Computers and Structures 165 (2016) 1–9

Contents lists available at ScienceDirect

Computers and Structures

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

Dolphin monitoring for enhancing metaheuristic algorithms: Layout optimization of braced frames

A. Kaveh *, N. Farhoudi

Centre of Excellence for Fundamental Studies in Structural Engineering, School of Civil Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Iran

ARTICLE INFO

Article history:
Received 20 March 2015
Accepted 23 November 2015
Available online 24 December 2015

Keywords:
Dolphin Monitoring (DM)
Layout optimization
Steel braced frames
Meta-heuristic algorithms

ABSTRACT

In this study, Dolphin Monitoring (DM) is utilized to improve the performance of the metaheuristic algorithms for layout optimization of structures. DM is a method to control the progress of metaheuristic algorithms using some features of Dolphin Echolocation Optimization (DEO) rules. In the present work, DM is incorporated in GA, ACO, PSO, BB-BC, CBO and ECBO and applied to layout optimization of steel braced frames. Three frames are studied to show how the use of DM improves the results of the standard versions of all these algorithms.

1. Introduction

Structural optimization helps engineers to design more economical structures spending less computational time and human effort. Metaheuristic algorithms are robust tools for structural optimization. Genetic Algorithms (GA) [1,2], Simulated Annealing (SA) [3], Ant Colony Optimization (ACO) [4], Differential Evolution (DE) [5], Harmony Search (HS) algorithm [6], Particle Swarm Optimization (PSO) [7], Charged System Search (CSS) method [8], Bat algorithm (BA) [9], Water Cycle Algorithm (WCA) [10], Ray Optimization (RO) [11], Krill-herd algorithm (KA) [12], Dolphin Echolocation Optimization (DEO) [13], Colliding Bodies Optimization (CBO) [14], Enhanced Colliding Bodies Optimization (ECBO) [15] are some of the meta-heuristic algorithms. These methods have found applications in optimal design of structures [16–18].

In this paper, layout optimization (simultaneous size and topology optimization) of dual systems is studied. A dual building frame system is a structural system with an essentially complete space frame providing support for gravity loads. Layout optimization of braced frames of this study includes finding the best placement for bracings and the best cross sections for elements of a dual system of moment frames together with X-bracings. Benchmark examples of this study are optimized by GA, ACO, PSO, BB-BC, CBO and their modified variants [19]. These examples have also been studied by DE and DEO, Kaveh and Farhoudi [20]. In this study, Dolphin Monitoring (DM) is presented to show how the performance of metaheuristic algorithms can be improved by taking their convergence under control. DM helps metaheuristic algorithms to perform a global search and prevents them from being trapped in local optima. When DM is incorporated in a metaheuristic, the dependency of the algorithm on its own parameters gets reduced. DM can perform the same task for all metaheuristic algorithms and has only one parameter for controlling the convergence rate.

Numerical examples show that the performance of all the above mentioned optimization algorithms are improved by the addition of dolphin monitoring.

This paper consists of seven sections. First section is devoted to introduction. In the second section, GA, ACO, PSO, BB-BC and CBO are briefly discussed. In the third section, dolphin monitoring is presented. In the fourth section, formulation for the optimization problem is provided. In the fifth and sixth sections, numerical examples are presented and discussed. Last section is devoted to the concluding remarks.

2. Meta-heuristic algorithms utilized in this study

2.1. Genetic algorithm

The main steps of Genetic algorithm are as follows:

1. Initiation: Individuals of a population that are mimicked as chromosomes are selected randomly from search space.
2. Fitness based selection: Chromosomes with better fitnesses are selected to produce the next population.

* Corresponding author. Tel.: +98 21 44202710; fax: +98 21 77240398.
E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

http://dx.doi.org/10.1016/j.compstruc.2015.11.012
0045-7949/

3. Crossover: Each pair of selected chromosomes (considered as parents) are replaced by a pair of new chromosomes (considered as children) in a way that each gene of child is gained from one of the parents.
4. Mutation: Some genes of a chromosome are changed randomly by a specified probability.
5. Termination: Steps 2 to 4 are repeated until convergence criterion is satisfied.

2.2. Ant colony optimization

Trace of ants is a kind of hormone called pheromone that is layed on their trails from nest to food source, and as the pheromone on a specific path increases, the probability of that path to be chosen by ants increases. This idea is applied in ACO to define optimum design of a problem.

