INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2013; 3(4):617-633

A SURVEY OF CHAOS EMBEDDED META-HEURISTIC ALGORITHMS

R. Sheikholeslami[1] and A. Kaveh[*, †, 2]
[1]Department of Civil and Environmental Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
[2]Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

ABSTRACT

This article presents a comprehensive review of chaos embedded meta-heuristic optimization algorithms and describes the evolution of this algorithms along with some improvements, their combination with various methods as well as their applications. The reported results indicate that chaos embedded algorithms may handle engineering design problems efficiently in terms of precision and convergence and, in most cases; they outperform the results presented in the previous works. The main goal of this paper is to providing useful references to fundamental concepts accessible to the broad community of optimization practitioners.

Received: 25 April 2013; Accepted: 24 November 2013

KEY WORDS: Chaos theory**,** Meta-heuristics, Chaotic maps, Global optimization

1. INTRODUCTION

In nature complex biological phenomena such as the collective behavior of birds, foraging activity of bees or cooperative behavior of ants may result from relatively simple rules which however present nonlinear behavior with sensitivity to initial conditions. Such systems are generally known as “deterministic nonlinear systems” and the corresponding theory as “chaos theory”. Thus real world systems that may seem to be stochastic or random

[*]Corresponding author: A. Kaveh, School of Civil Engineering, Iran University of Science and Technology, Tehran-16, Iran.
[†]E-mail address: alikaveh@iust.ac.ir (A. Kaveh)

may present a nonlinear deterministic and chaotic behavior. 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, many meta-heuristic 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 meta-heuristic optimization techniques. Due to these common properties between chaos and optimization algorithms, simultaneous use of these concepts seems to improve the performance [1]. Seemingly the benefits of such combination is a generic for other stochastic optimization and experimental studies confirmed this; although, this has not mathematically been proved yet [2].

Recently, chaos and meta-heuristics have been combined in different studies for different purposes. Some of the works have intended to show the chaotic behaviors in the meta-heuristic algorithms. In some of the works, chaos has been used to overcome the limitations of meta-heuristics. Hence previous research can be classified into two types.

In the first type, chaos is inserted into the meta-heuristics instead of a random number generator, *i.e.*, the chaotic signals are used to control the value of parameters in the meta-heuristic's equations. The convergence properties of meta-heuristics are closely connected to the random sequence applied on their operators during a run. In particular, when starting some optimizations with different random numbers, experience shows that the results may be very close but not equal, and require also different numbers of generations to reach the same optimal value. The random numbers generation algorithms, on which most used meta-heuristics tools rely, usually satisfy on their own some statistical tests like chi-square or normality. However, there are no analytical results that guarantee an improvement of the performance indexes of meta-heuristics algorithms depending on the choice of a particular random number generator [3].

In the second type, chaotic search is incorporated into the procedures of the meta-heuristics in order to enrich the searching behavior and to avoid being trapped in local optimums. A traditional chaos optimization algorithm (COA) which is a stochastic search technique was proposed based on the advantages of chaos variables. The simple philosophy of the COA includes two main stages: firstly mapping from the chaotic space to the solution space, and then searching optimal regions using chaotic dynamics instead of random search [4]. However, COA also has some disadvantages. For example, in the large-scale optimization problems the efficiency of the algorithm will be very low and the COA often needs a large number of iterations to reach the global optimum.

The main contribution of this paper is to provide a state of the art review of the combination of chaos theory and meta–heuristics, and reports the evolution of these algorithms along with some improvements, their combinations with various methods as well as their applications.

2. AN OVERVIEW OF CHAOTIC SYSTEMS

In mathematic chaos is defined as “randomness” generated by simple deterministic systems. The randomness is a result of the sensitivity of chaotic systems to the initial conditions; it means that slight changes in the parameters or the starting values for the data lead to vastly different future behaviors, such as stable fixed points, periodic oscillations, bifurcations, and ergodicity. However, because the chaotic systems are deterministic, chaos implies order. A system can make the transformation from a regular periodic system to a complex chaotic system simply by changing one of the controlling parameters. Also a chaotic movement can go through every state in a certain area according to its own regularity, and every state is obtained only once [6]. An example of chaotic map is shown in Figure 1.

