Advances in Engineering Software 67 (2014) 136–147

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Chaotic swarming of particles: A new method for size optimization of truss structures

A. Kaveh [a,*], R. Sheikholeslami [b], S. Talatahari [c], M. Keshvari-Ilkhichi [b]

[a] Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran 16, Iran
[b] Department of Civil and Environmental Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
[c] Department of Civil Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article history:
Received 29 August 2012
Received in revised form 13 September 2013
Accepted 22 September 2013
Available online 13 October 2013

Keywords:
Meta-heuristic algorithm
Swarm intelligence
Particle swarm optimizer
Chaos theory
Truss structures
Size optimization

ABSTRACT

A new combination of swarm intelligence and chaos theory is presented for optimal design of truss structures. Here 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 particle swarm optimization (CPVPSO) and the second phase is utilized for local search (CLSPSO). Some truss structures are optimized using the CSP algorithm, and the results are compared to those of the other meta-heuristic algorithms showing the effectiveness of the new method.

1. Introduction

Truss structures form a broad category of man-made structures, including bridges, towers, cranes, roof support trusses, building exoskeletons, and temporary construction frameworks. Trusses derive their utility and distinctive look from their simple construction: rod elements (bars) which exert only axial forces, connected concentrically with welded, pinned or bolted joints. These utilitarian structures are ubiquitous in the industrialized world and can be extremely complex and thus difficult to model. For example, the Eiffel Tower, perhaps the most famous truss structure in the world, contains over 15,000 girders connected at over 30,000 points [1] and even simpler structures, such as railroad bridges, routinely contain hundreds of members of varying lengths [2]. Therefore due to the increasing tendency on the prices of materials, the utilization of the modern design optimization tools becomes a necessity. In engineering the main objective of optimization is to comply with basic standards but also to achieve good economic results. Design variables involved in optimum design of truss structures can be considered as sizing (finding the optimal sections for elements), lay out (the optimum location of the joints in the structure), and topology (finding the number of members of the structure and the way in which these members are connected to each other) variables.

On the other hand, optimal design of (minimizing the weight) a structure while at the same time satisfying various requirements on structural response, cost, aesthetics, and manufacturing is a complicated task. Experienced engineers may be able to come up with solutions that fulfill some of the requirements, but they will seldom be able to come up with the optimal structure [3]. In order to both optimize the structure and meet the given requirements, researchers use various techniques. One of them is design optimization with meta-heuristic algorithms. Meta-heuristics are developed to tackle complex optimization problems where other optimization methods have failed to be either effective or efficient. Meta-heuristic algorithms which are created by the simulation of the natural processes try to find the optimal solution in a stochastic manner and avoid local optimum solutions. These algorithms impose fewer mathematical requirements and they do not require very well defined mathematical models.

Among these phenomenon-mimicking methods, algorithms inspired from the collective behavior of species such as ants, bees, wasps, termite, fishes, and birds are referred as swarm intelligence algorithms [4]. Here, we introduce a new combination of chaos theory and swarm intelligence which is called the chaotic swarming of particles (CSP) for optimum design of truss structures which can be considered as a suitable field to investigate the efficiency of this algorithm. One of these swarm intelligence algorithms is particle swarm optimization (PSO) which is a population-based meta-heuristic discovered through simulation of social models of

* Corresponding author.
E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

0965-9978/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2013.09.006

bird flocking, fish schooling, and swarming able to find optimal solution(s) to the non-linear numeric problems. PSO was first introduced in 1995 by Eberhart and Kennedy [5], and has attracted much attention in various fields. However, PSO can easy be trapped in local optimal point when dealing with some complex and multi-modal functions.

Recently, chaos and PSO have been combined in different studies for different purposes. Some of the works have intended to show the chaotic behaviors in the PSO process. In some of the works, chaos has been used to overcome the limitations of PSO [6]. Hence previous research can be categorized into two types. In the first type, chaos is embedded into the PSO velocity updating equation, *i.e.*, a chaotic map is used to control the value of parameters in the velocity updating equation [7–12]. In the second type, chaotic search is inserted in the PSO formulation [13–16]. In this article, we combined the two types of improvements to control the value of the parameters and to increase the local PSO search capability. This enhances search behavior and allows to avoid local optima.

The present article consists of five sections. After the introduction in Section 1, the proposed method is described in Section 2. The formulation of truss sizing problems is presented in Section 3. Benchmark examples are studied in Section 4 and the results of the proposed method are compared with literature. The paper is concluded in Section 5.

