (10) and (11) while the new position of CPs is obtained by Eq. (12).

4. UTILIZED CHAOTIC MAPS

There are developed different chaotic maps. The selected chaotic maps for the experiments of this study are listed in the following subsections.

4.1. Logistic map

This map, whose equation appears in nonlinear dynamics of biological population, highlights the chaotic behavior [17]

$$cm_{k+1} = a \cdot cm_k (1 - cm_k) \tag{13}$$

In this equation, x_k is the kth chaotic number, with k denoting the iteration number. Obviously, $cm_k \in (0,1)$ under the conditions that the initial $cm_o \in (0,1)$ and $cm_o \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$. In the experiments $a = 4$ is used.

4.2. Tent map

Tent map [18], the following form, resembles the logistic map. It generates chaotic sequences in (0,1) assuming

$$cm_{k+1} = \begin{cases} cm_k / 0.7 & cm_k < 0.7 \\ 10/3 cm_k (1 - cm_k) & otherwise \end{cases} \tag{14}$$

4.3. Sinusoidal map

This map [17] is represented by

$$vm_{k+1} = a \cdot cm_k^2 \sin(\pi \cdot cm_k) \tag{15}$$

For a = 2.3 and it has the following simplified form

$$cm_{k+1} = \sin(\pi \cdot cm_k) \tag{16}$$

It generates chaotic sequence in (0, 1).

4.4. Liebovtech map

As the last example of chaotic maps, Liebovitch map [19], consisting of three piecewise linear segments on non-overlapping subintervals on the interval (0,1). This map is defined by the following equations

$$x_{k+1} = \begin{cases} \alpha_1 x_k & 0 < x_k \le d_1, \\ \dfrac{d_2 - x_k}{d_2 - d_1} & d_1 < x_k \le d_2, \\ 1 - \alpha_2 (1 - x_k) & d_2 < x_k \le 1 \end{cases} \tag{17}$$

where d_1, $d_2 \in (0,1)$ with $d_1 < d_2$ and

$$\begin{aligned} \alpha_1 &= \frac{d_2}{d_1}\left(1 - (d_2 - d_1)\right), \\ \alpha_2 &= \frac{1}{d_2 - 1}\left((d_2 - 1) - d_1(d_2 - d_1)\right). \end{aligned} \tag{18}$$

5. PRACTICAL OPTIMIZATION PROBLEMS

In order to compare the variants of the new method, some well-known practical optimization problems are considered from literature. Fifty different runs for each setting with completely different initial conditions are used. Then, the statistical measures such as mean objective values and their standard deviations are utilized to measure the performance of the algorithm, rather than relying simply on a few runs. The simulations for map limits of k_a as (0,0.5) results in a better performance while for the other generated maps, no changes are observed. For all examples, the size of the problem is set to 30 CPs, obtained using some extensive sensitivity studies of the population size. With a fixed number of CPs at each run, the examples are optimized within 250 iterations for these examples. This means that the number of function evaluations is set to 7,500. The explanations of practical optimization problems as well as the obtained results are presented in the following sub-sections.

5.1. Design of an I-shaped beam

The goal is to minimize the vertical deflection of an I-beam as shown in Figure 2. This example is modified from the original problem reported in Ref. [20]. It simultaneously satisfies the cross-section area and stress constraints under the given loads.

Minimize the vertical deflection $f(x) = PL^3/48EI$ when the length of the beam (L) and modulus of elasticity (E) are 5200 *cm* and 523104 kN/cm^2, respectively. Thus, the objective function of the problem is considered to be as follows

$$\text{Minimize:} \quad f(b,h,t_w,t_f) = \frac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^{\,3}}{6} + 2bt_f\left(\frac{h-tf}{2}\right)^2} \tag{19}$$

subjected to a cross section area less than 300 cm^2.

