

Research Article

Chaotic Charged System Search with a Feasible-Based Method for Constraint Optimization Problems

B. Nouhi,¹ S. Talatahari,² H. Kheiri,¹ and C. Cattani³

¹ Department of Mathematical Sciences, University of Tabriz, Tabriz, Iran

² Marand Faculty of Engineering, University of Tabriz, Tabriz, Iran

³ Department of Mathematics, University of Salerno, Via Ponte Don Melillo, 84084 Fisciano, Italy

Correspondence should be addressed to S. Talatahari; talatahari@tabrizu.ac.ir

Received 15 December 2012; Accepted 23 January 2013

Academic Editor: Shengyong Chen

Copyright © 2013 B. Nouhi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently developed chaotic charged system search was combined to feasible-based method to solve constraint engineering optimization problems. Using chaotic maps into the CSS increases the global search mobility for a better global optimization. In the present method, an improved feasible-based method is utilized to handle the constraints. Some constraint design examples are tested using the new chaotic-based methods, and the results are compared to recognize the most efficient and powerful algorithm.

1. Introduction

The charged system search (CSS) is known as one of the efficient optimization algorithms among metaheuristic algorithms in recent years. The large number of researches on applying and enhancing this method proves this [1–11]. This algorithm uses the governing laws of electrostatics in physics and the governing laws of motion from the Newtonian mechanics [1].

In physics, the space surrounding an electric charge has a property known as the electric field. This field exerts a force on other electrically charged objects. The electric field surrounding a point charge is specified by Coulomb's law. Coulomb confirmed that the electric force between any two small charged spheres is inversely proportional to the square of the separation distance between the particles directed along the line joining them and proportional to the product of the charges of the two particles. Also, the magnitude of the electric field at a point inside a charged sphere can be obtained using Gauss's law that is proportional to the separation distance between the particles. In the CSS, charged particles (CPs) or solution candidates are treated as a charged sphere that can exert electrical forces on each other according to the Coulomb and Gauss laws of electrostatics. The resultant force acts on each CP creating an acceleration according to Newton's second law by which, in combination

with Newtonian mechanics, the position of each CP can be determined [1].

Recently, Talatahari et al. [7] developed chaotic charged system search algorithms (CCSS) based on the idea of using chaotic systems instead of random processes. In the CCSS algorithms, the role of randomness can be played by a chaotic dynamics. Experimental studies show the benefits of using chaotic signals instead of random signals. Nine chaos-based CSS methods were developed, and then, for each variant, the performance of ten different chaotic maps was investigated to identify the most powerful variant. As suggested in [7], here more elaborated experiments are performed to discover the adaptive algorithm for solving optimization problems. Also, this paper combines the most powerful CCSS algorithms with feasible-based method in order to solve constraint engineering design problems. In order to evaluate these algorithms, some well-studied numerical engineering examples are analyzed using some best CCSS methods and compared to the other methods.

2. Standard and Chaotic-Based Charged System Search Algorithm

2.1. Standard Charged System Search Algorithm. The charged system search contains a number of charged particle (CPs)

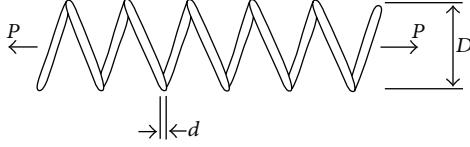


FIGURE 1: Schematic of tension/compression string.

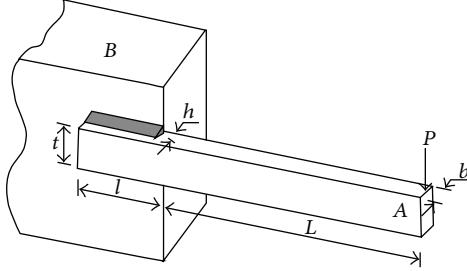


FIGURE 2: Schematic of welded beam structure.

where each one is treated as a charged sphere and can insert an electric force to the others. The pseudocode for the CSS algorithm is summarized as follows [1].

Step 1 (initialization). The magnitude of charge for each CP is defined as

$$q_i = \frac{\text{fit}(i) - \text{fitworst}}{\text{fitbest} - \text{fitworst}} \quad i = 1, 2, \dots, N, \quad (1)$$

where fitbest and fitworst are the best and the worst fitness of all the CPs, fit(i) represents the fitness of the agent i , and N is the total number of CPs. The separation distance r_{ij} between two charged particles is defined as follows:

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

where \mathbf{X}_i and \mathbf{X}_j are the positions of the i th and j th CPs, respectively, \mathbf{X}_{best} is the position of the best current CP, and ε is a small positive number. The initial positions of CPs are determined randomly as

$$x_{i,j}^{(0)} = x_{i,\min} + \text{rand}_{ij} \cdot (x_{i,\max} - x_{i,\min}), \quad i = 1, 2, \dots, N, \quad (3)$$

where $x_{i,j}^{(0)}$ 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.

Step 2 (CM creation). A number of the best CPs and the values of their corresponding fitness functions are saved in the charged memory (CM).