The main steps of the ant colony optimization algorithm are as follows:

1. Initiation: Individuals that are mimicked as the ants' paths are selected randomly from search space.
2. Fitness based pheromone intensity adjustment: Fitness of each path is calculated and the amount of pheromone on each path increases according to its fitness (in some variants of ACO, in this step a little amount of pheromone is added to all paths).
3. Evaporation: The amount of pheromone of previously passed paths decreases according to the evaporation rules.
4. Probability calculation: Probability of each path to be chosen in the next loop is calculated based on its pheromone amount.
5. Termination: Steps 2 to 4 are repeated until convergence criterion is satisfied.

2.3. Particle swarm optimization

The main steps of the particle swarm optimization algorithm are as follows:

1. Initiation: Individuals that are mimicked as the position of some particles in the swarm are selected randomly from search space.
2. Fitness based position update: velocity vectors are defined to lead particles toward both global and local best positions. Particles' positions are updated by their velocities.
3. Step 2 is repeated until convergence criterion is satisfied.

2.4. Big Bang–Big Crunch

The Main steps of Big Bang–Big Crunch optimization algorithm are as follows:

1. Initiation (Big Bang phase): Individuals that are mimicked as positions of some particles in a search space are selected randomly.
2. Center of mass calculation: Fitness of each particle is calculated and is considered as its mass. Center of mass of particles is calculated.
3. Position update (Big Crunch phase): New positions are generated around the center of mass, in a specified radius which decreases in each loop.
4. Steps 2 to 3 are repeated until convergence criterion is satisfied.

2.5. Colliding bodies optimization

The main steps of colliding bodies optimization are as follows [14]:

1. Initiation: Individuals that are mimicked as positions of some bodies in a search space are selected randomly.
2. Grouping: The magnitude of the body mass for each CB is defined in a way that, good values of cost function will correspond to a larger mass of colliding bodies. Bodies are sorted in ascending order based on their mass and are divided equally into two groups: stationary (better CBs) and moving bodies (worse CBs). For each stationary body, a moving body is paired to perform one by one collision. It should be noted, before collision, velocity of stationary bodies are considered as zero and velocity of moving parts are calculated according to the distance between them and their pair in stationary group.
3. Collision: Collision occurs between each moving body and its corresponding stationary body. After collision, the new positions of the colliding bodies are updated based on the collision laws.
4. Steps 2 to 3 are repeated until convergence criterion is satisfied.

It should be added that in this study Enhanced form of CBO namely ECBO is also applied on numerical examples. ECBO have a memory to save better results of CBO, in addition it takes advantage of a random vector to perform a global search [15].

3. Dolphin monitoring

The successful execution of metaheuristic algorithms mostly depends on proper tuning of their parameters, and the results may differ considerably for different values of the selected parameters. Studies show that the Convergence Factor (CF) is a good representative for convergence rate of an optimization algorithm and also metaheuristic algorithms perform reasonable when their convergence factor is properly controlled [19].

Dolphin Echolocation Optimization is a metaheuristic algorithm based on convergence rate control. In this study dolphin monitoring is presented to apply Dolphin Echolocation Optimization (DEO) rules to metaheuristic algorithms, in order to take the convergence rate of the algorithms under control. DM is added as a function at the end of each loop of metaheuristic algorithms. This addition, changes nothing in the original algorithm. When DM is incorporated, the algorithm can be set to converge in a predefined number of loops. DM has only one parameter to be set and that is the convergence rate.

Thus user can define a curve for convergence of the algorithm and persuade the algorithm to follow this curve. In this study, Eq. (1) is chosen as the convergence curve formula.

$$PP_i = 30 + 70 * \frac{(i^{Power} - 1)}{(NL^{Power} - 1)} \quad (1)$$

where PP_i is the predefined probability of the ith loop and *Power* is the variable that controls the convergence speed of algorithm. *NL* is number of loops. Changes in *PP* by alteration of the *Power* are depicted in Fig. 1. It can be seen that the convergence rate decreases by an increase in the value of the *Power*.

In this study, GA, ACO, PSO, BB-BC and CBO with dolphin monitoring being incorporated are called GA-DM, ACO-DM, PSO-DM, BB-DM and CBO-DM, respectively.

3.1. Dolphin monitoring algorithm

At the end of each iteration perform these two steps:

1. Calculate *PP* according to Eq. (1).
2. For each variable, calculate the mode value in the entire population for the given variable. Most frequent value in a data set is called the mode value.