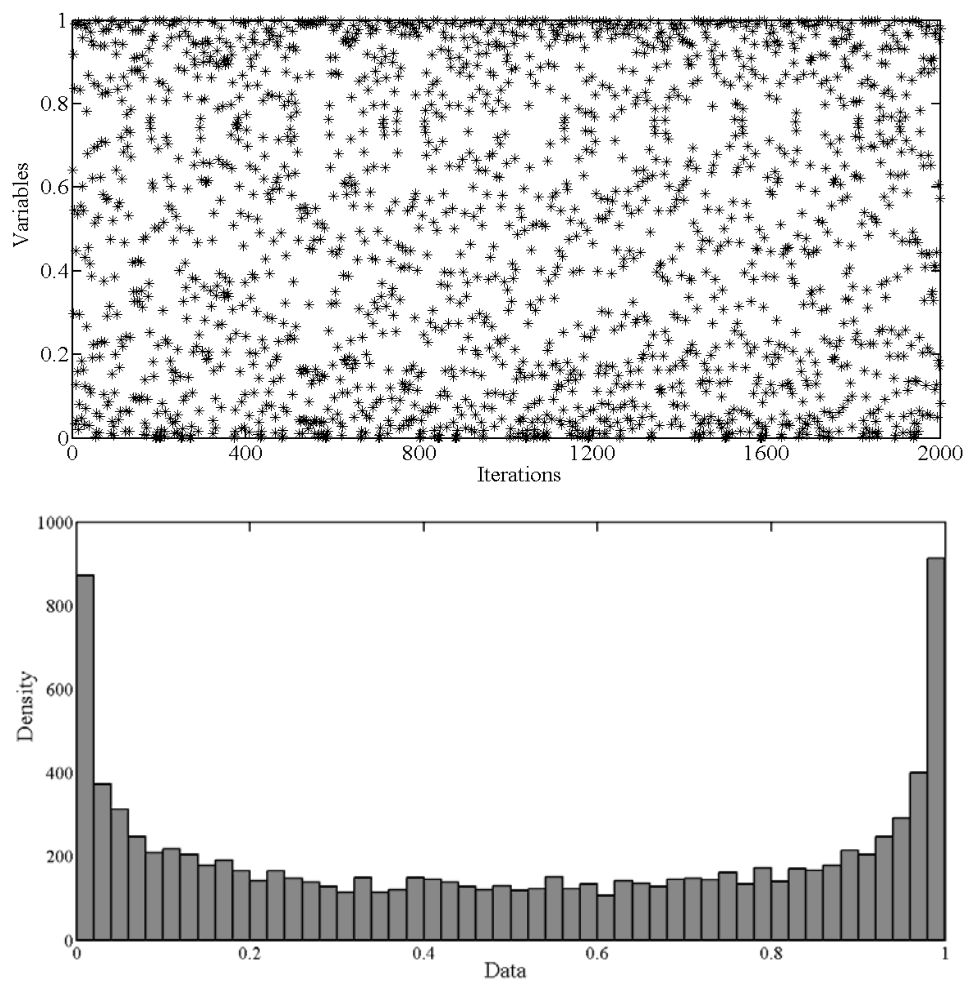

Figure 1. An example of chaotic map (Logistic map)

Considering a discrete-time series we can define chaos in the sense of Li-Yorke. A one-dimensional iterated map is based on a function of a real variable and takes the form

$$x_{t+1} = F(x_t) \tag{1}$$

where $x(t) \in \Re^n$, $t = 1, 2, 3, \ldots$ and F is a map from $\Re^n$ to itself.

Let $F^{(p)}$ denotes the composition of F with itself $p > 0$ times, then a point x is called a p-periodic point of F if $F^{(p)}(x) = x$ but $F^{(k)}(x) \neq x$ for all k such that $k \leq p$. In particular, a point x satisfying $F(x) = x$ is called a fixed point of F. The ε-neighborhood $N_\varepsilon(x)$ of a point x is defined by

$$N_\varepsilon(x) = \left\{ y \in \Re^n \mid \|x - y\| \leq \varepsilon \right\} \tag{2}$$

where ||.|| denotes the Euclidean norm in $\Re^n$. Then, we introduce the following definition of chaos in the sense of Li-Yorke [6]:

Definition 1. If a discrete-time series satisfies the following conditions, then it is called chaotic:

1) There exist a positive constant N such that for any $p \geq N$, F has a p-periodic point.

2) There exists an uncountable set $S \subset \Re^n$, which does not include any periodic point of F and satisfies the following conditions

(a) $F(S) \subset S$

(b) For any points $x, y \in S \; (x \neq y)$

$$\lim_{n \to \infty} \sup \left\| F^{(n)}(x) - F^{(n)}(y) \right\| > 0,$$

and for any $x \in S$ and any periodic point y of F,

$$\lim_{n \to \infty} \sup \left\| F^{(n)}(x) - F^{(n)}(y) \right\| > 0.$$

(c) There exists an uncountable subset $S_0 \subset S$ such that for any $x, y \in S_0$,

$$\lim_{n \to \infty} \inf \left\| F^{(n)}(x) - F^{(n)}(y) \right\| = 0$$

The set S in the above definition is called the scrambled set.

Then, it is well known that the existence of a fixed point called a snap-back repeller in a system implies that the system is chaotic in the sense of Li-Yorke [7]. So a system is chaotic if it contains infinitely many periodic orbits whose periods are arbitrarily large. This definition essentially is a result of Sarkovskii's theorem which was proved by the Russian mathematician A.N. Sarkovskii in 1964; however apparently presented in a famous paper by Li and Yorke [6] in which the word chaos first appeared in its contemporary scientific meaning [8].

A chaotic map can be used as spread-spectrum sequence for random number sequence. Chaotic sequences have been proven to be easy and fast to generate and store, and therefore there is no need for storaging long sequences. One needs only a few functions (chaotic maps) and few parameters (initial conditions) for very long sequences. Also an enormous number of different sequences can be generated simply by altering its initial condition. In addition, these sequences are deterministic and reproducible. The choice of chaotic sequences can justified theoretically by their unpredictability, corresponding to their spread-spectrum characteristic and ergodic properties [9]. Therefore when a random number is needed, 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 run. The literature is rich in chaotic time series sequences, some of them are listed in following subsections.

