Progress in Nuclear Energy 76 (2014) 112–121

Contents lists available at ScienceDirect

Progress in Nuclear Energy

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

A new hybrid method for multi-objective fuel management optimization using parallel PSO-SA

F. Khoshahval*, A. Zolfaghari, H. Minuchehr, M.R. Abbasi

Engineering Department, Shahid Beheshti University, G.C, P.O. Box: 19839-63113, Tehran, Iran

ARTICLE INFO

Article history:
Received 13 February 2014
Received in revised form
25 April 2014
Accepted 17 May 2014
Available online 11 June 2014

Keywords:
Fuel management
Optimization
PSO
SA

ABSTRACT

In this paper, we developed a new parallel optimization algorithm, P-PSOSA, for performing the fuel management optimization; we define two different fitness function considering the multiplication factor maximizing and power peaking factor minimizing objectives simultaneously. For this purpose, we developed a FORTRAN program in order to gain the possible minimum fitness value for the loading pattern optimization operation. The P-PSOSA has been performed for KWU reactor. Numerical results of P-PSOSA confirm that the proposed algorithm has a great strength to obtain a near global core pattern as respect to considered objective functions during suitable consuming runtime.

1. Introduction

The problem of loading pattern optimization (LPO) in pressurized water reactors is complex as it represents a combinatorial optimization problem. The aim is to place N fuel assemblies in N channels in the reactor core in order to be able to extract the maximum energy and satisfy several safety constraints. In this investigation, we use a multi-objective fitness function with two main goals in the nuclear core pattern design i.e. maximizing the core multiplication factor (K_{eff}) in order to extract the maximum cycle energy and flattening the power distribution due to safety constraints.

Compared with single-objective (SO) optimization problems, which have a unique solution, the solution to multi-objective (MO) problems consists of sets of tradeoffs between objectives. The goal of multi-objective optimization (MOO) algorithms is to generate these trade-offs. Exploring all these trade-offs is particularly important because it provides the system designer/operator with the ability to understand and weigh the different choices available to them. The optimization solution results in a single value that reflects a compromise between all objectives.

Several metaheuristics or computational intelligence approaches are applied to LPO problem, 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), Simulated Annealing (Smuc et al., 1994; Stevens, 1995), Ant Colony algorithm (Machado and Schirru, 2002), Genetic algorithms (Yamamuto, 1997), Particle Swarm Optimization (Meneses et al., 2009; Khoshahval et al., 2010), Improved pivot particle swarm method (Liu and Cai, 2012), Cellular Automata (Fadaei and Setayeshi, 2009), and Artificial Intelligence techniques like fuzzy logic and Artificial Neural networks (Kim et al., 1993; Sadighi et al., 2002; Erdog and Geçkinli, 2003).

Although most of the above mentioned studies have used single objective as fitness function in their optimization methods. In this paper, the parallel particle swarm optimization along with simulated annealing is applied to in-core fuel management optimization.

This proposed algorithm has been implemented successfully for the multi-objective optimization of fuel loading pattern design based on flattening radial power distribution and maximizing effective multiplication factor. In order to evaluate and demonstrate the performance of the method for the fuel management optimization of nuclear reactor core, we provided a parallel PSO-SA optimization program. In addition, for treatment of transport equation, we developed a neutronic module in which it solves the two dimensional, multi group transport equations using variational treatment of even parity transport equation. This code has been validated against various benchmarks, FEMPT (Abbasi et al., 2011).

* Corresponding author. Fax: +98 21 29902546.
E-mail addresses: f_khoshahval@sbu.ac.ir, f_khoshahval@yahoo.com (F. Khoshahval).

http://dx.doi.org/10.1016/j.pnucene.2014.05.014
0149-1970/

To reduce the computation time required by FEMPT, we used parallel computing. Motivation to perform parallel computing is rapidly increasing because of the evolution of computer hardware and networks. However, the parallel execution of current PSO is somewhat difficult because it utilizes an inherently sequential algorithm (Axmann, 1997; Lee et al., 1999).