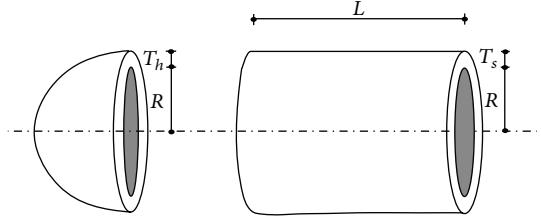


FIGURE 3: Schematic of pressure vessel.

Step 3 (the probability function and kind of the forces determination). The probability of moving each CP toward the others is determined using the following function:

$$p_{ij} = \begin{cases} 1, & \frac{\text{fit}(i) - \text{fitbest}}{\text{fit}(j) - \text{fit}(i)} > \text{rand} \vee \text{fit}(j) > \text{fit}(i), \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The kind of the forces can be attractive or repelling determined by using the force parameter ar_{ij} defined as

$$ar_{ij} = \begin{cases} +1, & k_t < \text{rand}_{ij}, \\ -1, & k_t > \text{rand}_{ij}, \end{cases} \quad (5)$$

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.

Step 4 (forces determination). The resultant force vector for each CP is calculated as

$$\mathbf{F}_j = q_j \sum_{i,i \neq j} \left(\frac{q_i}{a^3} r_{ij} \cdot i_1 + \frac{q_i}{r_{ij}^2} \cdot i_2 \right) ar_{ij} p_{ij} (\mathbf{X}_i - \mathbf{X}_j), \quad j = 1, 2, \dots, N$$

$$i_1 = 1, \quad i_2 = 0 \iff r_{ij} < a$$

$$i_1 = 0, \quad i_2 = 1 \iff r_{ij} \geq a.$$

Step 5 (solution construction). Each CP moves to the new position as

$$\mathbf{X}_{j,\text{new}} = \text{rand}_{j1} \cdot k_a \cdot \frac{\mathbf{F}_j}{m_j} \cdot \Delta t^2 + \text{rand}_{j2} \cdot k_v \cdot \mathbf{V}_{j,\text{old}} \cdot \Delta t + \mathbf{X}_{j,\text{old}}, \quad (7)$$

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

where k_a and k_v are the acceleration and the velocity coefficients, respectively, and rand_{j1} and rand_{j2} are two random numbers uniformly distributed in the range (0, 1).

Step 6 (CP position correction). If each CP swerves off the predefined bounds, its position is corrected using the harmony search-based handling approach.