2.1. Logistic map

This map, whose equation appears in nonlinear dynamics of biological population evidencing chaotic behavior [10]

$$x_{k+1} = a x_k (1 - x_k) \tag{3}$$

In this equation, x_k is the kth chaotic number, with k denoting the iteration number. Obviously, $x_k \in (0,1)$ under the conditions that the initial $x_o \in (0,1)$ and that $x_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$. In the experiments a = 4 is used.

2.2. Tent map

Tent map [11] resembles the logistic map. It generates chaotic sequences in (0, 1) assuming the following form

$$x_{k+1} = \begin{cases} x_k / 0.7 & x_k < 0.7 \\ 10/3 x_k (1 - x_k) \ldots. & otherwise \end{cases} \tag{4}$$

2.3. Sinusoidal map

This iterator [10] is represented by

$$x_{k+1} = a x_k^2 \sin(\pi x_k) \tag{5}$$

For a = 2.3 and $x_o = 0.7$ it has the following simplified form:

$$x_{k+1} = \sin(\pi x_k) \tag{6}$$

It generates chaotic sequence in (0, 1).

2.4. Gauss map

The Gauss map is utilized for testing purpose in the literature [11] and is represented by

$$x_{k+1} = \begin{cases} 0 & x_k = 0 \\ 1/x_k \bmod(1)\ldots & otherwise \end{cases}$$
$$1/x_k \bmod(1) = \frac{1}{x_k} - \left[\frac{1}{x_k}\right] \tag{7}$$

Here, [x] denotes the largest integer less than x and acts as a shift on the continued fraction representation of numbers. This map also generates chaotic sequences in (0, 1).

2.5. Circle map

The Circle map [12] is represented by

$$x_{k+1} = x_k + b - (a/2\pi)\sin(2\pi x_k) \bmod(1) \tag{8}$$

With $a = 0.5$ and $b = 0.2$, it generates chaotic sequence in (0, 1).

2.6. Sinus map

Sinus map is defined as

$$x_{k+1} = 2.3(x_k)^{2\sin(\pi x_k)} \tag{9}$$

2.7. Henon map

This map is a nonlinear 2-dimensional map most frequently employed for testing purposes, and it is represented by

$$x_{k+1} = 1 - ax_k^2 + bx_{k-1} \tag{10}$$

The suggested parameter values are $a = 1.4$ and $b = 0.3$.

2.8. Ikeda map

An Ikeda map is a discrete-time dynamical system defined by [13]

$$x_{n+1} = 1 + 0.7\left(x_n \cos(\theta_n) - y_n \sin(\theta_n)\right),$$
$$y_{n+1} = 0.7\left(x_n \sin(\theta_n) + y_n \cos(\theta_n)\right),$$
$$\theta_n = 0.4 - \frac{6}{1 + x_n^2 + y_n^2} \tag{11}$$

2.9. Zaslavskii map

One of the interesting dynamic systems evidencing chaotic behavior is the Zaslavskii map [14], the corresponding equation is given by:

$$\begin{aligned} x_{k+1} &= x_k + v + \alpha y_{k+1} \pmod{1} \\ y_{k+1} &= \cos(2\pi x_k) + e^{-r} y_k \end{aligned} \tag{12}$$

where mod is the modulus after division and v = 400, r = 3, α = 12.6695. In this case, $y_t \in$[–1.0512, 1.0512].

3. USE OF CHAOTIC SYSTEMS IN META-HEURISTICS

In the artificial intelligence community, the term meta–heuristic was created and is now well accepted for general algorithms that represent a family of approximate optimization methods which are not limited to a particular problem. There were many attempts to give a rigorous mathematical definition of meta-heuristics. Here are some of them, accompanied by explanations.

1) "*They are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space.*" [15]

2) "*These methods can be defined as upper level general methodologies that can be used as guiding strategies in designing underlying heuristics to solve specific optimization problems.*" [16]

3) "*They are a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems with relatively few modifications to make them adapted to a specific problem.*" [17].

Design and implementation of such optimization methods has been at the origin of a multitude of contributions to the literature in the last 50 years. Genetic algorithms (GA) [18], simulated annealing (SA) [19], ant colony optimization (ACO) [20], particle swarm optimization (PSO) [21], harmony search algorithm (HS) [22], big bang-big crunch optimization (BB-BC) [23], imperialist competitive algorithm (ICA) [24], firefly algorithm (FA) [25], cuckoo search (CS) [26], charged system search algorithm (CSS) [27], magnetic charged system search algorithm (MCSS) [28], and ray optimization (RO) [29], are some familiar examples of meta-heuristics. Generally, a meta-heuristic algorithm uses two basic strategies while searching for the global optima; exploration and exploitation. The exploration enables the algorithm to reach at the best local solutions within the search space, and the exploitation provides the ability to reach at the global optimum solution which may exist around the local solutions obtained. In exploitation, the promising regions are explored more comprehensively, while in exploration the non-explored regions are visited to make sure that all the regions of the search space are fairly explored.

