Nuclear Engineering and Design 257 (2013) 161–174

Contents lists available at SciVerse ScienceDirect

Nuclear Engineering and Design

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

Differential harmony search algorithm to optimize PWRs loading pattern

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

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

HIGHLIGHTS

- Exploit of DHS algorithm in LP optimization reveals its flexibility, robustness and reliability.
- Upshot of our experiments with DHS shows that the search approach to optimal LP is quickly.
- On the average, the final band width of DHS fitness values is narrow relative to HS and GHS.

ARTICLE INFO

Article history:
Received 5 April 2012
Received in revised form 24 January 2013
Accepted 26 January 2013

Keywords:
Fuel management
Differential harmony search algorithm
Coarse mesh calculation
Fitness function

ABSTRACT

The objective of this work is to develop a core loading optimization technique using differential harmony search algorithm in the context of obtaining an optimal configuration of fuel assemblies in pressurized water reactors. To implement and evaluate the proposed technique, differential harmony search nodal expansion package for 2-D geometry, DHSNEP-2D, is developed. The package includes two modules; in the first modules differential harmony search (DHS) is implemented and nodal expansion code which solves two dimensional-multi group neutron diffusion equations using fourth degree flux expansion with one node per a fuel assembly is in the second module. For evaluation of DHS algorithm, classical harmony search (HS) and global-best harmony search (GHS) algorithms are also included in DHSNEP-2D in order to compare the outcome of techniques together. For this purpose, two PWR test cases have been investigated to demonstrate the DHS algorithm capability in obtaining near optimal loading pattern. Results show that the convergence rate of DHS and execution times are quite promising and also is reliable for the fuel management operation. Moreover, numerical results show the good performance of DHS relative to other competitive algorithms such as genetic algorithm (GA), classical harmony search (HS) and global-best harmony search (GHS) algorithms.

1. Introduction

In-Core Fuel Management Optimization (ICFMO) is a branch of nuclear engineering which has been studied for more than 40 years. The ICFMO is a problem related to the optimization of nuclear fuel utilization in a Nuclear Power Plant (NPP). Indeed, it is a multi-objective problem which goals are related to economic and safety aspects. For instance, in a reload operation of a pressurized water reactor (PWR), approximately one third of the burned fuel assemblies (FAs) are substituted by fresh FAs during the reload operation, generating a loading pattern (LP) for the next cycle of operation. ICFMO is one of the most challenging areas of nuclear engineering which involves the optimal arrangement of hundreds of fuel assemblies in reactor cores. The optimization of a core arrangement is very important from economical point of view to make the nuclear power generating station competitive. An optimal nuclear reload design may be defined as a configuration which has the maximum cycle length for the given fuel inventory or uses the minimum amount of fissionable materials for the given cycle length while satisfying safety constraints such as limitation on power peaking factor. The main problem in the fuel assembly position determination is the large number of possible combinations for the fuel loading pattern in the core.

Various meta-heuristics or computational intelligence approaches have been developed for loading pattern optimization problem, such as: dynamic programming (Wall and Fenech, 1965), direct search (Stout, 1973), variational techniques (Terney and Williamson, 1982), backward diffusion calculation (Chao et al., 1986), reverse depletion (Downar and Kim, 1986; Kim et al., 1987), linear programming (Stillman et al., 1989), Population-based incremental learning (Baluja, 1994), simulated annealing (Smuc et al., 1994; Mahlers, 1994), harmony search algorithm (Poursalehi et al., 2013a), Ant Colony algorithm for maximizing boron concentration (Machado and Schirru, 2002), genetic algorithms (Yamamoto, 1997; Mohseni et al., 2008).

In 2001, Geem et al. (2001) proposed a new metaheuristic algorithm, (classical) harmony search (HS), which is inspired by the

* Corresponding author. Tel.: +98 21 22431595; fax: +98 21 29902546.
E-mail address: npsalehi@yahoo.com (N. Poursalehi).

0029-5493/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.nucengdes.2013.01.020

phenomenon of musician attuning. The HS algorithm is based on natural musical performance processes that occur when a musician searches for a better state of harmony. In the optimization process, global optimum solution may be found by performing several iterations through different values of decision variables. The quality of the obtained solutions was evaluated by Ayvaz (2009). The harmony in music is analogous to the optimization solution vector, and musician improvisations are analogous to local and global search schemes in optimization techniques. The HS algorithm does not require initial values for the decision variables. Furthermore, instead of a gradient search, the HS algorithm uses a stochastic random search that is based on the harmony memory considering rate and the pitch adjusting rate. Compared to earlier meta-heuristic optimization algorithms, the HS algorithm imposes fewer mathematical requirements, (Poursalehi et al., 2013a) and has been successfully applied to many engineering optimization problems such as pipe-network design, structural optimization and combined heat and power economic dispatch problem (Chakraborty et al., 2009). HS may be viewed as a simple real-coded GA, since it incorporates many important features of GA like mutation, recombination, and selection. The performance of classical HS over numerical benchmark functions like those used in Mahdavi et al. (2007) suffers from stagnation and/or false convergence. Recently, Omran and Mahdavi (2008) tried to improve the performance of HS by incorporating some techniques from swarm intelligence, (Kennedy et al., 2001). The new variant called by them as GHS (global best harmony search) reportedly outperformed other HS variant over the benchmark problems. Fesanghary et al. (2008) tried to improve the local search behavior of HS by hybridizing it with Sequential Quadratic Program (SQP). In the pitch adjustment phase of classical HS, each vector is probabilistically perturbed in random step size of fixed maximum amplitude. This step is quite similar to the mutation process employed for perturbing the search agents in Evolutionary Strategy (ES), (Beyer and Schwefel, 2002). Recently, Chakraborty et al. (2009) proposed a new strategy for perturbing each vector with a differential mutation operator borrowed from the realm of Differential Evolution (DE), (Storn et al., 2005). The new mutation strategy is inspired by Zaharie's seminal work, (Zaharie, 2002), where the author theoretically showed that the differential mutation scheme has greater explorative power than Evolutionary Algorithms (EA), (Ashlock, 2006), type mutation schemes. The new algorithm, DHS (differential harmony search), has been extensively compared with classical HS and GHS, (Chakraborty et al., 2009).

