Engineering Structures 62–63 (2014) 87–97

Contents lists available at ScienceDirect

Engineering Structures

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

Design of space trusses using modified teaching–learning based optimization

C.V. Camp *, M. Farshchin

Department of Civil Engineering, The University of Memphis, Memphis, TN 38152, USA

ARTICLE INFO

Article history:
Received 7 February 2013
Revised 13 January 2014
Accepted 16 January 2014
Available online 11 February 2014

Keywords:
Optimization
Structural engineering
Discrete optimization
Continuous optimization
Spaces trusses

ABSTRACT

A modified teaching–learning-based optimization (TLBO) algorithm is applied to fixed geometry space trusses with discrete and continuous design variables. Designs generated by the modified TLBO algorithm are compared with other popular evolutionary optimization methods. In all cases, the objective function is the total weight of the structure subjected to strength and displacement limitations. Designs are evaluated for fitness based on their penalized structural weight, which represents the actual truss weight and the degree to which the design constraints are violated. TLBO is conceptually modeled on the two types of pedagogy within a classroom: class-level learning from a teacher and individual learning between students. TLBO uses a relatively simple algorithm with no intrinsic parameters controlling its performance and can easily handle a mixture of both continuous and discrete design variables. Without introducing any additional algorithmic parameters, the modified TLBO algorithm uses a fitness-based weighted mean in the teaching phase and a refined student updating process. The computational performance of TLBO designs for several benchmark space truss structures is presented and compared with classical and evolutionary optimization methods. Optimization results indicate that the modified TLBO algorithm can generate improved designs when compared to other population-based techniques and in some cases improve the overall computational efficiency.

1. Introduction

Applications of population-based evolutionary optimization have become very common in science and engineering. Many of these methods seek to model both learned and innate natural behaviors that have been shown to be very efficient at transferring information among individuals within a group or population. Some of the most popular methods are: genetic algorithms (GAs), that first introduced by Holland [20] and Goldberg [17], which model the process of natural evolution; ant colony optimization (ACO), developed by Dorigo [10], which models some of the foraging behaviors of an ant colony; and particle swarm optimization algorithm (PSO), proposed by Kennedy and Eberhart [22], which simulates the social behavior and interactions that occur in a flock of birds or a school of fish. Typically, these metaheuristic methods systematically reduce the search space through iterative processes and focus the population, based on some measure of performance, towards feasible, high-quality solutions.

Another type of heuristic approach utilizes averaged values from a population of potential solutions or collection of information to guide the optimization. In these cases, the apparent accuracy of the mean, as first describe by Galton [15] and later by Helmer-Hirschberg [19] and popularized by Surowiecki [33], is utilized to provide an efficient global search algorithm for optimization problems. For example, Erol and Eksin [12] introduced big bang–big crunch (BB–BC) optimization that uses population averaging in an abstract model of the evolution of the universe.

Recently Rao et al. [29] introduced an innovative approach called teaching–tearning-based optimization (TLBO) which uses simple models for teaching and learning within a classroom as the basis for an evolutionary optimization algorithm. TLBO considers the teacher's effect on students as well as the interaction between students in an iterative process to increase the performance level of the students and overall performance of the class. A TLBO algorithm has two main parts: a Teacher Phase, where the average performance or knowledge of the class is moved towards that of a teacher; and a Learner Phase, where students share information and cooperatively interact with each other.

In structural optimization, according to the complexity of the search space and design constraints, applications of metaheuristic methods are often preferred to gradient based methods. Many

* Corresponding author. Address: Department of Civil Engineering, Engineering Science, Room 106B, The University of Memphis, Memphis, TN 38152, USA. Tel.: +1 901 678 3169; fax: +1 901 678 3026.

E-mail address: cvcamp@memphis.edu (C.V. Camp).

http://dx.doi.org/10.1016/j.engstruct.2014.01.020
0141-0296/© 2014 Elsevier Ltd. All rights reserved.

Nomenclature

k	student number	σ	stress
i	iteration number	δ	deflection
j	design variable's number	ϕ_σ	stress penalty
T	teacher	ϕ_δ	deflection penalty
M	mean	ψ	total penalty
X	design vector	$TLBO$	teaching–learning-based optimization
N	population size	$TLBO_{New}$	teaching–learning-based optimization algorithm when using weighted average approach
F	fitness function	P_x	nodal force in x direction
Δ	amount of update	P_y	nodal force in y direction
r	random value	P_z	nodal force in z direction
s	random value	P_{best}	in discrete design problems, this number represents the percent of population that converged to the optimum design
e	element number	W_{avg}	average weight
L	lower bound	W_{Stdv}	standard deviation weight
U	upper bound	N_{analyses}	number of analysis
N_m	total number of members		
c	connection (node) number		
N_c	total number of connections		

researchers have applied evolutionary algorithms to the optimization of truss structures: Rajeev and Krishnamoorthy [28] and Cao [7] used GAs, Fourie and Groenwold [13], Schutte and Groenwold [30] used PSO, Camp and Bichon [5] used ACO, Lee and Geem [24] used harmony search (HS), Li et al. [25] used heuristic particle swarm optimization (HPSO), Camp [4] used BB–BC, Kaveh and Talatahari [21] used hybrid BB–BC algorithms, Sonmez [32] used artificial bee colony algorithm with an adaptive penalty function approach (ABC-AP), and Degertekin [9] applied self-adaptive harmony search (SAHS). Toğan [34] employed TLBO for the design of planar steel frames with discrete design variables.