Due to common properties between chaos and meta-heuristic optimization algorithms, simultaneous use of these concepts seems to improve the performance and to overcome the limitations of meta-heuristics. The previous research can be categorized into two types. In the first type, chaotic system is inserted into the meta-heuristics instead of a random number generator for updating the value of parameters; and in the second type, chaotic search is incorporated into the procedures of the meta-heuristics in order to enrich the searching behavior and to avoid being trapped in local optimums using traditional chaos optimization algorithms (COA).

3.1 Chaotic update of internal parameters for meta-heuristics

For simulating complex phenomena, sampling, numerical analysis, decision making and in particular in meta-heuristic optimization, random sequences are needed with a long period and reasonable uniformity. On the other hand as mentioned before 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 sources of randomness [30-32].

However, meta-heuristics are non-typical; hence, the critical issue in implementing meta-heuristic methods is the determination of "proper" parameters which must be established before running these algorithms. The efficient determination of these parameters leads to a reasonable solution. That is why; these parameters may be selected chaotically by using chaotic maps. In this case, sequences generated from chaotic systems substitute random numbers for the parameters where it is necessary to make a random-based choice. By this way, it is intended to improve the global convergence and to prevent to stick on a local solution.

Alatas et al. [33] proposed different chaotic maps to update the parameters of PSO algorithm. This has been done by using of chaotic number generators each time a random number is needed by the classical PSO algorithm. Twelve chaos-embedded PSO methods have been proposed and eight chaotic maps have been analyzed in the unconstrained benchmark functions. The simulation results show that the application of deterministic chaotic signals may be a possible strategy to improve the performances of PSO algorithms. Also Alatas [32] presented another interesting application. He has integrated chaos search with HS for improved performance. Seven new chaotic HS algorithms have been developed using different chaotic maps. A similar utilizing of chaotic sequences for artificial bee colony (ABC) [34], BB-BC [35], ICA [1], and CSS [36] have been performed by researchers. Based on the results obtained from literature it is not easy to say which chaotic map performs the best. However, we can say that chaotic maps have a considerable positive impact on the performance of meta-heuristics.

In these studies generally unconstraint problems were considered. On the other hand, most of the real life problems including design optimization problems require several types of variables, objective and constraint functions simultaneously in their formulation. In engineering design as the first attempts to analyze the performance of meta-heuristics in which chaotic maps are used for parameters updating process, Talatahari et al. [37] combined the benefits of chaotic maps and the ICA to determine optimum design of truss structures. These different chaotic maps were investigated by solving two benchmark truss examples involving 25- and 56-bar trusses to recognize the most suitable one for this algorithm. As an example taken from the original paper a 56-bar dome truss structure is shown in Figure 2.

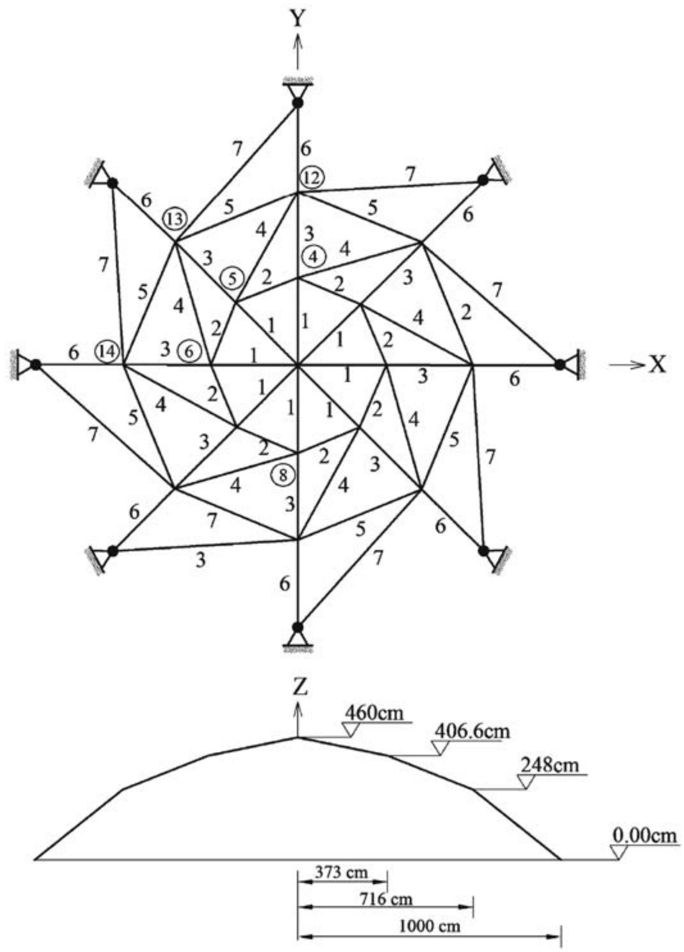

Figure 2. A 56-bar dome spatial truss structure [37]

Members of the dome are into 7 groups. Table 1 shows the statistical results and the optimum weight for the 56-bar dome truss using the ICA algorithms where cm is a chaotic map based on the Sinusoidal map for CICA-1, Logistic map for CICA-2, Zaslavskii map for CICA-3 and Tent map for CICA-4 [37]. The results show that the use of Sinusoidal map (CICA-1) results in a better performance for the chaotic ICA than others. Two other larger examples were also considered by Talatahari et al. [37] to obtain more clear details about the performance of the new algorithm. These were 200- and 244-bar trusses with 29 and 32 design variables, respectively. Almost for all examples, the performance of the new algorithm is far better than the original ICA; especially when the standard deviations of the results are compared. The standard deviation of the new algorithm is much better than the original ICA and this illustrates the high ability of the new algorithm.