$$g_1 = 2bt_w + t_w(h-2t_f) \le 300 \tag{20}$$

For the allowable bending stress of the beam taken as 56 kN/cm2, the stress constraint will be as

$$g_2 = \frac{18h\times10^4}{t_w(h-2t_f)^3 + 2bt_w\left(4t_f^{\,2} + 3h(h-2t_f)\right)} + \frac{15b\times10^3}{(h-2t_f)t_w^{\,3} + 2t_w b^3} \le 6 \tag{21}$$

where the initial design space is specified by

$$\begin{aligned} 10 \le h \le 80, \\ 10 \le b \le 50, \\ 0.9 \le t_w \le 5\ , \\ 0.9 \le t_f \le 5. \end{aligned} \tag{22}$$

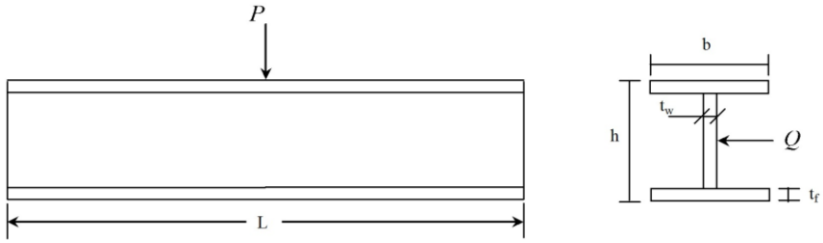

Figure 2. An I-shaped beam (P = 5600 *kN* and Q=550 *kN*)

The statistical results of the I-shaped beam problem obtained by the CSS and the variants of the CCSS algorithms are collected in Table 1. The all of the CCSS-1 methods improve the performance of the algorithm comparing to the standard CSS. The mean and worst values for the CCSS-1 somewhat are reduced while the standard deviations are improved considerably. The CCSS-2 and CCSS-3 algorithms with Sinusoidal and Tent maps have the better standard deviations. Almost all the best Statistical results (containing best result, best mean, best worst, and best standard deviations) are belonged to the CCSS-3 using Tent map among all of the algorithms. Although the differences between the best and mean results of the CCSS algorithms and the CSS are small, however the standard deviations as well as the reliability of the algorithm are improved considerably by utilizing the chaotic maps.

Table 1. Statistical results of the I-shaped beam problem for the CCSS algorithms.

Chaotic Map	Best	Mean	Worst	Std. Dev.
ECSS	0.013086	0.013181	0.013622	1.31E-04
CCSS-1				
Logistic map	0.013078	0.013121	0.013192	3.25E-05
Tent map	0.013076	0.013103	0.013272	4.34E-05
Sinusoidal map	0.013076	0.013105	0.013167	2.51E-05
Liebovtech map	0.013082	0.013142	0.013235	4.68E-05
CCSS-2				
Logistic map	0.013082	0.013126	0.013203	3.36E-05
Tent map	0.013081	0.013107	0.013162	2.32E-05
Sinusoidal map	0.013090	0.013126	0.013168	2.46E-05
Liebovtech map	0.013079	0.013116	0.013225	3.51E-05
CCSS-3				
Logistic map	0.013082	0.013128	0.013368	6.32E-05
Tent map	**0.013075**	**0.013095**	**0.013140**	**1.83E-05**
Sinusoidal map	0.013079	0.013109	0.013177	2.56E-05
Liebovtech map	0.013077	0.013156	0.013376	8.48E-05

5.2. Design of a Tubular column

Figure 3 presents a uniform column of tubular section to carry a compressive load of P = 2500 *kgf* at minimum cost [21]. The column is made of a material with a yield stress σ_y of 500 kgf/cm^2, a modulus of elasticity E of 0.85 × 106 kgf/cm^2, and a density ρ equal to 0.0025 kgf/cm^3. The length L of the column is 250 *cm*. The stress included in the column should be less than the buckling stress (constraint g_1) and the yield stress (constraint g_2). The mean diameter of the column is restricted to a value between 2 and 14 *cm* (constraint g_3 and g_4), and columns with thickness outside the range 0.2 – 0.8 *cm* are not commercially available (constraint g_5 and g_6). The cost of the column includes the material and construction costs. This cost is taken as the objective function. The optimization model of this problem is given