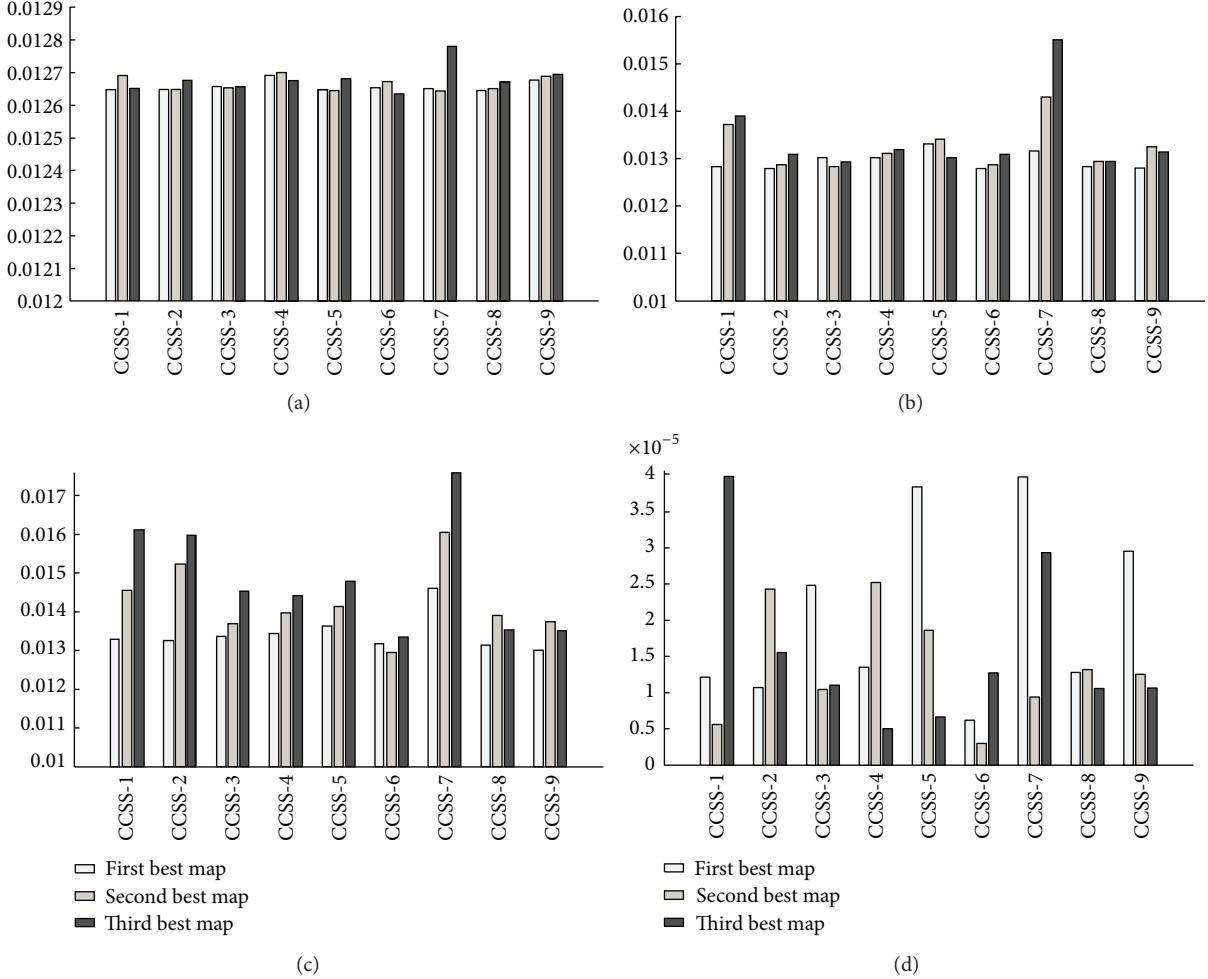


FIGURE 4: The results obtained by the CCSS methods for the string problem: (a) best results (b) mean of results, (c) worst result, and (d) Standard deviation.

Step 7 (CM updating). The better new vectors are included in the CM, and the worst ones are excluded from the CM.

Step 8 (terminating criterion control). Steps 3–7 are repeated until a terminating criterion is satisfied.

2.2. Chaotic Charged System Search Algorithm. To the best of our knowledge, random initialization of the CSS and the adjusted limit parameters may affect the performance of the algorithm and reduce or increase its convergence speed. In other words, the parameters of the algorithm such as k_t , k_a , and k_v are the key factors to control the balance of the exploration and exploitation of the algorithm. This problem was solved by introducing chaotic charged system search [7] where the suitable values or the initial population can be generated chaotically by using chaotic maps. Chaos is a deterministic, random-like process found in nonlinear, dynamical system, which is nonperiod, nonconverging, and bounded [12]. The nature of chaos looks to be random and unpredictable, possessing an element of regularity and randomness of a simple deterministic dynamical system,

and chaotic system may be considered as the source of randomness [13, 14].

The chaotic CSS algorithms, denoted by CCSS, can be obtained by using the values generated by a chaotic map instead of one or more random parameters needed in the CSS algorithm. Therefore, considering which parameter is defined chaotically, we can specify different algorithms. Table 1 summarizes different chaotic-based CSS algorithms. There are different chaotic maps to be used as listed in Table 2.

3. Present Algorithms

In [7], by hybridizing the charged system search and chaos, different methods were developed to solve numerical global optimization problems. The proposed approaches utilized different chaotic maps to adapt the parameters of the CSS algorithm. Nine chaotic CSS algorithms by using ten different chaotic maps are defined resulting in 90 different methods. These methods were then analyzed for the benchmark mathematical functions. The simulation resulting demonstrated that some tested CCSS approaches are efficient methods to

