Computers and Structures 87 (2009) 435–443

Contents lists available at ScienceDirect

Computers and Structures

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

A heuristic particle swarm optimization method for truss structures with discrete variables

L.J. Li *, Z.B. Huang, F. Liu

Faculty of Construction Engineering, Guangdong University of Technology, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 26 January 2008
Accepted 8 January 2009
Available online 18 February 2009

Keywords:
Heuristic particle swarm optimization
Discrete variables
Truss structures
Size optimization

ABSTRACT

A heuristic particle swarm optimizer (HPSO) algorithm for truss structures with discrete variables is presented based on the standard particle swarm optimizer (PSO) and the harmony search (HS) scheme. The HPSO is tested on several truss structures with discrete variables and is compared with the PSO and the particle swarm optimizer with passive congregation (PSOPC), respectively. The results show that the HPSO is able to accelerate the convergence rate effectively and has the fastest convergence rate among these three algorithms. The research shows the proposed HPSO can be effectively used to solve optimization problems for steel structures with discrete variables.

1. Introduction

In the past 30 years, many algorithms have been developed to solve the structural engineering optimization problems. Most of these algorithms are based on the assumption that the design variables are continuously valued and the gradients of functions and the convexity of the design problem satisfied. However, in reality, the design variables of optimization problems such as the cross-section areas are discretely valued. They are often chosen from a list of discrete variables. Furthermore, the function of the problems is hard to express in an explicit form. Traditionally, the discrete optimization problems are solved by mathematical methods by employing round-off techniques based on the continuous solutions. However, the solutions obtained by this method may be infeasible or far from the optimum solutions [1].

Recently, some papers, published on the subject of the structural engineering optimization, are about the evolutionary algorithms such as the genetic algorithm (GA) [2], the particle swarm optimizer algorithm (PSO) [3,4] and other stochastic search techniques based on natural phenomena [5]. The PSO algorithm has fewer parameters and easier to implement, and it has shown a fast convergence rate than other evolutionary algorithms for some problems [6]. Most of the applications of the PSO algorithm to structural optimization problems are based on the assumption that the variables are continuous. Only in few papers PSO algorithm is used to solve the discrete structural optimization problems [7,8].

This paper presents a heuristic particle swarm optimizer (HPSO) algorithm, which is based on the standard particle swarm optimize (PSO) and the harmony search (HS) scheme, and is applied to the discrete valued structural optimization problems. The HPSO algorithm has all the advantages that belong to the PSO and the PSOPC algorithms. Furthermore, it has faster convergence rate than the PSO and the PSOPC algorithms, especially in the early iterations [9].

This paper introduces the formulation of the discrete valued optimization problems in Section 2. The PSO and the PSOPC algorithms for the discrete valued variables are presented in Sections 3 and 4, respectively. The HPSO algorithm for the discrete valued variables is introduced in Section 5, and it is tested on several examples in Section 6. Section 7 concludes this paper.

2. Mathematical model for discrete structural optimization problems

A structural optimization design problem with discrete variables can be formulated as a nonlinear programming problem. In the size optimization for a truss structure, the cross-section areas of the truss members are selected as the design variables. Each of the design variables is chosen from a list of discrete cross-sections based on production standard. The objective function is the structure weight. The design cross-sections must also satisfy some inequality constraints equations, which restrict the discrete variables. The optimization design problem for discrete variables can be expressed as follows:

* Corresponding author. Tel.: +86 20 39322513; fax: +86 20 39322511.
E-mail address: lilj@scnu.edu.cn (L.J. Li).

0045-7949/$ - see front matter
doi:10.1016/j.compstruc.2009.01.004

$$\begin{array}{ll} \min & f(x^1,x^2,\ldots,x^d), \quad d=1,2,\ldots,D \\ \text{subjected to}: & g_q(x^1,x^2,\ldots,x^d) \leqslant 0, \quad d=1,2,\ldots,D, \\ & q=1,2,\ldots,M \\ & x^d \in S_d = \{X_1,X_2,\ldots,X_p\} \end{array}$$

where $f(x^1,x^2,\ldots,x^d)$ is the truss's weight function, which is a scalar function. And $x^1,x^2,\ldots,x^d$ represent a set of design variables. The design variable x^d belongs to a scalar S_d, which includes all permissive discrete variables $\{X_1,X_2,\ldots,X_p\}$. The inequality $g_q(x^1,x^2,\ldots,x^d) \leqslant 0$ represents the constraint functions. The letter D and M are the number of the design variables and inequality functions, respectively. The letter p is the number of available variables.

3. The particle swarm optimizer (PSO) algorithm for discrete variables

The PSO algorithm was first invented by Kennedy and Eberhart [6]. It is a population-based algorithm with fewer parameters to implement. The PSO algorithm is first applied to the optimization problems with continuous variables. Recently, it has been used to the optimization problems with discrete variables [7]. The optimization problem with discrete variables is a combination optimization problem which obtains its best solution from all possible variable combinations. The scalar S includes all permissive discrete variables arranged in ascending sequence. Each element of the scalar S is given a sequence number to represent the value of the discrete variable correspondingly. It can be expressed as follows:

$$S_d = \{X_1,X_2,\ldots,X_j,\ldots X_p\}, \quad 1 \leqslant j \leqslant p$$

A mapped function $h(j)$ is selected to index the sequence numbers of the elements in set S and represents the value X_j of the discrete variables correspondingly.

$$h(j) = X_j$$

Thus, the sequence numbers of the elements will substitute for the discrete values in the scalar S. This method is used to search the optimum solution, and makes the variables to be searched in a continuous space.

