Annals of Nuclear Energy 75 (2015) 64–71

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

Pressurized water reactor in-core nuclear fuel management by tabu search

Natasha J. Hill, Geoffrey T. Parks *

Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom

A R T I C L E   I N F O

Article history:
Received 6 April 2014
Received in revised form 30 July 2014
Accepted 31 July 2014

Keywords:
Tabu search
Reload core design
Pressurized water reactor

A B S T R A C T

Optimization of the arrangement of fuel assemblies and burnable poisons when reloading pressurized water reactors has, in the past, been performed with many different algorithms in an attempt to make reactors more economic and fuel efficient. The use of the tabu search algorithm in tackling reload core design problems is investigated further here after limited, but promising, previous investigations. The performance of the tabu search implementation developed was compared with established genetic algorithm and simulated annealing optimization routines. Tabu search outperformed these existing programs for a number of different objective functions on two different representative core geometries.

1. Introduction

The design of pressurized water reactors (PWR) reload cores is a formidable combinatorial optimization problem. The designer's task is to find the configuration of fresh and partially burnt fuel and burnable poisons (BPs) that optimizes the performance of the reactor over the subsequent cycle, while ensuring that various operational constraints are satisfied. Such problems have a number of different possible objectives, constraints and many local optima (Galperin, 1995).

Over the years this problem has been tackled in many different ways. Naft and Sesonske (1972) sought to minimize the ratio of peak-to-average power by direct search using heuristic shuffling rules. Federowicz and Stover (1973) also tried to minimize power peaking by successive application of integer linear programming. Ahn and Levine (1985) used a gradient projection method and linear programming in a series of stepwise optimization calculations to minimize the cost of the reload core. Hobson and Turinsky (1986) coupled a first-order accurate perturbation theory model to a Monte Carlo integer programming algorithm to search for loading patterns (LPs) that maximized the energy production over a cycle, subject to constraints on power peaking and fuel burn-up. Kim et al. (1987) developed a two-stage procedure for maximizing cycle length, subject to power peaking constraints, by decoupling the fuel and BP placement problems. Stillman et al. (1989) used the backward diffusion calculation (Chao et al., 1986) and successive linear programming to determine theoretically optimal fuel and two-dimensional (2D) power distributions for a PWR, minimizing fissile material and BP inventories. Kropaczek and Turinsky (1991) combined the simulated annealing (SA) stochastic optimization technique with a core physics model based on second-order accurate generalized perturbation theory (GPT) to find near-optimal LPs for a variety of different objectives and constraints.

Since the pioneering work of Kropaczek and Turinsky (1991), other researchers, including Mahlers (1994), Šmuc et al. (1994) and Stevens et al. (1995), have developed SA variants to optimize PWR LPs or applied other stochastic/heuristic optimization methods to this problem and/or the closely related boiling water reactor (BWR) LP optimization problem. These other methods have included: genetic algorithms (GAs) (Poon and Parks, 1993; DeChaine and Feltus, 1995; Chapot et al., 1999; François and López, 1999; Ortiz and Requena, 2004; Martín-del-Campo et al., 2004); estimation of distribution algorithms (Jiang et al., 2006); ant colony optimization (De Lima et al., 2008; Esquivel-Estrada et al., 2011; Wang and Lin, 2009; Lin and Lin, 2012); particle swarm optimization (Alvarenga de Moura et al., 2009; Khoshahval et al., 2010; Liu and Cai, 2012); and harmony search (Poursalehi et al., 2013).

A couple of studies have previously investigated the performance of tabu search (TS) on PWR reload core design problems (Lin et al., 1998; Ben Hmaida et al., 1999). These both considered the problem of minimizing the power peaking factor, identifying small improvements in performance compared to a GA implementation. TS implementations have also been applied to various BWR applications: fuel lattice design (François et al., 2003), reload core design (Castillo et al., 2004), control rod design (Castillo et al.,

* Corresponding author. Tel.: +44 1223 332600; fax: +44 1223 332662.
E-mail address: gtp10@cam.ac.uk (G.T. Parks).

http://dx.doi.org/10.1016/j.anucene.2014.07.051
0306-4549/

2005) and a combination of fuel loading and control rod pattern optimization (Castillo et al., 2007).

This paper investigates the performance of a TS implementation on representative PWR reload core design problems, seeking optimal values for the parameters that control the algorithm for a range of different objective functions, and then comparing the performance of the resulting TS implementation with that of established SA and GA implementations.

2. Tabu search

Originally developed by (Glover and McMillan, 1986), TS is a meta-heuristic algorithm based on local (or neighborhood) search which has found wide application (Glover and Laguna, 1997), particularly for combinatorial optimization problems. Meta-heuristic algorithms iteratively try to improve the solution but cannot guarantee that the optimum is ever found.

