Annals of Nuclear Energy 70 (2014) 56–63

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

A novel approach in optimization problem for research reactors fuel plate using a synergy between cellular automata and quasi-simulated annealing methods

Ramin Barati *

Fars Science and Research Branch, Islamic Azad University, Fars, Iran

ARTICLE INFO

Article history:
Received 2 November 2013
Received in revised form 3 February 2014
Accepted 5 February 2014
Available online 25 March 2014

Keywords:
Multi-objective optimization
Research reactor
Fuel plate
Cellular automata
Quasi-simulated annealing

ABSTRACT

This paper presents a new and innovative optimization technique utilizing combination of cellular automata (CA) and quasi-simulated annealing (QSA) as solver concerning conceptual design optimization which is indeed a multi-objective optimization problem. Integrating CA and QSA into a unified optimizer tool has a great potential for solving multi-objective optimization problems. Simulating neighborhood effects while taking local information into account from CA and accepting transitions based on decreasing of objective function and Boltzmann distribution from QSA as transition rule make this tool effective in multi-objective optimization. Optimization of fuel plate safety design while taking into account major goals of conceptual design such as improving reliability and life-time – which are the most significant elements during shutdown – is a major multi-objective optimization problem. Due to hugeness of search space in fuel plate optimization problem, finding optimum solution in classical methods requires a huge amount of calculation and CPU time. The CA models, utilizing local information, require considerably less computation. In this study, minimizing both mass and deformation of fuel plate of a multipurpose research reactor (MPRR) are considered as objective functions. Results, speed, and qualification of proposed method are comparable with those of genetic algorithm and neural network methods applied to this problem before.

1. Introduction

Optimization is a prominent problem in various branch of science including nuclear engineering which is, in general, to find optimum (minimum, in the present paper) values of a function in a given domain, and to find the value of variables where the optimum is reached in the domain. During a nuclear reactor design, many optimization problems arise from the necessity of reaching high efficiency, availability and safety levels, under a lot of design constraints. Whence, development and improvement of optimization techniques in nuclear power plants have been of special interest to people involved in such field (Arab-Alibeik and Setayeshi, 2003, 2005; Fadaei and Setayeshi, 2008, 2009a,b; Fadaei et al., 2010; Habibiyan et al., 2004; Sadighi et al., 2002a,b; Zaferanlouei et al., 2010).

Multi-objective problems are commonly found in a nuclear reactor design. They arise when in practice multiple criteria in reactor designs are to be dealt with at the same time. In multi-objective optimization one is interested in optimizing under the constraint that several objectives must be taken care of, not only one such as maximization or minimization of a specific fitness function. As these objectives are usually incompatible, one is led to consider tradeoffs in the way in which resources are allocated. Actually, many real-world problems are of this kind.

The common approach for these problems is to seek a solution that satisfies all objects acceptably. In fact the more common situation is where the problem itself is so complex that finding the best possible solution could cost more than the benefit, so the optimization models generally do not attempt to find the best possible solution, but instead seek for extremely good solutions within reasonable cost and time.

A lot of different methods have been proposed dealing with optimization problems each have some pros and cons. Not so far, the use of linear-programming together with perturbation analysis has been used as a popular optimization technique (Rozon and Beaudet, 1992). More recently, due to the high complexity, nonlinearity, multi-modality and, principally, the lack of knowledge about the search domain of most problems involved in nuclear designs, the use of more robust and appropriate techniques such

* Tel.: +98 9171192059.
E-mail address: barati.ramin@aut.ac.ir

http://dx.doi.org/10.1016/j.anucene.2014.02.027
0306-4549/

as cellular automata (CA) have been proposed (Fadaei and Setayeshi, 2009a,b). Regarding this subject, recently CA (Neumann et al., 1966) has been developed in investigating complex systems (Bar-Yam, 1997; Hoekstra et al., 2010; Murtaza et al., 2007). It has been shown that CA forms a general class of models of dynamical systems which are appealingly simple and yet capture a rich variety of behavior. This has made them a favorite tool for studying the generic behavior of and modeling complex dynamical systems.

Combination of CA and simulated annealing (Kirkpatrick et al., 1983) was used successfully in a previously published paper (Fadaei and Setayeshi, 2009a) in fuel loading pattern of VVER-1000 NPP. After this successful combination, a quasi-simulated annealing (QSA) scheme is put forward, which is based on the frame of the CA and benefited from features of SA method.

Integrating CA and QSA is a powerful optimization technique providing mechanisms which are intended to avoid convergence to local optimum.

The obtained results have shown that the use of combination of CA and QSA in such kind of problems enhances the quality of optimization outcome, providing a better and more realistic support to the nuclear engineer decision. Flexible and adaptable structure with complex physical systems, reasonable calculation time, simplicity in implementing, and ability to find good solutions near as comparable with global minimum are the most privileges of this method compare with former works.