as follows

$$\text{Minimize:} \quad f(d,t) = 9.8dt + 2d \tag{23}$$

Subject to

$$g_1 = \frac{P}{\pi dt\sigma_y} - 1 \le 0 \tag{24}$$

$$g_2 = \frac{8PL^2}{\pi^3 Edt\left(d^2 + t^2\right)} - 1 \le 0 \tag{25}$$

$$g_3 = \frac{2.0}{d} - 1 \le 0 \tag{26}$$

$$g_4 = \frac{d}{14} - 1 \le 0 \tag{27}$$

$$g_5 = \frac{0.2}{t} - 1 \le 0 \tag{28}$$

$$g_6 = \frac{t}{0.8} - 1 \le 0 \tag{29}$$

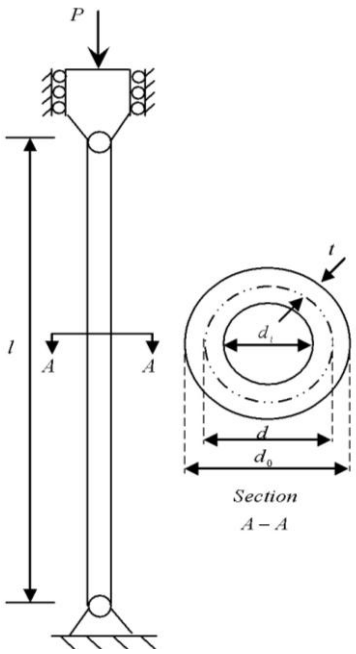

Figure 3. A tubular column

Table 2 presents the statistical results of the tubular column problem. Many statistical measures justify the superiority of the proposed methods in compression to the standard CSS. The low standard value of the proposed methods ensures the degree of consistency in producing the global optimal value. The results of the table show the superior of the CCSS-3 and CCSS-2 method (with Tent and Sinusoidal maps) to the other CCSS approaches. The CCSS-3 can find the best minimum value and the CCSS-2 is capable of reaches the best average and standard deviation.

Table 2. Statistical results of the tubular column problem for the CCSS algorithms.

Chaotic Map	Best	Mean	Worst	Std. Dev.
ECSS	26.62	27.35	29.35	2.35
CCSS-1				
Logistic map	26.58	27.01	27.91	1.44
Tent map	26.54	27.12	27.81	1.53
Sinusoidal map	**26.53**	27.08	27.55	1.33
Liebovtech map	26.54	27.28	28.12	1.43
CCSS-2				
Logistic map	26.57	27.00	27.90	1.35
Tent map	26.56	**26.85**	27.38	**1.23**
Sinusoidal map	26.55	26.88	**27.36**	1.28
Liebovtech map	26.58	26.95	27.66	1.36
CCSS-3				
Logistic map	26.56	27.28	28.35	1.85
Tent map	**26.53**	26.92	**27.36**	1.32
Sinusoidal map	**26.53**	27.00	27.65	1.33
Liebovtech map	26.57	27.30	28.33	1.92

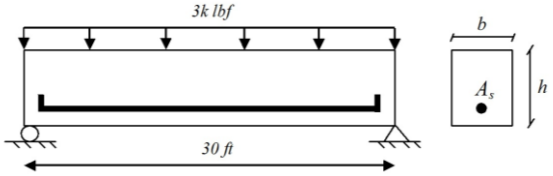

Figure 4. A reinforced concrete beam.