2. Particle swarm intelligence

Particle swarm optimization (PSO) is one of the most popular optimization frameworks based on the original notion of swarm intelligence. PSO is a population-based stochastic approach for solving continuous and discrete optimization problems. PSO was first introduced by Kennedy and Eberhart in 1995, which is inspired by social behavior of bird flocking and fish schooling. In PSO, a set of randomly generated solutions propagates in the design space towards the optimal solution over a number of iterations based on large amount of information about the design space that is assimilated and shared by all members of the swarm. In PSO, each particle moves in the search space with a velocity according to its own previous best solution and their group previous best solution. The position of a particle represents a candidate solution to treat the optimization considered problem. The following equations are used to iteratively modify the particle velocities and positions at each time step, i.e:

$$v_{\mathrm{id}}^{t+1} = w^t v_{\mathrm{id}}^t + c_1 r_1^t (\mathrm{pbest}_{\mathrm{id}} - x_{\mathrm{id}}^t) + c_2 r_2^t (\mathrm{gbest}_d - x_{\mathrm{id}}^t) \quad (1)$$

$$x_{\mathrm{id}}^{t+1} = x_{\mathrm{id}}^t + v_{\mathrm{id}}^{t+1} \quad (2)$$

where

$i = [1,2,...,n]$
$d = [1,2, \ldots,\mathrm{m}]$
n = number of particles in a group
m = elements of particle vectors

and

v_{id}^t = Velocity of the particle at time step t.
v_{id}^t = Position at time step t.
c_1, c_2 = Acceleration constants
r_1, r_2 = Random number between 0,1.
w^t = Inertia weight at time step t.
$\mathrm{Pbest}_{\mathrm{id}}$ = Previous best position of the particle at time step t. (sometimes called best neighbor)
gbest_d = Best position among all particles at time step t.
t = Current iteration.

It can be observed the new particle position is obtained by adding the particle's current position and the new velocity using Eq. (1). One of the drawbacks of this method is parameter dependency of the method. The PSO parameters are very important since they have significant impact on optimization results. The number of particles and maximum iteration number are selected 20 and 100 respectively as suggested by Khoshahval et al., 2010. In this investigation, it is found that, taking $c_1 = c_2 = 2$ are proper choice. The inertia weight could be picked out as a constant value but it is preferred to define a variable w which decreases linearly during a run, i.e.:

$$w^t = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{t_{\max}}\right) t \quad (3)$$

where $w_{\max}$ is initial weight, $w_{\min}$ is final weight, $t_{\max}$ is maximum iteration (generation) number and t is current iteration number. According to Khoshahval et al. (2010), we used $w_{\max} = 0.8$, $w_{\min} = 0.4$ and $t_{\max} = 100$. It should be said, finding the best value for the parameters in PSO is not an easy task and it may differ from one problem to another.

3. Simulated annealing

As we know a lump of material at a high temperature has a higher energy state than an identical lump at a lower temperature. To reduce the energy of the material to the lowest possible value, we should do this gradually, because rapid cooling result in many imperfections in the crystal structure, or in a substance that is glasslike, with no crystalline structure at all (Freeman and Skapura, 1991).

Simulated annealing was originally proposed by Metropolis in the early 1950s as a model of the crystallization process. It was only in the 1980s, however, that independent research done by Kirkpatrick et al., 1983 noted the similarities between the physical process of annealing and some combinatorial optimization problems. They noted that there is a correspondence between the alternative physical states of the matter and the solution space of an optimization problem. It was also observed that the objective function of an optimization problem corresponds with the free energy of the material. An optimal solution is associated with a perfect crystal, whereas a crystal with defect corresponds with a local optimal solution (Van Laarhoven and Aarts, 1987).

In an optimization problem, often the solution space has many local minima. A simple local search algorithm proceeds by choosing random initial solution and generating a neighbor from that solution. If it is a minimum fitness transition then the neighboring solution is accepted. Such an algorithm has the drawback of often converging to a local minimum. The simulated annealing algorithm avoids getting trapped in a local minimum by accepting cost increasing neighbors with some probability. In SA method we begin at an initial temperature (T_0) and reduce it gradually. In the other words this process begins with a random guess of the cost (fitness) function variable values. Heating means randomly modifying the variable values. Higher heat implies greater random fluctuations. The cost function returns the output, f, associated with a set of variables. If the output decreases, then the new variable set replaces the old variable set. If the output increases, then the output is accepted by a probability which has Boltzmann distribution:

$$P(\Delta E) = \exp(-\Delta E/KT) \quad (4)$$

where k is Boltzmann constant, ΔE is change in the energy or fitness function (See Section 8.2) of the system (reactor core) and T is the temperature. If $P(\Delta E) >$ rand the new state is acceptable, (rand is a uniform random number between 0 and 1), otherwise is rejected. Thus, even if a variable set leads to a worse cost, it can be accepted with a certain probability. As the temperature reduces more and more, the probability of accepting $\Delta E > 0$ is reduced and finally the solutions approaches to the near global minimum.