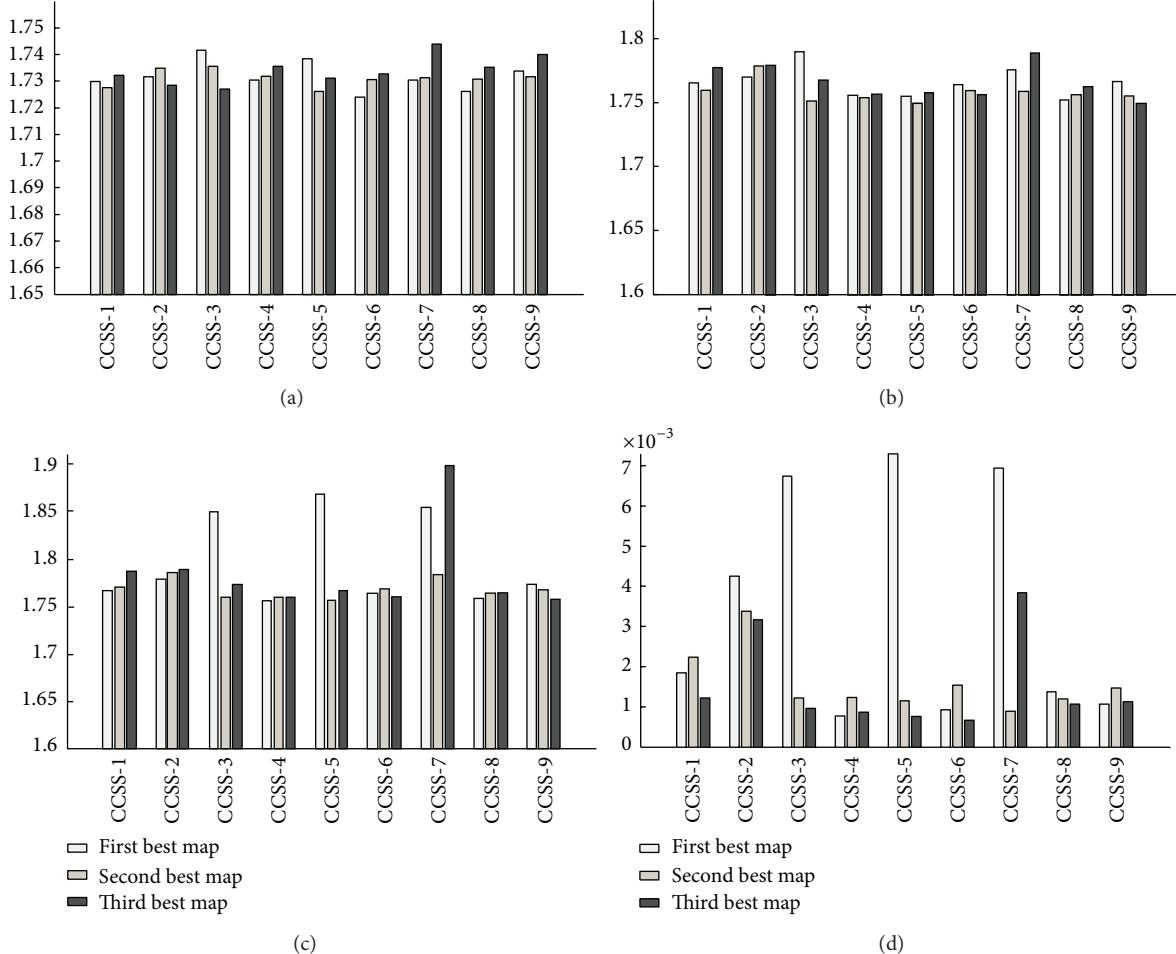


FIGURE 5: The results obtained by the CCSS methods for the welded beam problem: (a) best results (b) mean of results, (c) worst result, (d) Standard deviation.

TABLE 1: Different chaotic-based CSS algorithms.

Algorithm	Equation	Condition
CCSS-1	$x_{i,j} = x_{i,\min} + cm_{ij} \cdot (x_{i,\max} - x_{i,\min})$	The initial positions of CPs are determined chaotically.
CCSS-2	$ar_{ij} = \begin{cases} +1 & k_t < cm_{ij} \\ -1 & k_t > cm_{ij} \end{cases}$	The kind of the force is determined chaotically.
CCSS-3	$p_{ij} = \begin{cases} 1 & \frac{\text{fit}(i) - \text{fitbest}}{\text{fit}(j) - \text{fit}(i)} > cm_{ij} \vee \text{fit}(j) > \text{fit}(i) \\ 0 & \text{otherwise} \end{cases}$	The probability of moving each CP toward the others is determined chaotically.
CCSS-4	CCSS-2 + CCSS-3	The a_{ij} and p_{ij} are determined chaotically.
CCSS-5	$\mathbf{X}_{j,\text{new}} = cm_{j1} \cdot \frac{\mathbf{F}_j}{m_j} \cdot \Delta t^2 + \text{rand}_{j2} \cdot k_v \cdot \mathbf{V}_{j,\text{old}} \cdot \Delta t + \mathbf{X}_{j,\text{old}}$	The coefficient of the force is determined chaotically.
CCSS-6	$\mathbf{X}_{j,\text{new}} = \text{rand}_{j1} \cdot k_a \cdot \frac{\mathbf{F}_j}{m_j} \cdot \Delta t^2 + cm_{j2} \cdot \mathbf{V}_{j,\text{old}} \cdot \Delta t + \mathbf{X}_{j,\text{old}}$	The coefficient of the velocity is determined chaotically.
CCSS-7	$\mathbf{X}_{j,\text{new}} = cm_{j1} \cdot \frac{\mathbf{F}_j}{m_j} \cdot \Delta t^2 + cm_{j2} \cdot \mathbf{V}_{j,\text{old}} \cdot \Delta t + \mathbf{X}_{j,\text{old}}$	The coefficients of the force and velocity are determined chaotically.
CCSS-8	CCSS-4 + CCSS-7	The initial positions of CPs are determined randomly, and the rest random generators are placed to chaotic maps.
CCSS-9	CCSS-1 + CCSS-8	All random generators are placed to chaotic maps.