While this study considers the design of truss structures, the objective is to demonstrate the advantages of TLBO over other metaheuristic methods. Primarily, because of the intrinsic simplicity of the classroom learning concept and the resulting algorithm, TLBO does not rely on the value of any control parameters to direct its search. For example in GAs, depending on optimization problem, a variety of parameters like crossover rate, mutation rate, etc. must be adjusted to obtain good results. In keeping with the original framework of the algorithm as presented by Rao et al. [29], a modified TLBO is proposed that does not introduce any additional parameters and provides comparable results to other heuristic methods for both discrete and continuous variable truss design problems.

2. Truss design

Truss optimization is typically focused on finding a design that minimizes the structural weight while satisfying stress and displacement constraints. When the truss geometry is fixed, the objective of the optimization is to select a cross-sectional area for each member so that the weight is minimized while stress and deflection constraints are met. In this case, a truss optimization problem can be expressed as:

$$\begin{aligned} &\text{Minimize } w = \sum_{e=1}^{N_m} \gamma_e L_e A_e \\ &\text{Subject to}: \ \sigma^L \le \sigma_e \le \sigma^U \\ &\qquad\qquad\quad \delta^L \leqslant \delta_c \leqslant \delta^U \\ &\qquad\qquad\quad A^L \leqslant A_e \leqslant A^U \end{aligned} \tag{1}$$

where w is the weight of the truss composed of N_m members and for each member e: γ_e is the material unit weight; L_e is the length; and A_e is the cross-sectional area. The truss design must satisfy limits on member stresses σ_e and deflections δ_c at each connection c. Limits on cross-sectional area, stresses, and deflections are given by values defined at lower L and upper U boundaries [5].

In this formulation, the stress σ_e in each member of the truss is compared with the maximum allowable stresses $\sigma^{L,U}$ and the deflections δ_c of each connection c of the truss are compared with the maximum allowable deflections $\delta^{L,U}$ as described in Eq. (1).

A penalty function is used to account for infeasible truss designs. In this formulation, the value of the objective function (the structural weight) is multiplied by a cumulative penalty that is proportional to the amount of any stress and deflection constraint violations [4]. The stress penalty ϕ_σ^e for each member e, is defined as:

$$\text{If } \sigma^L \le \sigma_e \le \sigma^U, \quad \text{then} \quad \phi_\sigma^e = 0 \tag{2}$$

$$\text{If } \sigma_e < \sigma^L \text{ or } \sigma_e > \sigma^U, \quad \text{then} \quad \phi_\sigma^e = \left| \frac{\sigma_e - \sigma^{L,U}}{\sigma^{L,U}} \right| \tag{3}$$

The total stress penalty ϕ_σ^k for a truss design, k, is:

$$\phi_\sigma^k = \sum_{e=1}^{N_m} \phi_\sigma^e \tag{4}$$

Penalties for excessive deflection in the x, y, and z, directions $\phi_{\delta x}^c$, $\phi_{\delta y}^c$, and $\phi_{\delta z}^c$ are computed at each connection c. The general form of the deflection penalty is:

$$\text{If } \delta^L \le \delta_{c(x,y,z)} \le \delta^U, \text{ then } \phi_{\delta(x,y,z)}^c = 0 \tag{5}$$

$$\text{If } \delta_{c(x,y,z)} < \delta^L \text{ or } \delta_{c(x,y,z)} > \delta^U, \text{ then } \phi_{\delta(x,y,z)}^c = \left| \frac{\delta_{c(x,y,z)} - \delta^{L,U}}{\delta^{L,U}} \right| \tag{6}$$

The total deflection penalty ϕ_δ^c for a truss design k with N_c, connections is then determined by:

$$\phi_\delta^k = \sum_{c=1}^{N_c} [\phi_{\delta x}^c + \phi_{\delta y}^c + \phi_{\delta z}^c] \tag{7}$$

The total penalty ψ^k for truss design k is then equal to the summation of the stress and deflection penalties defined as:

$$\psi^k = (1 + \phi_\sigma^k + \phi_\delta^k)^\varepsilon \tag{8}$$

where ε is a positive penalty exponent. The value of penalized fitness function F^k is a product of the weight of truss design k and its total penalty:

$$F^k = \psi^k w^k \tag{9}$$