The PSO algorithm includes a number of particles, which are initialized randomly in the search space. The position of the ith particle in the space can be described by a vector x_i,

$$x_i = (x_i^1,x_i^2,\ldots,x_i^d,\ldots,x_i^D), \quad 1 \leqslant d \leqslant D, \quad i=1,\ldots,n$$

where D is the dimension of the particle, and n is the sum of all particles. The scalar $x_i^d \in \{1,2,\ldots,j,\ldots,p\}$ corresponds to the discrete variable set $\{X_1,X_2,\ldots,X_j,\ldots,X_p\}$ by the mapped function $h(j)$. Therefore, the particle flies through the continuous space, but only stays at the integer space. In other words, all the components of the vector x_i are integer numbers. The positions of the particles are updated based on each particle's personal best position as well as the best position found by the swarm at each iteration. The objective function is evaluated for each particle and the fitness value is used to determine which position in the search space is the best of the others. The swarm is updated by the following equations:

$$V_i^{(k+1)} = \omega V_i^{(k)} + c_1r_1(P_i^{(k)} - x_i^{(k)}) + c_2r_2(P_g^{(k)} - x_i^{(k)}) \tag{1}$$

$$x_i^{(k+1)} = INT(x_i^{(k)} + V_i^{(K+1)}) \tag{2}$$

$$1 \leqslant i \leqslant n$$

where $x_i^{(k)}$ and $V_i^{(k)}$ represent the current position and the velocity of each particle at the kth iteration, respectively, $P_i^{(k)}$ the best previous position of the ith particle (called *pbest*) and $P_g^{(k)}$ the best global position among all the particles in the swarm (called *gbest*), r_1 and r_2 are two uniform random sequences generated from $U(0,1)$, and ω the inertia weight used to discount the previous velocity of the particle persevered [10]. The object function and the constraint functions can be expressed by the scalar x_i^d as follows:

$$f(h(x_i^1),h(x_i^2),\ldots,h(x_i^d),\ldots,h(x_i^D))$$
$$g_q(h(x_i^1),h(x_i^2),\ldots,h(x_i^d),\ldots,h(x_i^D))$$

4. The PSOPC algorithm for discrete valued variables

It is known that the PSO may outperform other evolutionary algorithms in the early iterations, but its performance may not be competitive when the number of the iterations increases [11]. He and Wu have improved the particle swarm optimizer with passive congregation (PSOPC), which can improve the convergence rate and accuracy of the PSO efficiently [12]. The PSOPC algorithm is first used in optimization problems with continuous variables [13]. By modification it can be used to the optimization problems with discrete variables [8,14]. The formulations of the PSOPC algorithm for discrete variables can be expressed as follows:

$$V_i^{(k+1)} = \omega V_i^{(k)} + c_1r_1(P_i^{(k)} - x_i^{(k)}) + c_2r_2(P_g^{(k)} - x_i^{(k)}) + c_3r_3(R_i^{(k)} - x_i^{(k)}) \tag{3}$$

$$x_i^{(k+1)} = INT(x_i^{(k)} + V_i^{(k+1)}) \tag{4}$$

$$1 \leqslant i \leqslant n$$

where R_i is a particle selected randomly from the swarm, c_3 the passive congregation coefficient, and r_3 a uniform random sequence in the range $(0,1)$: $r_3 - U(0,1)$.

5. The heuristic particle swarm optimizer (HPSO) for discrete variables

The heuristic particle swarm optimizer (HPSO) algorithm, which is based on the PSOPC algorithm and the harmony search (HS) scheme, is introduced by Li [9] and is first used in continuous variable optimization problems. The HPSO algorithm presented by Li [9] makes it possible that the particle meets the demand of constraints' boundary or the variables' boundary for each fly. Similarly, The HPSO algorithm for the discrete valued variables can be expressed as follows:

$$V_i^{(k+1)} = \omega V_i^{(k)} + c_1r_1(P_i^{(k)} - x_i^{(k)}) + c_2r_2(P_g^{(k)} - x_i^{(k)}) + c_3r_3(R_i^{(k)} - x_i^{(k)}) \tag{5}$$

$$x_i^{(k+1)} = INT(x_i^{(k)} + V_i^{(k+1)}) \tag{6}$$

$$1 \leqslant i \leqslant n$$

where x_i is the vector of a particle's position, and x_i^d is one component of this vector. After the (k + 1)th iterations, if $x_i^d < x^d(LowerBound)$ or $x_i^d > x^d(UpperBound)$, the scalar x_i^d is regenerated by selecting the corresponding component of the vector from *pbest* swarm randomly, which can be described as follows:

$$x_i^d = (P_b)_t^d, \quad t = INT(rand(1,n))$$

where $(P_b)_t^d$ denotes the dth dimension scalar of *pbest* swarm of the tth particle, and t denotes a random integer number. The pseudocode for the HPSO algorithm is listed in Table 1.

6. Numerical examples

In this section, the HPSO algorithm is tested by five truss structures. The algorithm proposed is coded in FORTRAN language and executed on a Pentium 4, 2.93 GHz machine.

The PSO, the PSOPC and the HPSO algorithms for discrete variables are applied to all these examples and the results are compared in order to evaluate the performance of the HPSO