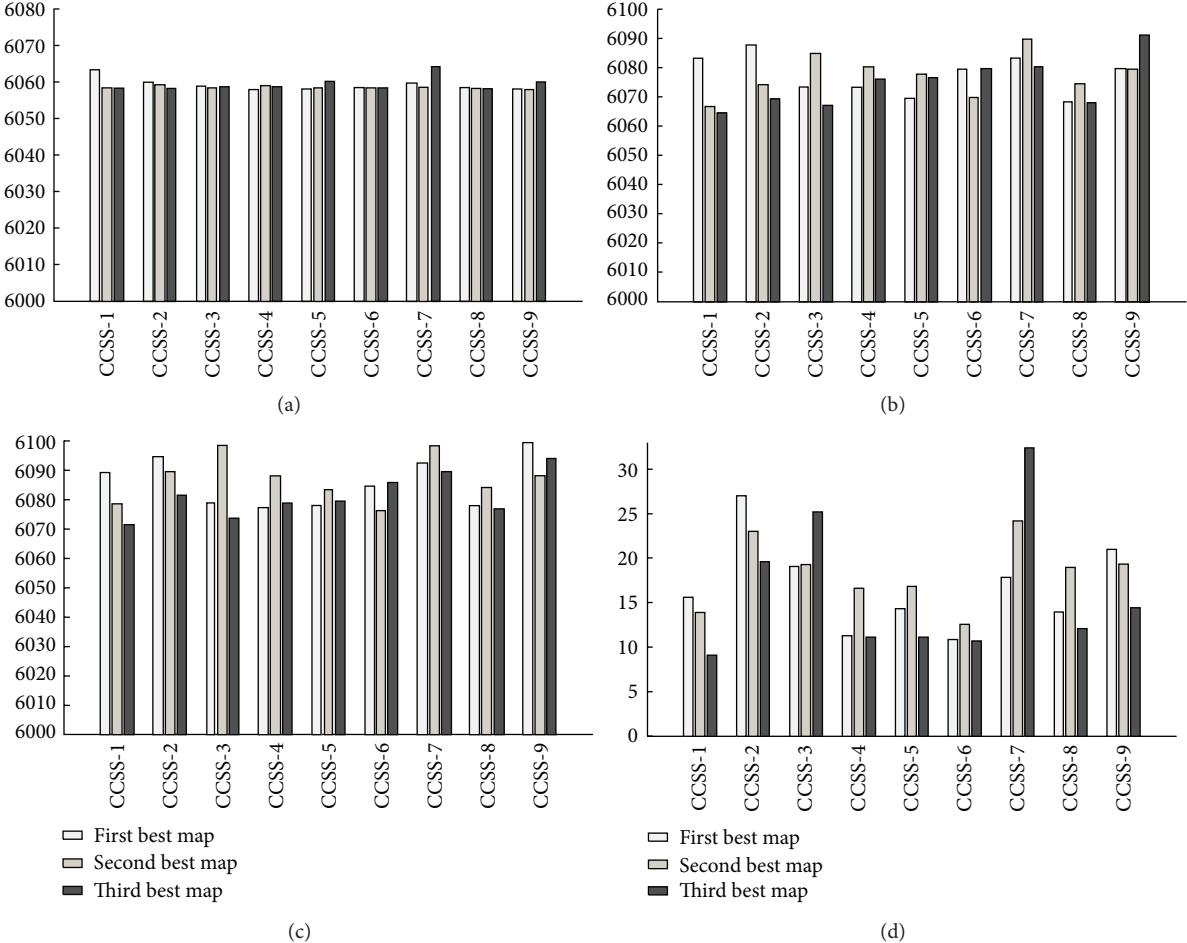


FIGURE 6: The results obtained by the CCSS methods for the pressure vessel problem: (a) best results (b) mean of results, (c) worst result, and (d) Standard deviation.

explore the search space and discover the global solution. From the statistical investigation, it was shown that the CCSS-8, utilizing the chaotic maps, instead of a_{ij} , p_{ij} , k_v , and k_a , has better performance than other approaches. The second best methods were CCSS-6 and CCSS-9. In order to simplify determining the most adaptive maps, for each algorithm three maps are chosen as the best ones as summarized in Table 3. In this paper, the selected maps are utilized for solving constraint engineering design examples.

On the other hand, it is necessary to handle the constraints of the problem by using a suitable method [20–22]. The feasible-based approach [23] deals with constrained search spaces by using the separation of constraints and objectives. In this method, the idea is to avoid the combination of the value of the objective function and the constraints of a problem to assign fitness, like when using a penalty function [23–26]. Kaveh and Talatahari [27] have presented a modified feasible-based method which employs the following four rules as.

Rule 1. Any feasible solution is preferred to any infeasible solution.

Rule 2. Infeasible solutions with slight violations of the constraints are treated as feasible ones.

Rule 3. Between two feasible solutions, the one with better objective function value is preferred.

Rule 4. Between two infeasible solutions, the one having smaller sum of constraint violations is preferred.

By using the first and fourth rules, the search tends to the feasible region rather than infeasible region, and by employing the third rule, the search tends to the feasible region with good solutions [28]. For most of the engineering optimization problems, the global minimum locates on or close to the boundary of a feasible design space. By applying Rule 2, the CPs approach to the boundaries and can go near to the global minimum in a great probability.

If the location of CPs become out of the variable boundaries, the solutions cannot be used. In this paper, using the harmony search-based handling approach, this problem is dealt with. According to this mechanism, any component

TABLE 2: Different utilized chaotic maps.