5.3. Design of a reinforced concrete beam

A simplified optimization of the total cost of a reinforced concrete beam, shown in Figure 4, was presented by Amir and Hasegawa [22]. The beam is assumed to be simply supported

with a span of 30 *ft* subjected to a live load of 2.0 *klbf* and a dead load of 1.0 *klbf* including the weight of the beam. The concrete compressive strength (Φ_c) is 5 *ksi*, the yield stress of the reinforcing steel (Φ_y) is 50 *ksi*. The cost of concrete is 0.02 *\$/in^2/linear ft* and the cost of steel is 1.0 *\$/in^2/linear ft*. It is required to determine the area of the reinforcement A_s, the width of the beam *b*, and the depth of the beam *h*, such that the total cost of structure is minimized. Herein, the cross-sectional area of the reinforcing bar A_s is taken as a discrete type variable that must be chosen from the standard bar dimensions listed in [22]. The width of concrete beam *b* is assumed to be an integer variable. The variable *h* denoting the depth of the beam is a continuous variable. The effective depth is assumed to be 0.8x2.0.

The structure should be proportioned to have a required strength based upon the ACI building code 318-77 as follows:

Table 3. Statistical results of the reinforced concrete beam for the CCSS algorithms.

Chaotic Map	Best	Mean	Worst	Std. Dev.
ECSS	364.05	480.34	670.25	85.35
CCSS-1				
Logistic map	363.4	440.58	560.23	50.21
Tent map	361.20	435.36	552.32	48.36
Sinusoidal map	361.01	432.32	545.56	46.32
Liebovtech map	364.00	450.23	570.61	58.98
CCSS-2				
Logistic map	363.56	439.89	561.35	52.36
Tent map	362.14	435.23	548.63	47.68
Sinusoidal map	363.35	438.36	550.36	48.63
Liebovtech map	362.23	442.32	570.35	56.32
CCSS-3				
Logistic map	362.62	440.35	570.32	55.32
Tent map	**360.52**	**425.32**	522.35	45.62
Sinusoidal map	361.56	435.23	**520.35**	**42.32**
Liebovtech map	362.96	445.36	572.32	50.23

$$Mu = 0.9A_s\sigma_y(0.8h)\left(1.0 - 0.59\frac{A_s\sigma_y}{0.8bh\sigma_c}\right) \geq 1.4M_d + 1.7M_l \tag{30}$$

in which M_u, M_d and M_l are the flexural strength, dead load and live load moments of the beam, respectively. In this case, M_d = 1350 *in.kip* and M_l = 2700 *in.kip*. The depth to width ratio of the beam is restricted to be less than or equal to 4. This optimization problem can be expressed as

$$\text{Minimize: } f(A_s, b, h) = 2.9A_s + 0.6bh \tag{31}$$

subject to

$$g_1 = \frac{h}{b} - 4 \le 0 \tag{32}$$

$$g_2 = 180 + 7.375\frac{A_s^{\,2}}{b} - A_s h \le 0 \tag{33}$$

The variables bound are: A_s : {6.0, 6.16, 6.32, 6.6, 7.0, 7.11, 7.2, 7.8, 7.9, 8.0, 8.4} in^2, b: {28, 29, 30, 31, …, 38, 39, 40} in, and $5 \le h \le 10$ in. The constrained functions of g_1 and g_2 are the same as derived by Amir and Hasegawa [22].

The results obtained for this example is presented in Table 3. The CCSS methods have shown better performance than the standard CSS method. The CCSS-3 performs better when the statistical analyses are compared. Similar to the previous example, The CCSS-3 (with Tent and Sinusoidal maps) have the best results while CCSS-2 methods are placed in the second place.

6. CONCLUSION

Different chaotic-based CSS algorithms are developed where the random nature of chaotic maps are utilized to adapt the parameters of the CSS algorithm. Three CCSS algorithms contain: CSS-1 (the kind of the forces as well as the probability function are determined by using chaotic maps), CCSS-2 (The coefficients of the force and velocity are determined chaotically) and CCSS3 (all required parameters are determined chaotically). Three civil engineering problems are considered to investigate the methods numerically. Performances are assessed on the basis of the best fitness values and the statistics results of the new approaches from 50 runs with different seeds. The results show that:

- Almost all the CCSS are more reliable compared to the standard CSS; this is because of decreasing standard deviations of the examples.
- When all parameters are determined chaotically, most efficient algorithm (CCSS-3) is obtained.
- Tent and Sinusoidal maps are known as the most useful maps to be used in the CCSS algorithms.
- Difficult civil engineering problems can be solved by using CCSS without needing to find suitable parameters of the algorithm.