A main problem in a research reactor design optimization is fuel plate safety design optimization (Wahed and Ibrahim, 2010; Wahed et al., 2008). Optimization of fuel plate safety design while taking into account major goals of conceptual design such as improving reliability and life-time – which are the most significant elements during shutdown – is a major multi-objective optimization problem. Due to hugeness of search space in fuel plate optimization problem, finding optimum solution in classical methods requires a huge amount of calculation and CPU time. In a fuel plate optimization task, not only mass of fuel plate but also deformation of fuel plate should be kept minimum which make it classified as a multi-objective problem with an extremely huge search space. In this paper we develop a CA–QSA based optimization and apply it to a multipurpose research reactor (MPRR) fuel plate design for a safety shut down using MATLAB.

Following the introduction, Section 2 presents concept of CA briefly. Section 3 describes SA. In Section 4, quasi-simulated annealing is introduced. Sections 5 and 6 are devoted to multi-objective optimization problem and optimization algorithm by combining CA and SA respectively. Discussion about implementation of optimization algorithm in fuel plate optimization problem is provided in Section 7. A discussion about simulation results is presented in Section 8. Finally, Section 9 summarizes the conclusion remarks.

2. Cellular automata

2.1. A brief history

The notion of cellular automata has a long, living history going back to Ulam (1952) and Neumann et al. (1966). Von Neumann's cellular automaton (in plural cellular automata) represents a direct predecessor of cellular automata (CAs). Shortly, von Neumann introduced CA as a computational medium for machine self-replication motivated by a simple question: "What kind of logical organization is sufficient for an automaton to be able to reproduce itself?". In other words "Can we reproduce computationally and in silico what living cells do?".

Much of von Neumann's work was completed and extended by Burks (1970). A burst of CA activity occurred in the 1970s with the introduction of John Conway's game of "life" (Gardner, 1970). Life was motivated as a simple model of an ecology containing 24 cells, which live and die according to a few simple rules. This most familiar example of a CA displays rich patterns of activity and is capable of supporting many intricate structures.

2.2. The structure

CA can be described in several ways. The description, which is perhaps most useful for physics, is to think of a CA as an entirely discrete version of a physical field. Space, time, field variables, and even the dynamical laws can be completely formulated in terms of operations on a finite set of symbols. The points (or cells) of the space consist of the vertices of a regular, finite-dimensional lattice, which may extend to infinity, though in practice, periodic boundary conditions are often assumed (Wolfram, 1986). Time progresses in finite steps and is the same at all points in space. Each point has dynamical state variables, which range over a finite number of values. The time evolution of each variable is governed by a local, deterministic dynamical law (usually called a rule): the value of a cell at the next time step depends on the current state of a finite number of "nearby" cells called the neighborhood. Finally, the rule acts on all points simultaneously in parallel and is the same throughout space for all times (Smith, 1994). Figs. 1 and 2 show 1d and 2d examples of CA, respectively.

2.3. Neighborhood concept

A cellular automata rule is local, by definition. The updating of a given cell requires one to know only the state of the cells in its vicinity. The spatial region in which a cell needs to search is called the *neighborhood*. In principle, there is no restriction on the size of the neighborhood, except that it is the same for all cells. However, in practice, it is often made up of adjacent cells only.

For two-dimensional cellular automata, two neighborhoods are often considered: the von Neumann neighborhood, which consists of a central cell (the one which is to be updated) and its four geographical neighbors north, west, south and east. The Moore neighborhood contains, in addition, second nearest neighbors north-east, north-west, south-east and south-west, that is a total of nine cells. Fig. 3 illustrates these two standard neighborhoods.

2.4. Boundary conditions

In practice, when simulating a given cellular automata rule, one cannot deal with an infinite lattice. The system must be finite and have boundaries. Clearly, a site belonging to the lattice boundary does not have the same neighborhood as other internal sites. In order to define the behavior of these sites, the neighborhoods for the sites at the boundary are extended. For instance, a very common solution is to assume periodic (or cyclic) boundary conditions, that is one supposes that the lattice is embedded in a torus-like topology. In the case of a two-dimensional lattice, this means that the left and right sides are connected, and so are the upper and lower sides. Other possible types of boundary conditions are illustrated in Fig. 4.

2.5. Deterministic approach

The concept of CA begins from the concept of space and the locality of influence. We assume that the system we would like to represent is distributed in space, and that nearby regions of space have more to do with each other than regions far apart. The idea that regions nearby have greater influence upon each other is often associated with a limit (such as the speed of light) to how fast information about what is happening in one place can move to another place. Once we have a system spread out in space, we mark off the space into cells. We then use a set of