Name	Equation	Condition
Logistic map [15]	$cm_{k+1} = a \cdot cm_k(1 - cm_k)$	$cm_o \in (0, 1), cm_o \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and $a = 4$
Tent map [16]	$cm_{k+1} = \begin{cases} \frac{cm_k}{0.7} & cm_k < 0.7 \\ \frac{10}{3cm_k(1 - cm_k)} & \text{otherwise} \end{cases}$	
Sinusoidal map [16]	$cm_{k+1} = a \cdot cm_k^2 \sin(\pi \cdot cm_k)$	$a = 2.3$ and $cm_o = 0.7$
Gauss map [16]	$cm_{k+1} = \begin{cases} 0 & cm_k = 0 \\ \frac{1}{cm_k \bmod(1)} & \text{otherwise} \end{cases}$	$\frac{1}{cm_k \bmod(1)} = \frac{1}{cm_k} - \left[\frac{1}{cm_k} \right]$
Circle map [17]	$cm_{k+1} = cm_k + b - \left(\frac{a}{2\pi} \right) \sin(2\pi \cdot cm_k) \bmod(1)$	$a = 0.5$ and $b = 0.2$
Sinus map	$cm_{k+1} = 2.3(cm_k)^2 \sin(\pi \cdot cm_k)$	
Hénon map	$cm_{k+1} = 1 - a \cdot cm_k^2 + b \cdot cm_{k-1}$	$a = 1.4$ and $b = 0.3$
Ikeda map [18]	$x_{n+1} = 1 + 0.7(x_n \cos(\theta_n) - y_n \sin(\theta_n))$ $y_{n+1} = 0.7(x_n \sin(\theta_n) + y_n \cos(\theta_n))$	$\theta_n = 0.4 - \frac{6}{1 + x_n^2 + y_n^2}$
Liebovtech map [19]	$x_{k+1} = \begin{cases} \alpha_1 x_k & 0 < x_k \leq d_1, \\ \frac{d_2 - x_k}{d_2 - d_1} & d_1 < x_k \leq d_2, \\ 1 - \alpha_2(1 - x_k) & d_2 < x_k \leq 1 \end{cases}$	$\alpha_1 = \frac{d_2}{d_1}(1 - (d_2 - d_1)),$ $\alpha_2 = \frac{1}{d_2 - 1}((d_2 - 1) - d_1(d_2 - d_1)).$
Zaslavski map [16]	$y(k+1) = [y(k) + v + az(k+1)] \bmod 1$ $z(k+1) = \cos(2\pi y(k)) + e^{-r} z(k)$	$v = 400, r = 3, a = 12.6695$ and $z(t) \in [-1.0512, 1.0512]$

TABLE 3: Best maps for each CCSS.

Best maps	CCSS								
	1	2	3	4	5	6	7	8	9
1	Tent	Tent	Sinusoidal	Tent	Liebovtech	Liebovtech	Liebovtech	Sinus	Sinus
2	Circle	Gauss	Ikeda	Ikeda	Sinus	Sinusoidal	Zaslavski	Liebovtech	Tent
3	Sinus	Ikeda	Sinus/Gauss	Circle	Tent	Zaslavski	Sinus	Tent	Gauss

of the CP's vector violating the variable boundaries can be generated randomly from CM as

4. Numerical Engineering Problems

The population size $N = 20$ is sufficient for most of the problems while the maximum number of function evolutions for the examples is set to 4,000 (200 iterations). The explanation of the examples is presented in Section 4.1, and the performance of the CCSS algorithms to optimize these functions is investigated in the next subsection.

4.1. Description of the Examples. Three engineering design problems which have been previously solved using a variety of other techniques are considered to perform investigation on efficiency of the proposed algorithms. The description of these examples is as the following.

4.1.1. A Tension/Compression String Design Problem. This problem consists of minimizing the weight of a tension/compression spring subject to constraints on shear stress, surge frequency, and minimum deflection as shown

$$x_{i,j} = \begin{cases} \text{w.p. HMCR} ==> \text{select a new value} \\ \quad \text{for variable from CM} \\ \quad ==> \text{w.p. (1 - PAR)} \\ \quad \quad \text{do nothing} \\ \quad ==> \text{w.p. PAR choose} \\ \quad \quad \text{a neighboring value} \\ \text{w.p. (1 - HMCR)} ==> \text{select a new value} \\ \quad \text{randomly from} \\ \quad \text{the allowable list.} \end{cases} \quad (8)$$

in Figure 1. The design variables are the mean coil diameter $D(= x_1)$, the wire diameter $d(= x_2)$, and the number of active coils $N(= x_3)$. The problem is defined with details in [1].

4.1.2. A Welded Beam Design Problem. The welded beam structure, shown in Figure 2, is a practical design problem that often has been used as a benchmark problem for testing different optimization methods. The objective is to find the minimum fabricating cost of the welded beam subject to constraints on shear stress (τ), bending stress (σ), buckling load (P_c), end deflection (δ), and side constraint. There are four design variables, namely, $h(= x_1)$, $l(= x_2)$, $t(= x_3)$, and $b(= x_4)$. The detailed information about the constraints and objective function is presented in [1].

4.1.3. A Pressure Vessel Design Problem. A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure 3. The objective is to minimize the total cost, including the cost of material, forming, and welding.

The variables contain x_1 as the thickness of the shell (T_s), x_2 as the thickness of the head (T_h), x_3 as the inner radius (R), and x_4 as the length of cylindrical section of the vessel, not including the head (L). Details related to the objective and constraint functions are available in [1].

4.2. Experimental Results

4.2.1. Results for the Tension/Compression String Problem. For this example, performances are assessed on the basis of the best fitness values and the statistics results of the new approaches as reported in Figure 4. Simulation results show that all the proposed methods perform satisfactorily. However, the best feasible solutions obtained by the methods CCSS-8 (with Sinus map), CCSS-6 (with Zaslavski map), and CCSS-5 (with Sinus map) are the best results among the others. Meanwhile, the CSS-8 and CCSS-6 methods (almost with all the three maps) have better performance in relation to the mean and worst values. Moreover, the standard deviation of the results by the CSS-6 in 50 independent runs for this problem is the smallest among the others.

