Automation in Construction 69 (2016) 21–33

Contents lists available at ScienceDirect

Automation in Construction

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

A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure

Min-Yuan Cheng [a], Doddy Prayogo [a,b,*], Yu-Wei Wu [a], Martin Marcellinus Lukito [a]

[a] *Dept. of Civil and Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan, ROC*

[b] *Dept. of Civil Engineering, Petra Christian University, Jalan Siwalankerto 121–131, Surabaya 60236, Indonesia*

ARTICLE INFO

Article history:
Received 17 August 2015
Received in revised form 20 April 2016
Accepted 22 May 2016
Available online 2 June 2016

Keywords:
Harmony Search algorithm
Global-best PSO
Neighbourhood search
Hybrid Harmony Search algorithm
Discrete sizing optimization of truss structures

ABSTRACT

This paper presents a new variant of the Harmony Search (HS) algorithm. This Hybrid Harmony Search (HHS) algorithm follows a new approach to improvisation: while retaining HS algorithm Harmony Memory and pitch adjustment functions, it replaces the HS algorithm randomization function with Global-best Particle Swarm Optimization (PSO) search and neighbourhood search. HHS algorithm performance is tested on six discrete truss structure optimization problems under multiple loading conditions. Optimization results demonstrate the excellent performance of the HHS algorithm in terms of both optimum solution and the convergence behaviour in comparison with various alternative optimization methods.

1. Introduction

Structural optimization has gained much attention because of its direct applicability to the design of structures [16]. Number of design variables, size of search area, and number of design constraints are factors that influence the time needed by designers to find optimized designs. Designers and owners desire optimized structures in order to reduce building structure costs [20]. Optimized structures should minimize the cost of a structure while meeting code-specified behaviour and performance requirements. Optimization allows to yield better designs at the lowest cost in terms of time and money.

Most recent studies on optimal structure designs have adopted continuous variables [7,8,21,23,33]. However, the availability of standard member sizes and precision limitations inherent in the modern steel manufacturing sector suggests to select cross-sectional areas from an available list of discrete values. Discrete optimization problems are far more difficult to solve than continuous problems [25,35]. Traditionally, researchers have used mathematical methods that employ rounding off techniques based on continuous solutions to solve discrete optimization problems. However, these methods may become infeasible or generate increasingly suboptimal solutions with larger numbers of variables [26]. This drawback has led researchers to rely on simulation-based metaheuristic algorithms to solve engineering optimization problems. Metaheuristic algorithms combine rules and randomness to imitate natural phenomena and try to find the optimum design using 'trial and error' in a reasonable amount of computing time [40]. The capability to balance intensification and diversification during a search determines the efficiency of a specific metaheuristic algorithm. Intensification (exploitation) aims to identify the best solution and select during the process a succession of best candidates/solutions. Diversification (exploration) ensures, usually by randomization, that the algorithm explores the search space efficiently. To address global search needs, modern metaheuristic algorithms have evolved to incorporate 3 main purposes: solving problems faster, solving larger problems, and enhancing algorithm robustness [5,13]. Modern metaheuristic algorithms include: Genetic Algorithms (GA) [18], Particle Swarm Optimization (PSO) [22], Differential Evolution (DE) [34], Artificial Bee Colony (ABC) [19], Bees Algorithm (BA) [29], Firefly Algorithm [39], Cuckoo Search (CS) [41], and Symbiotic Organisms Search (SOS) [4], among others.

Rather drawing its inspiration from biological or physical processes, the HS algorithm originally proposed in [14] is inspired by an artistic-creative process. The HS algorithm conceptualizes the behaviour of musicians searching for harmony and then continuing to refine their tune to achieve an increasingly better state of harmony. Musical harmony is analogous to an optimization solution vector and a musician's improvisations are analogous to local and global search schemes in optimization techniques. Due to its ease of application and simplicity, the HS algorithm has garnered growing attention and been successfully employed to a wide variety of practical structural optimization problem, such as truss structures [24], steel sway frames [9], grillage systems [32],

---

* Corresponding author at: Dept. of Civil Engineering, Petra Christian University, Jalan Siwalankerto 121–131, Surabaya 60236, Indonesia.

E-mail addresses: myc@mail.ntust.edu.tw (M.-Y. Cheng), doddyprayogo@ymail.com (D. Prayogo), d9305503@mail.ntust.edu.tw (Y.-W. Wu), martin.marcello@hotmail.com (M.M. Lukito).

http://dx.doi.org/10.1016/j.autcon.2016.05.023
0926-5805/

cellular beams [11], and web-expanded beams [10]. In comparison to earlier metaheuristic algorithms, the HS algorithm imposes fewer mathematical requirements and is easily adopted to solve various engineering optimization problems. In addition, this algorithm does not require initial values for decision variables, thus it may escape the local optima. The HS algorithm generates a new vector after considering all existing vectors based on the Harmony Memory Considering Rate (HMCR) and Pitch Adjusting Rate (PAR) rather than considering only two (parents) as in the Genetic Algorithm. Furthermore, instead of using gradient search, the HS algorithm uses stochastic random search based on HMCR and PAR, which obviates the need for derivative information [15]. These features increase HS algorithm flexibility and produce better solutions. Although several variants of the HS algorithm have been proposed, their effectiveness in dealing with diverse problems remains unsatisfactory [38].

