Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Engineering Structures 28 (2006) 1019–1027
ENGINEERING STRUCTURES
www.elsevier.com/locate/engstruct

Optimization of 3d trusses with adaptive approach in genetic algorithms

Vedat Toğan*, Ayşe T. Daloğlu

Karadeniz Technical University, Department of Civil Engineering, 61080 Trabzon, Turkey

Received 2 February 2005; received in revised form 14 November 2005; accepted 16 November 2005
Available online 18 January 2006

Abstract

This paper discusses the adaptive approach in genetic algorithms (GAs). It is tried to show how the adaptive approach affects the performance of GAs, suggesting some improvements in both the penalty function, and mutation and crossover. A strategy is also considered for member grouping to reduce the size of the problem. Some practical design of space truss examples taken from technical literature are optimized by the algorithm suggested in the current work. Design constraints such as displacement, tensile stress and stability given by national specifications are incorporated and the results are compared with the ones obtained by previous studies. It is concluded that the member grouping together with the adaptive approach increase the probability of catching the global solution and enhance the performance of GAs.

Keywords: Adaptive approach; Optimization; Space truss; Genetic algorithms; Structural optimization

1. Introduction

Due to the fact that material cost is one of the major factors in the construction of a building, it is preferable to reduce it by minimizing the weight or volume of the structural system. All of the methods used for minimizing the volume or weight intend to achieve an optimum design having a set of design variables under certain design criteria. It is necessary to understand the characteristics of the problem to select an appropriate optimization method for structural design. The important characteristic of structural design optimization is that the solution sought is the global optimal solution [1] and the design variables are discrete and must be chosen from a predetermined set.

The genetic algorithm (GA) that differs from other classical optimization in four ways [2] is a part of evolutionary computational technique and probabilistic and global search method. Due to these advantages, the GA has been preferred in wide ranges of optimization problems among researchers [3–9]. In order to apply the genetic algorithm, a population of solutions within a search space is initialized in contrast to the traditional optimization methods that start from a single point solution. The population can be viewed as points in the search space of all solutions to the optimization problem. Each individual in population has a fitness value defined by a fitness function. Then the artificial evolution processes, called the genetic loop, which mimic natural evolution are applied to produce new candidate solutions. At the end of the process, the newly created generation replaces the previous generation and revolution is repeated until a satisfying solution to the problem is obtained ensuring certain design criteria are satisfied or a maximum number of generations are reached.

In this study a new adaptive penalty scheme and adaptive mutation and crossover are proposed. A strategy is also adopted for member grouping. Thus, the intention is to be protected from becoming stuck on a local optimum, and to get close to the global optimum instead.

2. Adaptive penalty scheme

It is well-known that the GA is an unconstrained optimization method and it cannot explicitly handle constraints of the optimization problem. Therefore, the objective function of a structural optimization problem involves the penalty function which penalizes the design variables depending on the degree of violation of the constraints. Since there is not a unique way to define the penalty term, different forms of the penalty functions have been considered in the literature [10–15]. In all penalty schemes, the degree of penalty depends on the values

* Corresponding author. Tel.: +90 462 377 2671; fax: +90 462 377 2606.
E-mail address: togan@ktu.edu.tr (V. Toğan).

0141-0296/$ - see front matter
doi:10.1016/j.engstruct.2005.11.007

of the various coefficients treated as pre-defined constants during the calculation of the penalty function. Nanakorn and Meesomklin [1] proposed a new adaptive penalty function that will be able to adjust itself automatically during the evolutionary process in such a way that the desired degree of penalty is always obtained. Erbatur et al. [14] modified the penalty function presented by Joines and Houck [16] by normalizing the constraints. Therefore, the penalty given to the individual becomes very small [14] when the power of the constraints is imposed. Chen and Rajan [15] made some enhancements in the simple GA in order to increase the efficiency, reliability and accuracy of the methodology for code-based design of structures. They proposed an algorithm where the penalty weight is computed automatically and adjusted in an adaptive manner.

Thus far the coefficients given to the optimization problem with the GA are pre-defined and values of these coefficients are obtained by trail and error in most cases. The degree of penalty can be controlled by varying the values of the penalty parameters. It is impossible to judiciously select appropriate values for them. Even though, in common practice, one value is used for all coefficients, which significantly simplifies situation, the appropriate value of this one coefficient is still not obvious [1].