4.2.2. Results for the Welded Beam Problem. Figure 5 presents the statistical information of this problem obtained by the proposed algorithms. Clearly, the worst solutions belong to the CCSS-7, while the best ones belong to the CCSS-6 and CCSS-8. The CCSS-4 as well as CCSS-9 stand in the second place. Although some new methods have small differences in relation to best, mean, or worst results, obviously many of them perform very well in relation to the standard deviation. Similar to the previous example, the CCSS-6 has the best standard deviation compared to the other methods.

4.2.3. Results for the Pressure Vessel Problem. The obtained results using 9 variants of the presented algorithms are shown in Figure 6. Similar to the previous examples, many of the CCSS methods can find a suitable result successfully; however, the best solution found by CCSS-6 and CCSS-4 is better than the best solutions found by the other

techniques. Also, it can be seen that the average searching quality of the CCSS-8 (with Sinus and Tent maps) and CCSS-6 (with Sinusoidal map) algorithms is better than those of other methods. Many of the proposed methods improve the reliability of the algorithm by reducing the standard deviation values.

5. Conclusion Remarks

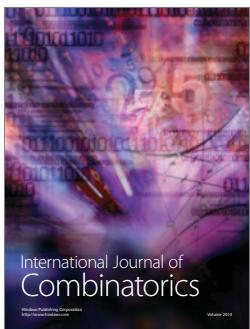
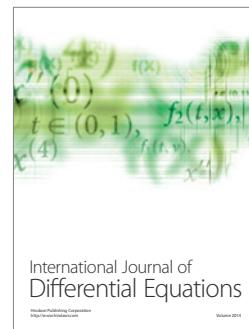
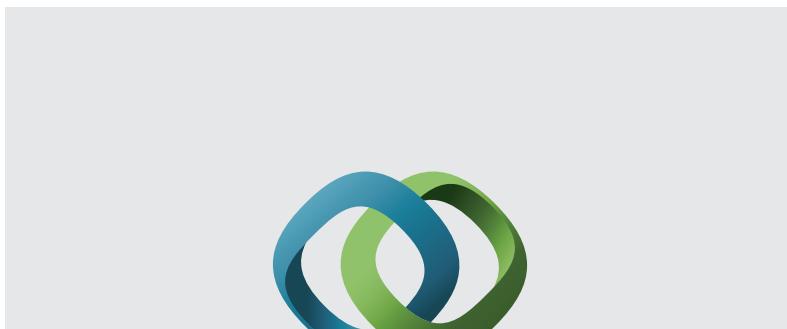
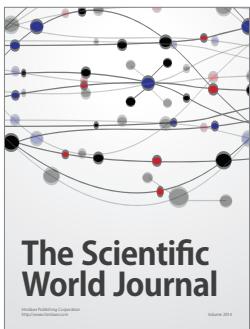
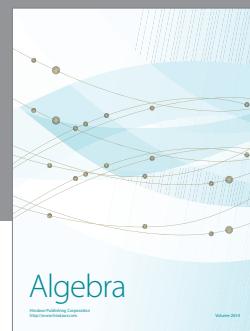
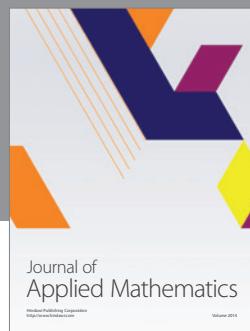
As suggested in [7] due to the superiorities of the CCSS methods, here more elaborated experiments are performed to discover the better methods which can be utilized in solving engineering problems. From 90 different chaotic charged system search algorithms (obtained by using 10 chaotic maps and 9 different methods), 27 most efferent methods (3 different chaotic maps for each algorithm) are chosen and investigated to solve constraint engineering problems.

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. Simulation results show that for all examples, the proposed methods perform satisfactorily. Almost all of the proposed methods improve the reliability of the algorithm by reducing the standard deviation values. From numerical results, it is clear that CCSS-6, in which the coefficient of the velocity is determined chaotically, is the most reliable algorithm having the smallest standard deviation values, while, the algorithm with chaotic coefficients for the force and velocity (CCSS-7) is the worst one. Meanwhile, the CSS-8 and CCSS-6 methods have better performance in relation to the best, mean, and worst values. In CCSS-8, all parameters of the algorithm are determined chaotically, but the initial positions of agents are defined randomly. To sum up, the coefficient of the velocity plays a key role in reliability of the algorithm and the results show that a chaotic velocity coefficient can improve the performance of the algorithm; in addition, using all chaotic parameters of the algorithm can improve the performance, as well; however, chaotic initialization has very small or even no influence on the final results. This chaotic charged model, which is a coupled system, for constraint optimization problems in future could also benefit from other coupled systems such as the kinetic models of competition and the corresponding hybrid competition models [29, 30].