While the HS algorithm is good at identifying the high performance regions of a solution space in a reasonable amount of time, this algorithm performs local searches for numerical applications poorly [28]. To improve the local search ability of the HS algorithm, this paper proposes a new algorithm called the Hybrid Harmony Search (HHS) algorithm. The HHS algorithm integrates the memory consideration and pitch adjustment process of the HS algorithm with Global-best PSO and neighbourhood search. Six classical truss design problems with sizing variables are solved in this study in order to demonstrate the efficiency of the HHS algorithm. It is shown that the present algorithm is very competitive with other optimization methods documented in literature.

The remainder of this paper is organized as follows: Section 2 presents the formulation of the discrete sizing optimization problem; Section 3 briefly reviews the HS and IHS algorithm; Section 4 describes the HHS algorithm in detail; Section 5 describes the test problems and discusses the optimization results; and Section 6 presents conclusions and recommendations for future research.

2. Discrete structural optimization problems

Since many problems in engineering have multiple solutions selecting. Discrete sizing optimization attempts to find the optimal cross-section of system elements in order to minimize structural weight. However, the minimum design must also satisfy inequality constraints that limit design variable sizes and structural responses [25].

The discrete structural optimization problem for a truss structure may be formulated as:

$$\text{Find } \begin{array}{l} A = [A_1, A_2, \ldots, A_{ng}], \\ A_i \in D_i, D_i = [d_{i,1}, d_{i,2}, \ldots d_{i,r(i)}] \end{array}$$

To minimize

$$W(A) = \sum_{i=1}^{nm} \gamma_i \times A_i \times L_i \quad (1)$$

$$\text{Subject to } \begin{array}{ll} \sigma_{\min} \le \sigma_i \le \sigma_{\max} & i = 1, 2, \ \ldots, n \\ \delta_{\min} \le \delta_i \le \delta_{\max} & i = 1, 2, \ \ldots, m \end{array}$$

where: A represents the set of design variables; Di is an allowable set of discrete values for design variable A_i; ng is the number of design variables or member groups; r(i) is the number of available discrete values for the ith design variable; W(A) is the weight of the structure; n is the number of component members in the structure; m is the number of nodes; γ_i is the material density of member i; L_i is the length of member i; δ_i is the nodal displacement/deflection at node i; σ_i is the stress developed in the i-th element; and min and max represent the lower and upper bounds, respectively.

The optimum design of truss structures must satisfy optimization constraints stated in Eq. (1). This paper uses a constraint handling procedure developed by Deb [6] to handle the problem-specific constraints. This procedure consists of the following three rules:

- Rule 1: Any feasible solution is preferred to any infeasible solution.
- Rule 2: Between two feasible solutions, the one having the better objective function value is preferred.
- Rule 3: Between two infeasible solutions, the one having the smaller constraint violation is preferred.

The first and third rules orient the search toward feasible regions. The second rule orients the search toward the feasible region with good solutions.

3. Harmony Search algorithm

3.1. Harmony Search algorithm

Harmony Search (HS) algorithm is a metaheuristic algorithm that imitates the natural music improvisation process that musicians use to achieve a perfect state of harmony such as that achieved during jazz improvisation. The HS algorithm holds several important advantages over other competing algorithms and has been applied successfully to a wide variety of optimization problems. Key advantages include ability to handle both discrete and continuous variables, conceptual simplicity, ease of implementation, and few parameters requiring adjustment.

The HS algorithm uses an optimization process to attain a global solution defined by an objective function similar to the way musicians attain aesthetic harmony as defined by an aesthetic standard. Each musician corresponds to one decision variable; the pitch range of musical instruments corresponds to the value range of the decision variable; musical harmony at a certain time corresponds to the solution vector at a certain iteration; and audience aesthetics correspond to the objective function.

In musical improvisation, each players plays at any pitch within the possible range, creating one harmony vector. If all pitches are in good harmony, the experience is stored in the memory of each player and the possibility of creating good harmony increases in the subsequent timeframe. In engineering optimization, each decision variable initially chooses any value within the possible range to create a solution vector. If all variable values create a good solution, the design is stored in the memory of each variable and the possibility of creating a good solution increases in the subsequent timeframe.

When a musician improves the musical harmony, he or she has three possible options: (1) playing any known tune exactly from memory, (2) playing a tune similar to a known tune, (3) composing a new tune at random. These three options correspond to the three main HS algorithm concepts of: memory consideration, pitch adjustment, and randomization. In general, the HS algorithm procedure consists of 5 steps:

Step 1. Initialize the problem and algorithm initial parameters.

The optimization problem is defined as Minimize f(x) subject to $LB_i \le Xi \le UB_i$, in which i = 1, 2,...., N. LB_i and UB_i are the lower and upper bounds for the decision variables. This step also specifies the HS algorithm parameters, including Harmony Memory Size (HMS), Harmony Memory Consideration Rate (HMCR), Pitch Adjusting Rate (PAR), bandwidth (bw), and number of improvisations (NI) or stopping criterion. The NI equals the total number of function evaluations. A good set of parameters will improve the ability of the algorithm to search for the global optimum with a high convergence rate.

Step 2. Initialize the Harmony Memory (HM).

The second step is Harmony Memory initialization. The HS algorithm has memory storage, called Harmony Memory (HM), in which