Annals of Nuclear Energy 64 (2014) 144–151

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

Bat algorithm for the fuel arrangement optimization of reactor core

S. Kashi, A. Minuchehr, N. Poursalehi *, A. Zolfaghari

Nuclear Engineering Department, Shahid Beheshti University, G.C., P.O. Box 1983963113, Tehran, Iran

ARTICLE INFO

Article history:
Received 1 September 2013
Received in revised form 21 September 2013
Accepted 25 September 2013
Available online 22 October 2013

Keywords:
Bat algorithm
Multiplication factor
Power peaking factor
Fuel arrangement optimization

ABSTRACT

In this paper, we develop a novel optimization algorithm, Bat Algorithm (BA), in order to implement in the Loading Pattern Optimization (LPO) of nuclear reactor core. For performing the fuel management optimization, we define a fitness function considering the multiplication factor maximizing and power peaking factor minimizing objectives simultaneously. For this purpose, we prepared a computer program i.e. Bat Algorithm Nodal Expansion Code (BANEC) in order to gain the possible maximum fitness value for the LPO operation. Fuel arrangement optimization using BANEC has been performed for two PWR test cases including KWU and BIBLIS reactors. Numerical results of BANEC confirm that the BA has a great strength to obtain a semioptimized core pattern as respect to considered objective functions during suitable consuming run time. At last, the results show that BA is a very promising algorithm for LPO problems and has the potential to use in other nuclear engineering optimization problems.

1. Introduction

In a commercial nuclear power plant, the maximization of profit is the main goal to improve the economics while satisfying the safety constraints such as the radial power peaking factor, the maximum burn-up, and the moderator temperature coefficient. Optimization in the use of the nuclear fuel can result in considerable cost savings. The main concerns in nuclear fuel-management optimization are the loading location of different fuel assemblies, burnable poisons and/or the control rods in the reactor core. The LP optimization is difficult due to the massive possible search space and the complexity of the objective function. Because of non linear nature and multimodality of fuel-management optimization problem, the stochastic optimization methods are desirable to find global optimum solution.

Artificial intelligence methods do not use knowledge of how the objectives are related to the control variables, but only ask for an evaluation of the candidate solutions, from whatever methods are available. Typical examples are as simulated annealing (Parks, 1987, 1990; Smuc et al., 1994; Stevens et al., 1995; Kropaczek and Turinsky, 1991; Kropaczek et al., 1991, etc.), genetic algorithms (Poon and Parks, 1993; Parks, 1996; Chapot et al., 1999; Erdogan and Geckinli, 2003; Pereira and Lapa, 2003; Ziver et al., 2004, etc.), artificial neural networks (Faria and Pereira, 2003; Ortiz and Requena, 2004), tabu search (Lin et al., 1998; Del Campo and Francois, 2002; Castillo et al., 2004); ant colony optimization (Machado and Schirru, 2002; Lin and Lin, 2012), particle swarm optimization (Yadav and Gupta, 2011; Babazadeh et al., 2009), improved pivot particle swarm method (Liu and Cai, 2012), artificial bee colony algorithm (Oliveira and Schirru, 2011), harmony search algorithm (Poursalehi et al., 2013a), self-adaptive global best harmony search algorithm (Poursalehi et al., 2013b), differential harmony search algorithm (Poursalehi et al., 2013c), continuous firefly algorithm (Poursalehi et al., 2013d), discrete firefly algorithm (Poursalehi et al., 2013e), etc.

Bat Algorithm (BA) is a new metaheuristic method such as particle swarm optimization, firefly algorithm and harmony search. This algorithm is based on the echolocation behavior of bats. BA is potentially more powerful than other methods such as GA and PSO as well as HS. The primary reason is that BA uses a good combination of major advantages of these algorithms in some way. Moreover, PSO and HS are the special cases of BA under appropriate simplifications, (Yang and Gandomi, 2012). However, one can revert to Yang and Gandomi (2012) for obtaining more details.

In this work, the Bat algorithm is developed for the LP optimization problem. The proposed algorithm, BA, has been used successfully for the multi-objective optimization of fuel loading pattern design based on possible minimizing the radial power peaking factor associated with maximizing the effective multiplication factor (K_{eff}). For evaluating the proposed approach, a Bat Algorithm Nodal Expansion Code (BANEC) is prepared in order to optimize the fuel management of nuclear reactor core. In BANEC, the two-dimensional, multi-group diffusion equation is solved by coarse nodes i.e. one node per a fuel assembly (FA) within the framework of Average Current Nodal Expansion Code (ACNEC). This code already has been validated against various benchmarks, (Poursalehi et al., 2012, 2013f). The important advantage of nodal code on using in BANEC is to decrease the computational time of neutronic

* Corresponding author. Fax: +98 21 29902546.
E-mail address: npsalehi@yahoo.com (N. Poursalehi).