Table 1: Optimal design comparison for the 56-bar dome truss

	ICA	CICA-1	CICA-2	CICA-3	CICA-4
Best weight (kg)	546.14	546.13	546.16	546.15	546.15
Average weight (kg)	547.91	546.21	546.31	546.24	546.34
Std Dev (kg)	5.791	0.49	0.62	0.56	0.59

As another attempt in optimization problems related to the engineering design a new improved CSS using chaotic maps was presented for engineering optimization by Talatahari et al. [38]. They defined five different variants of the new methodology by adding the chaos to the enhanced CSS. Then, different chaotic systems were utilized instead of different parameters available in the algorithm. To evaluate the performance of the new algorithm two sets of examples were considered: In the first set four well-known benchmark examples including design of a piston lever, design of a welded beam, design of a four-storey, two-bay frame, and design of a car side impact were selected from literature to compare the variants of the new method. In the second set two mechanical examples consisting of a 4 step-cone pulley design and speed reducer design problems were utilized in order to compare the variants of the new method with other meta-heuristics. As an example taken from the original paper, in design of a 4 step-cone pulley the objective is to design a pulley with minimum weight using 5 design variables, as shown in Figure 3.

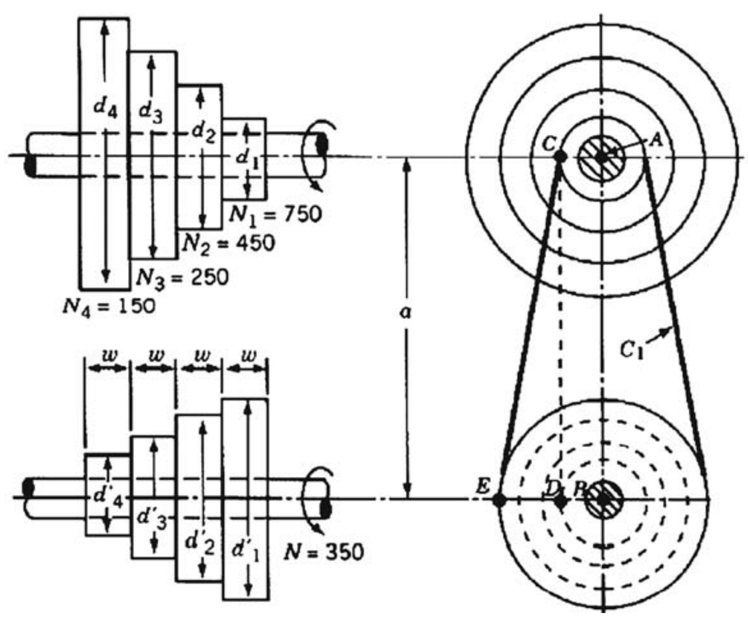

Figure 3. A 4 step-cone pulley [39]

Four design variables are associated with the diameters of each step, and the fifth corresponds to the width of the pulley. In this example, it is assumed that the widths of the cone pulley and belt are identical. There are 11 constraints, out of which 3 are equality constraints and the remaining are inequality constraints. The constraints are imposed to assure the same belt length for all the steps, tension ratios, and power transmitted by the belt. The 4 step pulley is designed to transmit at least 0.75 hp (0.75 · 745.6998W), with an input speed of 350 rpm and output speeds of 750, 450, 250, and 150 rpm. This problem is considered to compare the chaotic CSS (CCSS) method with other meta-heuristic algorithms which was solved by using Teaching–learning-based optimization (TLBO) and ABC, previously [39]. It is observed from Table 2 that CCSS gives better results than the other methods for the best, mean, and standard deviation [39].

Due to the simplicity and potency of these methods, it seems that they can easily be utilized for many engineering problems to find the optimum designs.

Table 2. Statistical results of the 4 step-cone pulley for different meta-heuristics

Method	Best	Mean	Std Dev
TLBO	16.63451	24.0113	0.34
ABC	16.63466	36.0995	0.06
CCSS	16.41235	29.1058	0.11

3.2 Chaotic search strategy in meta-heuristics

The basic idea of chaos optimization algorithm (COA) generally includes two major stages. Firstly, based on the selected chaotic map (cm) define a chaotic number generator for generating sequences of points then map them to a design space. Afterwards, evaluate the objective functions with respect to these points, and choose the point with the minimum objective function as the current optimum. Secondly, the current optimum is assumed to be close to the global optimum after certain iterations, and it is viewed as the center with a little chaotic perturbation, and the global optimum is obtained through fine search. Repeat the above two steps until some specified convergence criterion is satisfied, and then the global optimum is obtained [40]. The pseudo-code of COA is summarized as follows

Step 1 *Initialization*. Initialize the number N of chaotic search, different initial value of n chaos variables cm_i^0, and the lower and upper bound of the decision variables (X_L and X_U). Set the iteration counter as $k = 1$. Determine the initial design variables as

$$x_i^0 = X_{L_i} + cm_i^0 (X_{U_i} - X_{L_i}), \qquad i = 1, 2, ..., n \tag{13}$$

Evaluate the objective function and set $f^* = f(x^0)$.

