Engineering Structures 91 (2015) 16–25

Contents lists available at ScienceDirect

Engineering Structures

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

Optimum design of frame structures using the Eagle Strategy with Differential Evolution

Siamak Talatahari [a,*], Amir Hossein Gandomi [b], Xin-She Yang [c], Suash Deb [d]

[a] Department of Civil Engineering, University of Tabriz, Tabriz, Iran
[b] Department of Civil Engineering, University of Akron, Akron, OH, USA
[c] Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, UK
[d] Department of Computer Science & Engineering, C.V. Raman College of Engineering, Bidyanagar, Mahura, Janla, Bhubaneswar 752054, India

ARTICLE INFO

Article history:
Received 5 April 2013
Revised 19 February 2015
Accepted 21 February 2015

Keywords:
Frame structures
Optimum design
Eagle strategy
Differential evolution

ABSTRACT

Modern metaheuristic algorithms are in general suited for global optimization. This paper combines the recently developed eagle strategy algorithm with differential evolution. The new algorithm, denoted as the ES–DE, is implemented by interfacing SAP2000 structural analysis code and MATLAB mathematical software. The performance of the ES–DE is evaluated by solving four benchmark problems where the objective is to minimize the weight of steel frames. The optimized designs obtained by the proposed algorithm are better than those found by the standard differential evolution algorithm and also very competitive with literature. The overall convergence behavior is significantly enhanced by the hybrid optimization strategy.

1. Introduction

Metaheuristic algorithms represent a popular approach to the solution of complicated nonlinear optimization problems [1–3]. However, while most algorithms show a relatively high efficiency in approaching the best region of design space, there is no optimization formulation inherently able to always find the global optimum. Current research on metaheuristic algorithms hence focuses on increasing/improving their capability to carry out global search and find the global optimum (or some designs very close to it).

The efficiency of metaheuristic algorithms derives from the fact that they are designed to imitate the best features in nature inspiring to different sources (for example, bio-inspired krill herd [4], physics-inspired charged system search [5], music-inspired harmony search [6] etc.). Biological systems are the main source for proposing new nature-inspired metaheuristic approaches because the selection of the fittest in biological systems has evolved by natural selection over millions of years. There exist some algorithms for stochastic optimization such as, for example, the Eagle Strategy (ES) developed by Yang and Deb [7].

Frame design is one of the popular structural optimization benchmarks with a diverse range of design flexibility [8], and such design problems can be a mixed type. For frame structures applications, some modifications to the optimization algorithms are often required. Therefore, researchers have attempted to solve frame structures as a real-world, discrete-variable and nonlinear optimization problem using different methods [9]. The objective in these problems is usually minimizing the frame weight under complex nonlinear constraints [10–13]. The design variables of interest in frame structures are cross sections of beams and columns, which have to be chosen from standardized cross sections [14], and thus leading to discrete and mixed type of optimization.

This paper describes a hybrid metaheuristic algorithm combining the ES and Differential Evolution (DE). The new algorithm, referred to as the ES–DE in the rest of the paper, is a two-stage search method. The ES–DE is tested in four classical weight minimization problems of frame structures. The optimization algorithm is implemented by interfacing MATLAB with the SAP2000 structural analysis code. Optimization results are compared with the standard DE and other methods documented in literature. It appears that the proposed approach is very effective in frame design optimization.

2. Frame optimization problems

Optimal design of frame structures can be formulated as

$$\begin{aligned} &\text{Find} && \mathbf{X} = [x_1, x_2, \ldots, x_{ng}] \\ &\text{to minimize} && Mer(\mathbf{X}) = f(\mathbf{X}) \times f_{penalty}(\mathbf{X}) \end{aligned} \qquad (1)$$

* Corresponding author at: Department of Civil Engineering, University of Tabriz, Tabriz, Iran.
E-mail addresses: siamak.talat@gmail.com, talatahari@tabrizu.ac.ir (S. Talatahari).

http://dx.doi.org/10.1016/j.engstruct.2015.02.026
0141-0296/© 2015 Elsevier Ltd. All rights reserved.

where **X** is the design vector of the cross sectional areas of W sections; *ng* is the number of design variables or the number of member groups; *Mer*(**X**) is the merit function; *f*(**X**) is the cost function which is usually taken as the weight or volume of the structure; $f_{penalty}(\mathbf{X})$ is the penalty function which results from the violations of the optimization constraints on structural response.

The cost function, *f*(**X**) in the form of the total weight of the frame structure can be expressed as:

$$f(\mathbf{X}) = \sum_{i=1}^{nm} \gamma_i \cdot x_i \cdot L_i \tag{2}$$

where γ_i is the material density of *i*-th member and L_i is the length of *i*-th member; *nm* is the number of members making up the frame.