In this research, we developed differential harmony search (DHS) algorithm for the fuel management operation of nuclear reactor core to satisfy an arbitrary objective function along constraints. For this purpose, we prepared differential harmony search Nodal Expansion Package for 2D rectangular geometry (DHSNEP2D) in order to use in the fuel management operation. In this package for treatment of diffusion equation, a neutronic module which solves the two dimensional-multi group diffusion equation using second order of Average Current Nodal Expansion Method (ACNEM), (Poursalehi et al., 2012, 2013b), is developed. In order to demonstrate the performance of DHS method, the flattening of relative power distribution in two PWR test cases is chosen as fitness function although combination of other safety parameters can be taken as an objective function. To show DHS algorithm merit, classical HS and GHS algorithms also developed and implemented in DHSNEP for comparing purpose.

An outline of the remainder of this paper is as follows: Section 2 briefly outlines classical HS, GHS and DHS algorithms. Section 3 gives the mapping reactor core LP on HS; the fitness function for the fuel management optimization is defined in Section 4 and in Section 5, results are given and compared for two test cases and finally the paper is concluded in Section 6.

2. Harmony search optimization algorithms

In this section, classical harmony search (HS), global-best harmony search (GHS) and differential harmony search (DHS) optimization algorithms are reviewed and theirs characteristics given.

2.1. Classical harmony search algorithm

The HS algorithm is based on natural musical performance processes that occur when a musician searches for a better state of harmony, such as during jazz improvisation. Jazz improvisation seeks to find musically pleasing harmony (a perfect state) as determined by an aesthetic standard, just as the optimization process seeks to find a global solution (a perfect state) as determined by an objective function. The pitch of each musical instrument determines the aesthetic quality, just as the objective function value is determined by the set of values assigned to each decision variable.

Generally, when a musician improvises one pitch, usually he (or she) follows any one of three rules: (1) playing any one pitch from his (or her) memory, (2) playing an adjacent pitch of one pitch from his (or her) memory, and (3) playing totally random pitch from the possible sound range. Similarly, when each decision variable chooses one value in the HS algorithm, it follows any one of three rules: (1) choosing any value from the HS memory (defined as memory considerations), (2) choosing an adjacent value of repeated one from the HS memory (defined as pitch adjustments), and (3) choosing totally random value from the possible value range (defined as randomization), (Lee and Geem, 2005).

However, Fig. 1 shows the optimization procedure of the classical HS algorithm, which consists of Steps 1 through 5 i.e.:

2.1.1. Step 1: initializing the optimization problem and algorithm parameters

First, the optimization problem is specified as follows:

$$\begin{array}{ll}\text{Minimize (or Maximize)} & f(\vec{x}) \\ \quad \text{subjicted to } x_i \in X_i & i = 1, \ldots, N.\end{array} \tag{1}$$

where $f(\vec{x})$ = an objective function; x = the set of each decision variable x_i; X_i = the set of possible range of values for each decision variable, that is, $X_i = \{x_i(1), x_i(2), \ldots, x_i(k)\}$ for discrete decision variables $X_i = \{x_i(1) < x_i(2) < \ldots < x_i(k)\}$ or $x_{iL} \le X_i \le x_{iU}$ for continuous decision variables; N = the number of decision variables; and K = the number of possible values for the discrete variables. In our discrete loading pattern optimization problem, x_i is a possible integer number for the position of a FA in the core. Therefore, taking N as the number of FAs that can be dislocated, we have $\{x_i(1) = 1, x_i(k) = N\}$. Furthermore, the HS algorithm parameters which are required to solve the optimization problem (i.e., Eq. (1)) are also specified in this step i.e.: harmony memory size (number of solution vectors, HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR), and termination criterion (maximum number of searches). Here, HMCR and PAR are parameters that are used to improve the solution vector. Both are defined in Step 3.

2.1.2. Step 2: initializing the harmony memory (HM)

In Step 2, the "harmony memory" is filled with randomly generated solution vectors as the size of the HM (i.e., HMS) and sorted by the values of the objective function, $f(x)$, i.e.:

$$\mathrm{HM} = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^{\mathrm{HMS}} \end{bmatrix}. \tag{2}$$