Step 2 *Variable mapping*. Map chaotic variables cm^k into the variance range of the optimization variables by the following equation

$$x_i^{k+1} = X_{L_i} + cm_i^{k+1}(X_{U_i} - X_{L_i}), \qquad i = 1, 2, ..., n \tag{14}$$

Step 3 *Searching for optimal solution.* Evaluate the objective function.

If $k \leq N$, then

If $f(x^{k+1}) \leq f^*$, then $x^* = x^{k+1}, f^* = f(x^{k+1})$.

Set $k = k+1$, $cm^k = cm^{k+1}$, and go to step 2.

Else if $k > N$ is satisfied then stop.

Due to the pseudo-randomness of chaotic motion, the motion step of chaotic variables between two successive iterations is always big, which resulted in the big jump of the design variables in design space. Thus, even if the above COAs have reached the neighborhood of the optimum, it needs to spend much computational effort to approach the global optimum eventually by searching numerous points.

Hence, the hybrid methods attracted the attention of some researchers, in which chaos is incorporated into the meta-heuristics where the parallel searching ability of meta-heuristics and chaotic searching behavior are reasonably combined. Wu and Cheng [41] integrated GA with COA, which uses chaos sequence to generate original population and add chaotic fine search to genetic operation which can avoid premature convergence. Guo and Wang [42] presented a novel immune evolutionary algorithm (IEA) based on COA to improve the convergence performance of the IEA. Ji and Tang [43] and Liu et al. [4] suggested a hybrid method of SA and PSO combined with chaos search, and examined its efficiency with several nonlinear functions, respectively. Similar approaches were also presented for PSO by Wang and Liu [44], Gao and Liu [45], and He et al [46]. Finally Baykasoglu [47] presented how can the performance of great deluge algorithm (GDA) be enhanced by integrating with COA for solving constrained non-linear engineering design optimization problems. Such hybrid methods can save much CPU time and enhance the computational efficiency of algorithms.

4. DIRECTION OF FUTURE RESEARCH

Chaos embedded meta-heuristic optimization algorithms can be used to optimize single as well as multiobjective optimization problems. It has been applied in various fields like engineering design, operational research, etc. Though in the last decade some work has been done in this area, most of it has been concentrated on application of some conventional or ad-hoc techniques to certain difficult problems. Despite the success of these modern meta-heuristic algorithms, there are some important questions which remain unanswered. It is known that how these algorithms work, and also partly understood why these algorithms work. However, it is difficult to describe mathematically why these algorithms are successful. In fact, these are unresolved open problems. Any mathematical analysis will thus provide important insight into these algorithms. Another question is why does a balanced combination of randomization and a deterministic component lead to a much more efficient algorithm (than a completely deterministic and/or a completely random algorithm)? How to measure or test if a balance is reached?

In addition, a current trend is to use simplified meta-heuristic algorithms to deal with complex optimisation problems. Possibly, there is a need to develop more complex and hybrid meta-heuristic algorithms combined with chaos theory which can truly mimic the exact working mechanism of some natural and biological systems, leading to more powerful next generation, self-regulating, self-evolving, self-organizing and truly intelligent algorithms. For example Chaotic swarming of particles (CSP) is a newly developed type of meta-heuristic algorithms. This algorithm is proposed by Kaveh et al. [48]. The framework of the CSP is illustrated in Figure 4.

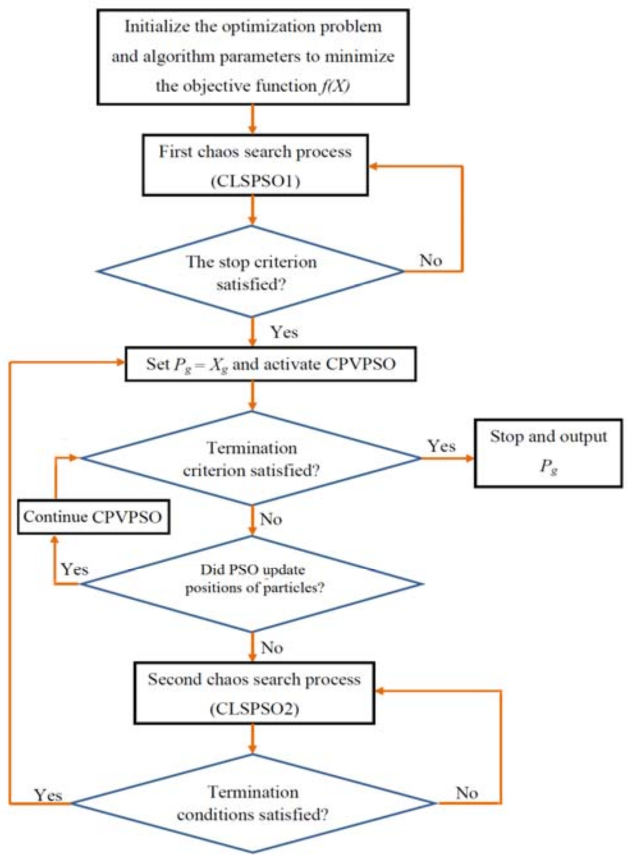

Figure 4. Flowchart of the CSP algorithm [48]