References

- [1] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3-4, pp. 267–289, 2010.
- [2] A. Kaveh and S. Talatahari, "A charged system search with a fly to boundary method for discrete optimum design of truss structures," *Asian Journal of Civil Engineering*, vol. 11, no. 3, pp. 277–293, 2010.
- [3] A. Kaveh and S. Talatahari, "Optimal design of skeletal structures via the charged system search algorithm," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 893–911, 2010.
- [4] A. Kaveh and S. Talatahari, "Charged system search for optimum grillage system design using the LRFD-AISC code,"

- Journal of Constructional Steel Research*, vol. 66, no. 6, pp. 767–771, 2010.
- [5] A. Kaveh and S. Talatahari, “Geometry and topology optimization of geodesic domes using charged system search,” *Structural and Multidisciplinary Optimization*, vol. 43, no. 2, pp. 215–229, 2011.
 - [6] A. Kaveh and S. Talatahari, “An enhanced charged system search for configuration optimization using the concept of fields of forces,” *Structural and Multidisciplinary Optimization*, vol. 43, no. 3, pp. 339–351, 2011.
 - [7] S. Talatahari, A. Kaveh, and R. Sheikholeslami, “An efficient charged system search using chaos for optimization problems,” *International Journal of Optimization in Civil Engineering*, vol. 1, no. 2, pp. 305–325, 2011.
 - [8] S. Talatahari, R. Sheikholeslami, M. Shadfar, and M. Porbaba, “Charged system search algorithm for optimum design of gravity retaining walls subject to seismic loading,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 301628, 10 pages, 2012.
 - [9] S. Talatahari, A. Kaveh, and R. Sheikholeslami, “Engineering design optimization using chaotic enhanced charged system search algorithms,” *Acta Mechanica*, vol. 223, no. 10, pp. 2269–2285, 2012.
 - [10] S. Talatahari, A. Kaveh, and N. Mohajer Rahbari, “Parameter identification of Bouc-Wen model for MR fluid dampers using adaptive charged system search optimization,” *Journal of Mechanical Science and Technology*, vol. 26, no. 8, pp. 2523–2534, 2012.
 - [11] T. Niknam, F. Golestaneh, and M. Shafiei, “Probabilistic energy management of a renewable microgrid with hydrogen storage using self-adaptive charge search algorithm,” *Energy*, vol. 49, pp. 252–267, 2013.
 - [12] B. Alatas, “Chaotic bee colony algorithms for global numerical optimization,” *Expert Systems with Applications*, vol. 37, pp. 5682–5687, 2010.
 - [13] G. G. Schuster, *Deterministic Chaos: An Introduction*, Federal Republic of Germany: Physick, GmnH, Weinheim, Germany, 2nd edition, 1988.
 - [14] L. D. S. Coelho and V. C. Mariani, “Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization,” *Expert Systems with Applications*, vol. 34, no. 3, pp. 1905–1913, 2008.
 - [15] R. M. May, “Simple mathematical models with very complicated dynamics,” *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.
 - [16] H.-O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals*, Springer, Berlin, Germany, 1992.
 - [17] W. M. Zheng, “Kneading plane of the circle map,” *Chaos, Solitons and Fractals*, vol. 4, no. 7, pp. 1221–1233, 1994.
 - [18] U. Dressler and J. D. Farmer, “Generalized Lyapunov exponents corresponding to higher derivatives,” *Physica D. Nonlinear Phenomena*, vol. 59, no. 4, pp. 365–377, 1992.
 - [19] A. Erramilli, R. P. Singh, and P. Pruthi, *Modeling Packet Traffic with Chaotic Maps*, Royal Institute of Technology, Stockholm-Kista, Sweden, 1994.
 - [20] C. Cattani, S. Y. Chen, and G. Aldashev, “Information and modeling in complexity,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 868413, 4 pages, 2012.
 - [21] S. C. Lim, C. H. Eab, K. H. Mak, M. Li, and S. Y. Chen, “Solving linear coupled fractional differential equations by direct operational method and some applications,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 653939, 28 pages, 2012.
 - [22] P. Lu, S. Y. Chen, and Y. Zheng, “Artificial intelligence in civil engineering,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 145974, 22 pages, 2012.
 - [23] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, pp. 311–338, 2000.
 - [24] C. A. Coello Coello, “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11–12, pp. 1245–1287, 2002.
 - [25] S. Y. Chen and Y. F. Li, “Automatic sensor placement for model-based robot vision,” *IEEE Transactions on Systems, Man and Cybernetics B*, vol. 34, no. 1, pp. 393–408, 2004.
 - [26] Y. Zheng, H. Shi, and S. Chen, “Fuzzy combinatorial optimization with multiple ranking criteria: a staged tabu search framework,” *Pacific Journal of Optimization*, vol. 8, no. 3, pp. 457–472, 2012.
 - [27] A. Kaveh and S. Talatahari, “Engineering optimization with hybrid particle swarm and ant colony optimization,” *Asian Journal of Civil Engineering*, vol. 10, no. 6, pp. 611–628, 2009.
 - [28] E. M. Montes and C. A. C. Coello, “An empirical study about the usefulness of evolution strategies to solve constrained optimization problems,” *International Journal of General Systems*, vol. 37, no. 4, pp. 443–473, 2008.
 - [29] C. Cattani and A. Ciancio, “Hybrid two scales mathematical tools for active particles modelling complex systems with learning hiding dynamics,” *Mathematical Models and Methods in Applied Sciences*, vol. 17, no. 2, pp. 171–187, 2007.
 - [30] C. Cattani and A. Ciancio, “Separable transition density in the hybrid model for tumor-immune system competition,” *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 610124, 6 pages, 2012.



Submit your manuscripts at
<http://www.hindawi.com>