REFERENCES

1. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mech*, 2010; **213**(3-4), 267-89.

2. Kaveh A, Talatahari S. A charged system search with a fly to boundary method for discrete optimum design of truss structures. *Asian J Civil Eng*, 2010; **11**(3): 277-93.
3. Kaveh A, Talatahari S. Optimal design of skeletal structures via the charged system search algorithm, *Struct Multidiscip Opt*, 2010; **41**(6): 893-911.
4. Kaveh A, Talatahari S. Charged system search for optimum grillage systems design using the LRFD-AISC code. *J Construct Steel Res*, 2010; **66**(6): 767-71.
5. Kaveh A, Talatahari S. Geometry and topology optimization of geodesic domes using charged system search. *Struct Multidiscip Opt*, 2011; **43** (2): 215-29.
6. Kaveh A, Talatahari S. An enhanced charged system search for configuration optimization using the concept of fields of forces. *Struct Multidiscip Opt*, 2011; **43**(3): 339-51.
7. Talatahari S, Kaveh A, Sheikholeslami R. Engineering design optimization using chaotic enhanced charged system search algorithms. *Acta Mech*, 2012; **223**(10):2269-85.
8. Talatahari S, Kaveh A, Sheikholeslami R. An efficient charged system search using chaos for optimization problems. *Int J Optim Civil Eng*, 2011; **1**(2):305-25.
9. Gharooni-fard G, Moein-darbari F, Deldari H, Morvaridi A. Scheduling of scientific workflows using a chaos-genetic algorithm. *Procedia Comput Sci*, 2010;**1**:1445–54.
10. Alatas B. Chaotic harmony search algorithms. *Appl Math Comput*, 2010;**216**:2687–99.
11. Mingjun J, Huanwen T. Application of chaos in simulated annealing. *Chaos Soliton Fract*, 2004;**21**:933–41.
12. Gandomi AH, Yun GJ, Yang XS, Talatahari S. Combination of chaos and accelerated particle swarm optimization. *Commun Nonlinear Sci Numer Simulat*, 2013;**18**(2):327-40.
13. Talatahari S, Farahmand Azar B, Sheikholeslami R, Gandomi AH. Imperialist competitive algorithm combined with chaos for global optimization. *Commun Nonlinear Sci Numer Simulat*, 2012;**17**:1312–9.
14. Gandomi AH, Yang X-S, Talatahari S, Alavi AH. Firefly Algorithm with Chaos, *Commun Nonlinear Sci Numer Simulat*, 2013; **18**(1):89-98.
15. Schuster GG. Deterministic chaos: An introduction (2nd ed.). Federal Republic of Germany: Physick-Verlag, GmnH, Weinheim 1998.
16. Coelho LS, Mariani VC. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Syst Appl*, 2008; **34**: 1905–13.
17. May RM. Simple mathematical models with very complicated dynamics. *Nature* 1976; **261**: 459.
18. Peitgen H, Jurgens H, Saupe D. *Chaos and Fractals*. Springer, Berlin, Germany, 1992.
19. Erramilli A, Singh RP, Pruthi P. 1994. Modeling packet traffic with chaotic maps, Royal Institute of Technology, Stockholm-Kista, Sweden.
20. Gold S, Krishnamurty S. Trade-offs in Robust Engineering Design. *Proceedings of the ASME Design Engineering Technical Conferences*, Saramento, 1997.
21. Rao SS. *Engineering Optimization: Theory and Practice* (2nd ed.), John Wiley & Sons, Chichester, UK, 1996.
22. Amir HM, Hasegawa T. Nonlinear mixed-discrete structural optimization. *J Struct Eng*,1989; **115**(3):626-45.