The CSP is inspired from the chaotic and collective behavior of species such as bees, fishes, and birds in which chaos theory is used to control the value of the parameters of PSO and to increase the local search capability of the PSO in order to enhance search behavior and skip local optima. The CSP approach not only performs exploration by using the population-based evolutionary searching ability of PSO, but also performs exploitation by using the chaotic local searching behavior. In the CLSPSO1 (chaotic local search) phase, the initial positions of the particles are determined chaotically in the search space. Also the values of the fitness function for the particles are calculated. The best particle among the entire set of particles is treated as a global best. After reaching a pre-defined number of iterations, the CLSPSO1 is stopped and switched to PSO while CPVPSO applies for updating the value of parameters in the velocity updating equation. In the second phase, the CLSPSO2 (updating process) is activated if PSO stops moving. CLSPSO2 causes the particles to escape from local minima using the logistic map. After a better solution is found by the CLSPSO2 or after a fixed number of iterations, the PSO will continue. The algorithm is terminated when the termination criterion has been met: that is, if there is no significant improvement in the solution [48].

5. CONCLUSIONS

As an important tool in optimization theory, meta-heuristic algorithms explore the search space of the given data in both exploration and exploitation manner and provide a near-optimal solution within a reasonable time. Meta-heuristics have many features that make them as suitable techniques not only as standalone algorithms but also to be combined with other optimization methods. Even the standard meta-heuristics have been successfully implemented in various applications; however, many modification and improvements to these algorithms have been also reported in the literature. Each of them is tightly related to some aspects of these algorithms such as parameters setting or balancing of exploration and exploitation. In this paper, we turn the attention to chaos embedded meta-heuristic algorithms and survey most of the modifications proposed in the literature.

Chaos is a bounded unstable dynamic behavior that exhibits sensitive dependence on initial conditions and includes infinite unstable periodic motions in nonlinear systems. Recently, the idea of using the benefits of chaotic systems has been noticed in several fields. One of these fields is optimization theory. Experimental studies show the performance of combining chaos and meta-heuristics. Here chaos embedded meta-heuristics are classified into two general categories. First category contains the algorithms in which chaos is used instead of random number generators. On the other hand in the second category chaotic search that uses chaotic map is incorporated into meta-heuristics to enhance searching behavior of these algorithms and to skip local optima.

Finally a new combination of swarm intelligence and chaos theory is introduced in which the tendency to form swarms appearing in many different organisms and chaos theory has been the source of inspiration, and the algorithm is called Chaotic Swarming of Particles (CSP). This method is a kind of multi-phase optimization technique which employs chaos theory in two phases, in the first phase it controls the parameter values of the PSO and the second phase is utilized for local search using COA.

Though we have already seen some examples of successful combinations of chaos and meta-heuristic algorithms, there still remain many open problems due to the existence of many inherent uncertain factors.

REFERENCES

1. Talatahari S, Farahmand Azar B, Sheikholeslami R, Gandomi AH. Imperialist competitive algorithm combined with chaos for global optimization. *Commun Nonlinear Sci Numer Simul*, 2012; **17**:1312-9.
2. Tavazoei MS, Haeri M. An optimization algorithm based on chaotic behavior and fractal nature. *J Comput Appl Math*, 2007; **206**:1070-81.
3. Bucolo M, Caponetto R, Fortuna L, Frasca M, Rizzo A. Does chaos work better than noise?. *IEEE Circuits Syst Mag*, 2002; **2**(3): 4-19.
4. Liu B, Wang L, Jin Y H, Tang F, Huang DX. Improved particle swarm optimization combined with chaos. *Chaos Soliton Fract*, 2005; **25**(5): 1261-71.
5. Liu S, Hou Z. Weighted gradient direction based chaos optimization algorithm for nonlinear programming problem. *Proceedings of the 4th World Congress on Intelligent Control and Automation*, 2002, 1779-83,
6. Li TY, Yorke JA. Period three implies chaos. *Amer Math Monthly*, 1975; **82**: 985-992.
7. Tatsumi K, Obita Y, Tanino T. Chaos generator exploiting a gradient model with sinusoidal perturbations for global optimization. *Chaos Soliton Fract*, 2009; **42**:1705-1723.
8. Hilborn RC. Chaos and Nonlinear Dynamics. Oxford University Press, USA, 2000.
9. Heidari-Bateni G, McGillem CD. A chaotic direct-sequence spread spectrum communication system, *IEEE Trans Commun*, 1994; **42**(2-4): 1524-7.
10. May R. Mathematical models with very complicated dynamics. *Nature*, 1976; **261**:459-67.
11. Peitgen H, Jurgens H, Saupe D. *Chaos and Fractals*. Springer-Verlag, Berlin, Germany, 1992.
12. Zheng WM. Kneading plane of the circle map. *Chaos Soliton Fract*, 1994; **4**(7):1221-33.
13. Dressler U, Farmer JD. Generalized Lyapunov exponents corresponding to higher derivatives. *Physica D*, 1992; **59**: 365-77.
14. Zaslavskii GM. The simplest case of a strange attractor. *Physics Letters A*, 1978; **69**(3): 145-7.
15. Glover F, Kochenberger GA. *Handbook of Metaheuristic*. Kluwer Academic Publishers, 2003.
16. Talbi EG. *Metaheuristics: from design to implementation*. John Wiley & Sons, New Jersey, 2009.
17. Dorigo M. *Metaheuristics network website, http://www.metaheuristics.net/*. Visited in January 2010.
18. Holland JH. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

19. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science, New Series*, 1983; **220**(4598):671-80.
20. Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybernet B*, 1996; **26**(1) 29–41.
21. Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. *In Proceedings of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995.
22. Geem ZW, Kim JH, Loganathan GV. (2001) A new heuristic optimization algorithm; harmony search. *Simul*, 2001; **76**:60–8.
23. Erol OK, Eksin, I. A new optimization method: big bang-big crunch. *Adv Eng Software*, 2006; **37**(2):106-11.
24. Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *In Proceedings of the IEE congress on evolutionary computation*, 2007, 4661-7,
25. Yang, XS. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
26. Yang XS, Deb S. Cuckoo search via L´evy flights. *In Proceedings of the World Congress on Nature & Biologically Inspired Computing*, 210-214, 2009.
27. Kaveh A, Talatahari S. Novel heuristic optimization method: charged system search. *Acta Mech*, 2010; **213**(3-4):267–89.
28. Kaveh A, Motie Share MA, Moslehi M. Magnetic charged system search: a new meta-heuristic algorithm for optimization. *Acta Mech*, 2013; **224**(1): 85-107.
29. Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization. *Copmut Struct*, 2012; **112-113**:283–94.
30. Schuster HG. *Deterministic Chaos: An Introduction*. 2nd Revised Ed, Physick-Verlag GmnH, D-6940 Weinheim, Federal Republic of Germany, 1988.
31. Coelho L, Mariani V. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Syst Appl*, 2008; **34**:1905-13.
32. Alatas B. Chaotic harmony search algorithm. *Appl Math Comput*, 2010; **29**(4):2687-99.
33. Alatas B, Akin E, Ozer AB. Chaos embedded particle swarm optimization algorithms. *Chaos Soliton Fract*, 2009; **40**:1715-34.
34. Alatas B. Chaotic bee colony algorithms for global numerical optimization. *Expert Syst Appl*, 2010; **37**:5682-7.
35. Alatas B. Uniform big bang-chaotic big crunch optimization. *Commun Nonlinear Sci Numer Simul*, 2011; **16**(9):3696-703.
36. Talatataharis S, Kaveh A, Sheikholeslami R. An efficient charged system search using chaos for global optimization problems. *Int J Optim Civil Eng*, 2011; **1**(2): 305-25.
37. Talatahari S, Kaveh A, Sheikholeslami R. Chaotic imperialist competitive algorithm for optimum design of truss structures. *Struct Multidiscip Optim*, 2012; **46**:355-67.

38. Talatahari S, Kaveh A, Sheikholeslami R. Engineering design optimization using chaotic enhanced charged system search algorithms. *Acta Mech*, 2012; **223**:2269-85.
39. Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des*, 2011; **43**:303-15.
40. Yang D, Li G, Cheng G. On the efficiency of chaos optimization algorithms for global optimization. *Chaos Soliton Fract*, 2007; **34**:1366-75.
41. Wu TB, Cheng Y, Zhou TY, Yue Z. Optimization control of PID based on chaos genetic algorithm. *Comput Simul*, 2009; **26**:202-4.
42. Guo ZL, Wang SA. The comparative study of performance of three types of chaos immune optimization combination algorithms. *J Syst Simul*, 2005; **17**:307-9.
43. Ji MJ, Tang HW. Application of chaos in simulated annealing. *Chaos Soliton Fract*, 2004; **21**:933-41.
44. Wang Y, Liu JH. Chaotic particle swarm optimization for assembly sequence planning. *Robot and Cim-Int Manuf*, 2010; **26**:212-22.
45. Gao L, Liu X. A resilient particle swarm optimization algorithm based on chaos and applying it to optimize the fermentation process. *Int J Inform Syst Sci*, 2009; **5**:380-91.
46. He Y, Zhou J, Li C, Yang J, Li Q. A precise chaotic particle swarm optimization algorithm based on improved tent map. *In Proceedings of the 4th International Conference on Natural Computation*, 2008, 569-73.
47. Baykasoglu A. Design optimization with chaos embedded great deluge algorithm. *Appl Soft Comput*, 2012; **12**(3):1055-67.
48. Kaveh A, Sheikholeslami R, Talatahari S, Keshvari-Ilkhichi M. Chaotic swarming of particles: a new method for size optimization of truss structures. *Adv Eng Softw*, 2014; **67**:136-47.