TS evaluates a set of solutions which are, by some definition, next to the current solution and moves to the best of these solutions, even if the objective function value deteriorates as a result of the move. A short-term memory (or *tabu list*) is used to store the most recently visited solutions, and these are not allowed to be revisited for a number of iterations equal to the *tabu tenure*. This feature allows the search to escape from local optima.

Intensification and *diversification* are two further strategies employed in many TS implementations when the progress of the search slows. These rely on medium- and long-term memories. The medium-term memory (MTM) stores a selection of the best solutions visited in the search. The long-term memory (LTM) records information on how frequently different regions of the search space have been visited.

The aim of *intensification* is to more thoroughly explore the search space close to the locations of the best solutions found. When *intensification* is performed, the search is returned to a solution determined by those in the MTM and search parameters can be adjusted.

Diversification aims to visit insufficiently explored regions of the search space. A random solution in an infrequently visited region (identified using the LTM) is selected and the search is restarted from there. A rudimentary *diversification* strategy does not use a LTM and instead just restarts from random locations in the search space.

3. PWR reload core design

A typical PWR core contains 193 fuel assemblies arranged with quarter-core (reflective or rotational) symmetry. At each refueling between one third and one quarter of these may be replaced. It is common practice for fresh fuel assemblies to carry BPs. It is also usual to rearrange old fuel in order to improve the characteristics of the new core. This shuffling can entail the exchange of corresponding assemblies between core quadrants, which is equivalent to changing the assembly 'orientations', or the exchange of different assemblies, which changes their locations and possibly their orientations also. Examples of each exchange are shown in Fig. 1.

Thus, a candidate LP of predetermined symmetry must specify:

- the fuel assembly to be loaded in each core location,
- the BP loading with each fresh fuel assembly, and
- the orientation of each burnt assembly.

One interesting point to emerge from a review of past work on PWR reload core design is the diversity in objective functions chosen for optimization. These have included:

1. Maximization of end-of-cycle (EOC) reactivity.
2. Maximization of discharge burn-up.
3. Minimization of feed enrichment.
4. Minimization of power peaking.
5. Minimization of the fresh fuel inventory.

An effective LP optimization method should ideally work well for any objective function chosen by the user, rather than enforce a choice of objective function on the user.

4. Algorithm implementation

4.1. Framework

The TS implementation was developed within the optimization framework provided in the Fuel Optimization for Reloads: Multiple Objectives by Simulated Annealing for PWRs (FORMOSA-P) nuclear fuel management optimization code (Kropaczek and Turinsky, 1991; Kropaczek et al., 1994; Maldonado et al., 1995). The original version of FORMOSA-P combined SA-based optimization with a 2D (radial) nodal expansion method simulator coupled with an assembly power response GPT LP evaluator. A GA implementation was subsequently added to FORMOSA-P (Poon and Parks, 1993; Parks, 1996).

Within FORMOSA-P each candidate LP is represented by three 2D arrays, corresponding to the physical layout of the fuel assemblies (with identifiers indicating unique fresh or burnt fuel designs), their BP loadings (with identifiers indicating individual options from the range available) and their orientations, respectively, as shown in Fig. 2.

For combinatorial optimization problems such as PWR reload core design, application-specific crossover operators are required in GA implementations to guarantee that valid offspring are produced; in this case, to ensure that the fuel assembly inventory is maintained. The FORMOSA-P GA implementation uses Poon and Parks' heuristic tie-breaking crossover (HTBX) operator (Poon and Parks, 1993). The HTBX maps the parent fuel assembly arrays to reactivity-ranked arrays based on the assemblies' beginning-of-cycle (BOC) reactivities. It then combines randomly selected complementary parts of these arrays through a 'cut and paste' operation and uses a simple tie-breaking algorithm to produce valid offspring reactivity-ranked arrays. Finally, the assembly-ranking mapping is reversed to produce the offspring assembly LPs. The BP loadings and assembly orientations are all inherited from one or other parent. Thus, the BOC reactivity distribution of an offspring LP resembles, but is not necessarily identical to, parts of both parents. The performance comparisons presented in Sections 6.2 and 6.3 are, of course, specific to this GA implementation.

The mutation operator from the FORMOSA-P GA implementation is used extensively in our TS implementation. The mutation operator performs a binary exchange of fuel assemblies and randomly changes the BP loading and orientation of the two fuel assemblies from within the ranges of values for these parameters allowed by the specified core symmetry and geometry and fuel and BP inventories and options.

The objective functions and constraints are handled in the same way as in FORMOSA-P and the reactor core analysis is also performed using GPT (Kropaczek et al., 1994; Maldonado et al., 1995). Four objective functions are available:

1. Maximization of the EOC soluble boron concentration (equivalent to maximizing the EOC reactivity).
2. Minimization of the radial power peaking.
3. Maximization of the discharge burn-up.
4. Minimization of the enrichment of fresh fuel.