The penalty function, $f_{penalty}(\mathbf{X})$, can be defined as [15]:

$$f_{penalty}(\mathbf{X}) = (1 + \varepsilon_1 \cdot \upsilon)^{\varepsilon_2}, \quad \upsilon = \sum_{i=1}^{n} \max[0, \upsilon_i] \tag{3}$$

where *n* represents the total number of constraints for each individual design. The constants ε_1 and ε_2 are selected, depending on the exploration and the exploitation rate of the search space. If stress/displacement constraints, υ_i, turn positive, the corresponding values taken by constraint functions are added as penalty terms. These constraints contain:

Element stresses

$$\upsilon_i^{\sigma} = 1 - \left|\frac{\sigma_i}{\sigma_i^a}\right| \le 0, \quad i = 1, 2, \ldots, nm \tag{4}$$

Maximum lateral displacement

$$\upsilon^{\Delta} = R - \frac{\Delta_T}{H} \le 0, \tag{5}$$

Inter-story displacements

$$\upsilon_j^d = R_I - \frac{d_j}{h_j} \le 0, \quad j = 1, 2, \ldots, ns \tag{6}$$

where σ_i is the stress in *i*-th member; σ_i^a is the allowable stress in *i*-th member; Δ_T is the maximum lateral displacement; *H* is the height of the frame structure; *R* is the maximum drift index; d_j is the inter-story drift; h_j is the story height of the *j*th floor; *ns* is the total number of stories; R_I is the inter-story drift index permitted by the standard design code in engineering practice.

Following AISC 2001 [16], the allowable inter-story drift index is set as 1/300. The LRFD interaction formula constraints (Equation H1-1a,b of AISC 2001) are stated as

$$\upsilon_i^I = 1 - \frac{P_u}{2\phi_c P_n} - \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) \le 0 \text{ For } \frac{P_u}{\phi_c P_n} < 0.2 \tag{7}$$

$$\upsilon_i^I = 1 - \frac{P_u}{\phi_c P_n} - \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right) \le 0 \text{ For } \frac{P_u}{\phi_c P_n} \ge 0.2 \tag{8}$$

where P_u is the required strength (tension or compression); P_n is the nominal axial strength (tension or compression); ϕ_c is the resistance factor ($\phi_c = 0.9$ for tension, $\phi_c = 0.85$ for compression); M_{ux} and M_{uy} are the required flexural strengths in the *x* and *y* directions; respectively; M_{nx} and M_{ny} are the nominal flexural strengths in the *x* and *y* directions (for two-dimensional structures, $M_{ny} = 0$); and ϕ_b is the flexural resistance reduction factor ($\phi_b = 0.90$).

To compute compression and Euler stresses, the effective length factors *K* are required. For beam and bracing members, *K* is taken equal to unity. For column members, *K* values are calculated by SAP2000. However, the following approximate effective length formulas can also be used based on Dumonteil [17], which are accurate within −1.0% and +2.0% [18]:

For unbraced members:

$$K = \sqrt{\frac{1.6G_AG_B + 4(G_A + G_B) + 7.5}{G_A + G_B + 7.5}} \tag{9}$$

For braced members:

$$K = \frac{3G_AG_B + 1.4(G_A + G_B) + 0.64}{3G_AG_B + 2(G_A + G_B) + 1.28} \tag{10}$$

3. Eagle strategy

Eagle strategy developed by Yang and Deb [7] is a two-stage method for optimization. In nature, "eagles hunting for prey" is a search process, and this process typically has two stages: a roaming stage and a chasing stage. During the roaming stage, an eagle searches a large but sparse area. Once a prey is seen, it switches to an intensive chasing stage so as to get to the prey as quickly as possible. This two-stage strategy can be formulated as a two-stage search algorithm, called eagle strategy, in which the first stage explores a large search area, while the latter stages focus on the quicker search steps. Eagle strategy uses a combination of a crude global search and an intensive local search with a proper balance of different algorithms to suit different purposes. In essence, the strategy first explores the search space globally using a Lévy-flight random walk. If a promising solution or a set of efficient designs is found, then an intensive local search is carried out by using a more efficient local optimizer such as hill-climbing and downhill simplex method. Then, this two-stage process restarts again with a new global exploration, followed by a local search in a new or more promising region.

The first stage is to generate solutions in the search domain using Lévy flights. That is:

$$\boldsymbol{X}_i^t = \boldsymbol{X}^* + \gamma L(s), \quad L(s) \sim \frac{\beta\Gamma(\beta)\sin(\pi\beta/2)}{\pi} \cdot \frac{1}{s^{1+\beta}}, \quad s \gg 0, \tag{11}$$

where $\sim$ means that $L(s)$ is drawn the above Lévy distribution when step sizes are big enough with the exponent $\beta = 1.5$. Here $\gamma = 0.01$ is often used as a scaling parameter; $\boldsymbol{X}^*$ is the best solution found so far.

The advantage of such a combination is to use a balanced trade-off between a global search (which is often slow) and a fast local search. Balance of search mechanisms and parameter tuning are usually important for metaheuristic algorithms. Another advantage of the two-stage optimization strategy is that one can choose any algorithm at each stage or even within the current iteration. This makes it easier to combine strength points of different algorithms to improve the optimization search. Fig. 1 presents a simplified pseudo-code of the ES process. It can be seen that the global and local searches are controlled by different loops. In particular, the global search is carried out in the outer loop, while the intensive local search is performed in the inner loop. In summary, the local search is based on the best solution found by the global search and analyzes the promising regions of the design space trying to converge to the global optimum as quickly as possible.

It should be noted that the ES is more than an optimization algorithm. In fact, one can use the most suited algorithm for each stage/iteration of the optimization process. The global search algorithm should include enough randomness to explore design space diversely and effectively. The global search may be slow in the early stages of the optimization process but should become faster as we approach to the optimum design. The local search stage should utilize a highly exploitative algorithm in order to find the best designs localized in a restricted region of the design space within a small number of function evaluations.