One of the main objectives of this work is to propose an adaptive penalty scheme that will be able to adjust itself automatically during the genetic process. Mathematical formulation of the penalty scheme is presented as follows,

$$\Phi(X) = F(X)(1 + \text{penalty}) \tag{1}$$

$$\text{penalty} = (g_{\max} + g(i))/(g_{\max} - g_{\text{ave}}) \qquad g(i) \geq g_{\text{ave}} \tag{2a}$$

$$\text{penalty} = (g_{\text{ave}} + g(i))/(g_{\text{ave}} - g_{\min}) \qquad g(i) < g_{\text{ave}} \tag{2b}$$

$$\text{penalty} = 0 \qquad g(i) = 0 \qquad i = 1, \ldots, n. \tag{2c}$$

In Eqs. (1) and (2), X is the vector of design variables, $F(X)$ is the objective function for minimum volume, $g(i)$ is the ith individual of unconstraint on the structural response, n represents the number of constraints, and $g_{\max}$, $g_{\min}$, and g_{ave} represent, respectively, maximum, minimum, and average violation value of the generation. Finally, $\Phi(X)$ is the modified objective function. The formulation of the unconstrained optimization problem is based on the violations of normalized constraints.

The adaptive penalty function given in Eq. (2) does not include a pre-defined coefficient or specified coefficient by trial and error. Also magnitudes of the violations are not characterized by a static rate for both near feasible and infeasible solutions. With the expressions in Eq. (2), instead of penalizing all of the infeasible solutions with the same rate, when the level of the violation of infeasible solution tends to get bigger, the magnitude of the penalty tends to get heavier. Thus the proposed penalty scheme is an adaptive approach and it adjusts itself from individual to individual and from generation to generation. It is known that the GA evolves a population of potential solutions for an optimization problem. The penalty functions presented in technical literature so far do not include the population, whereas, in this study, it is possible to establish a relationship between the penalty scheme and the population.

It is shown in Eq. (2) that the magnitude of the penalty increases as the violation value gets closer to $g_{\max}$. On the other hand it decreases as the violation value gets closer to $g_{\min}$. Thus, some infeasible individuals that are close to the feasible region in the search space will not disappear through the penalty scheme, and they will find a chance to survive. This may sustain the capacity of finding the global solution of the design problem of the GA.

In the GA, each solution in the population gets a fitness value after the penalty scheme is processed. And then the mating pool is formed of the solutions that have the average fitness value or higher. Solutions collected in the mating pool are imposed in a process known as "Structural Information Exchange". This process is performed by genetic operators, such as crossover, mutation, and elitism. Without an operator of this type some possibly important regions of the search space may never be explored [12]. Due to their power on GA functioning, various types of genetic operators have been proposed.

In addition to a new penalty scheme, adaptive mutation and crossover operators are proposed to obtain a global optimum or to get close to it.

3. Adaptive mutation and crossover

The genetic operators are applied to produce new candidate solutions or a better solution than previous one. New genetic operators or some suggestions to improve the previous ones are aimed to increase the performance of the GA. However, both in the simple GA and in improved versions of the GA, the crossover and mutation operators that generally take more attention than the other genetic operators are applied with pre-defined rates that are imposed on the algorithm by the designer. Although both the choice of mutation (p_m) and crossover (p_c) probability critically affect the performance of the GA, there are no fixed probabilities of these parameters. p_m and p_c are used with a specified interval in the literature. Traditional crossover and mutation operators are based on a randomization mechanism, i.e., generating a cut point, and determining the position of the bit shifted by mutation of the solution. But this is not the case in natural evaluation which is mimicked by the GA. Actually renewing the bits of the solution is dynamic or adaptive, but not random.

The slightly modified adaptive probabilities of crossover and mutation given by Srinivas and Patnaik [17] are used in the study to choose the probability of mutation and crossover according to the fitness value of the solutions and to relieve the user.

The modified expression for p_m and p_c are as follows:

$$p_m = 0.5(f_{\max} - f)/(f_{\max} - f_{\text{ave}}) \qquad f \geq f_{\text{ave}} \tag{3a}$$

$$p_m = (f_{\text{ave}} - f)/(f_{\text{ave}} - f_{\min}) \qquad f < f_{\text{ave}} \tag{3b}$$

$$p_c = (f_{\max} - f')/(f_{\max} - f_{\text{ave}}) \qquad f' \geq f_{\text{ave}} \tag{3c}$$

$$p_c = 1.0 \qquad f' < f_{\text{ave}}. \tag{3d}$$

Here, f is the fitness of an individual, f_{ave} the average fitness value of the population, and $f_{\max}$ and $f_{\min}$ the maximum and minimum fitness value of the population respectively. f' is the larger of the fitness values of the solutions to be crossed.