0306-4549/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.anucene.2013.09.044

reactor core calculations. For the validation and demonstration of BANEC strength in gaining the semioptimal core pattern, we also have run the Continuous Firefly Algorithm Nodal Expansion Code (CFANEC) (Poursalehi et al., 2013d), for the comparison of results considering the new fitness definition. For two popular LPO problems including BIBLIS and KWU PWRs, the results of BANEC and CFANEC reveal that the BA is a very promising scheme to obtain the near optimal fuel assembly arrangement in regard of considered objective functions.

The remainder of this paper is structured as follows. Section 2 presents the brief description of Bat Algorithm (BA) and Section 3 explains the mapping loading pattern on the proposed BA, in Section 4, our fitness function definition is present and Section 5 gives the results of LPO operation.

2. Bat optimization algorithm

A general view of bat algorithm is presented in this section and also the results of a test case are given for the validation of BA.

2.1. Behavior of micro-bats

Bats are fascinating animals. They are the only mammals with wings and they also have advanced capability of echolocation. Micro-bats use a type of sonar, called, echolocation, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. These bats emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. Their pulses vary in properties and can be correlated with their hunting strategies, depending on the species. Most bats use short, frequency-modulated signals to sweep through about an octave, while others more often use constant-frequency signals for echolocation. Their signal bandwidth varies depends on the species, and often increased by using more harmonics (Yang, 2010).

2.2. Algorithm of bat

If we idealize some of the echolocation characteristics of micro-bats, we can develop various bat inspired algorithms or bat algorithms. For simplicity, we now use the following approximate or idealized rules:

1. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way;
2. Bats fly randomly with velocity v_i at position x_i with a fixed frequency $f_{\min}$, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target.

As the speed of sound in air is typically $v = 340$ m/s, the wavelength λ of the ultrasonic sound bursts with a constant frequency f is given by.

$$\lambda = \frac{v}{f} \tag{1}$$

3. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value $A_{\min}$.

Based on these approximations and idealization, the basic steps of the Bat Algorithm (BA) can be summarized as the pseudocode shown in Fig. 1 that is defined by Yang (2010). He rules how their positions x_i and velocities v_i in a d-dimensional search space are updated. The new solutions x_i^t and velocities v_i^t at time step t are given by:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{2}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \tag{3}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{4}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. Here x_* is the current global best location (solution) which is located after comparing all the solutions among all the n bats. As the product $\lambda_i f_i$ is the velocity increment, we can use either f_i (or λ_i) to adjust the velocity change while fixing the other factor λ_i (or f_i), depending on the type of the problem of interest. In our implementation, we will use $f_{\min} = 0$ and $f_{\max} = 100$, depending the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from $(f_{\min}, f_{\max})$.

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk:

$$x_{new} = x_{old} + \varepsilon A^t \tag{5}$$

where $\varepsilon \in [-1, 1]$ is a random number, while $A^t = \langle A_i^t \rangle$ is the average loudness of all the bats at this time step.The update of the velocities and positions of bats have some similarity to the procedure in the standard particle swarm optimization (Yadav and Gupta, 2011) as f_i essentially controls the pace and range of the movement of the swarming particles. To a degree, BA can be considered as a balanced combination of the standard particle swarm optimization and the intensive local search controlled by the loudness and pulse rate (Yang, 2010).

Furthermore, the loudness A_i and the rate r_i of pulse emission have to be updated accordingly as the iterations proceed. As the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. For example, we can use $A_0 = 100$ and $A_{\min} = 1$. For simplicity, we can also use $A_0 = 1$ and $A_{\min} = 0$, assuming $A_{\min} = 0$ means that a bat has just found the prey and temporarily stop emitting any sound. Now we have:

$$A_i^{t+1} = \alpha A_i^t \tag{6}$$

```
Bat Algorithm
 Objective function f(x), x=(x_1, ..., x_d)
 Initialize the bat population x_i (i=1,2, ..., n) and v_i
 Define pulse rates r_i and the loudness A_i
 While( t< Max number of iterations)
 Generate (new solutions by adjusting frequency,
 and updating velocities and locations/solution [equation 1 to 3]
        if rand > r_i
        Select a solution among the best solutions
        Generate a local solution around the selected best solution
        End if
        Generate a new solution by flying randomly
        If ( rand < A_i & f(x_i) > f(x_*) )
       Accept the new solutions
       Increase r_i and decrease A_i
      End if
 Rank the bats and find the current best x_*
 end while
 return result
```

Fig. 1. Pseudo-code of BA.