2. Chaotic swarming of particles

2.1. Standard particle swarm optimizer

PSO involves a number of particles, which are initialized randomly in the space of the design variables. These particles fly through the search space and their positions are updated based on the best positions of individual particles and the best position among all particles in the search space which in truss sizing problems corresponds to a particle with the smallest weight [5]. In PSO, a swarm consists of N particles moving around in a D-dimensional search space. The position of the jth particle at the kth iteration is used to evaluate the quality of the particle and represents candidate solution(s) for the search or optimization problems. The update moves a particle by adding a change velocity V_j^{k+1} to the current position X_j^k as follows:

$$\begin{aligned} V_j^{k+1} &= wV_j^k + c_1 \times r_{1j}^k \otimes (P_j^k - X_j^k) + c_2 \times r_{2j}^k \otimes (P_g^k - X_j^k) \\ X_j^{k+1} &= X_j^k + V_j^k \end{aligned} \tag{1}$$

where w is an inertia weight to control the influence of the previous velocity; r_{1j}^k and r_{2j}^k are random numbers uniformly distributed in the range of (0,1); c_1 and c_2 are two acceleration constants namely called cognitive and social parameter, respectively; P_j^k is the best position of the jth particle up to iteration k; P_g^k is the best position among all particles in the swarm up to iteration k. In order to increase PSO's exploration ability, the inertia weight is now modified during the optimization process with the following equation:

$$w^{k+1} = w^k \times D_r \times rand \tag{2}$$

where D_r is the damping ratio which is a constant number in the interval (0,1); and *rand* is a uniformly distributed random number in the range of (0,1).

In standard PSO algorithm, the information of local best and global best were shared by next generation particles. In this paper we use an improved PSO, which uses the dynamic inertia weight that decreases according to iterative generation increasing (Eq. (2)). A larger inertia weight facilitates global exploration and a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area. The inertia weight, w, controls the momentum of the particle by weighing the contribution of the previous velocity: basically it controls how much memory of the previous flight direction should influence the new velocity [17]. Fig. 1 shows the flowchart of standard PSO.

2.2. Chaotic update of PSO internal parameters (CPVPSO)

Chaos and random signals share the property of long term unpredictable irregular behavior. Many random generators in programming softwares as well as the chaotic maps are deterministic. However, chaos can help order to arise from disorder. Similarly, many 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 dissymmetry [18]. On the other hand self-organization and evolution are two key factors of many stochastic optimization techniques such as PSO. Due to these common properties between chaos and optimization algorithms, simultaneous use of these concepts seems to improve the performance of the optimizer. Utilizing chaotic sequences for particle swarm optimization [7], harmony search algorithm [19], artificial bee colony [20], Big Bang–Big Crunch [21], imperialist competitive algorithm [18,22], and charged system search [23] are some familiar examples of this combination. Seemingly the benefit of such combination is a generic for other stochastic optimization and experimental studies have confirmed this; however, this has not yet mathematically been proven.

In this phase, when a random number is needed by PSO algorithm, it can be generated by iterating one step of the chosen chaotic map (cm) being started from a random initial condition of the first iteration of PSO. One-dimensional non-invertible maps are the simplest systems with capability of generating chaotic motion [24]. One of the well-known chaotic maps is the Logistic map. Logistic map is a polynomial map [25]. It is often cited as an example of how complex behavior can arise from a very simple non-linear dynamics equation. This map is defined by,

$$cm^{k+1} = \alpha \times cm^k(1 - cm^k) \tag{3}$$

where cm^k is the kth chaotic number, with k denoting the iteration number. It is trivial to show that if $0 < \alpha \leqslant 4$ then the interval (0,1) is mapped into itself, *i.e.* if $cm^0 \in (0,1)$ then $cm^k \in (0,1)$. It is proven that when $\alpha = 4$, Eq. (3) is totally in chaos state. The mathematical explanation is that all values between 0 and 1 except the fixed points (0.25, 0.5, 0.75) are produced randomly by iteration. Utilizing the chaos characteristic, which is sensitive to the initial value and setting n different initial values between 0 and 1 (except the fixed point) to cm^k in Eq. (3), one can get n chaos variables of different orbits.

In order to control values of PSO parameters by using chaotic maps, the approach described in [7] is followed in this research; r_{1j}^k, r_{2j}^k, and *rand* are generated from the iterations of chaotic map instead of using classical random number generator.

$$\begin{aligned} V_j^{k+1} &= w^k \times V_j^k + c_1 \times cm^k \otimes (P_j^k - X_j^k) + c_2 \times cm^k \otimes (P_g^k - X_j^k) \\ w^{k+1} &= w^k \times D_r \times cm^k \end{aligned} \tag{4}$$

2.3. Chaotic local search algorithm (CLSPSO)

In this phase, chaotic search is introduced in the PSO formulation. This is a kind of multi-phase optimization technique [14] because chaotic optimization and PSO coexist and are switched to each other according to certain conditions. Here, chaotic local