4. PSO-SA hybrid optimization method

The proposed system combines PSO and SA for global minimization. The drawback of PSO is that the swarm may prematurely converge. The underlying principle behind this problem is that, for the global best PSO, particles converge to a single point which is on the line between the global best and the personal best positions (Premalatha and Natarajan, 2010). This point is not guaranteed to be even a local optimum (Van den Bergh and Engelbrecht, 2004). A

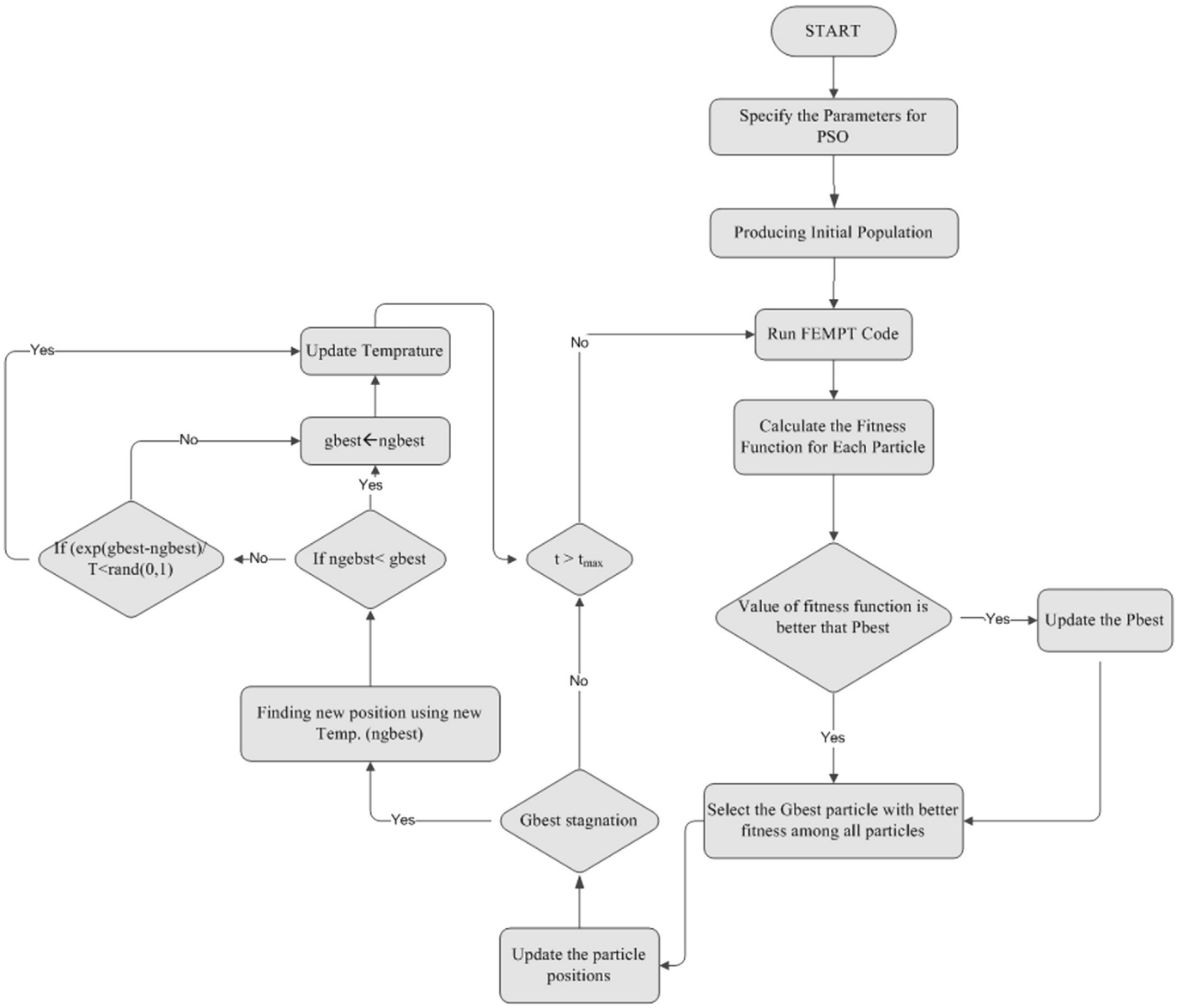

Fig. 1. Algorithm of PSO-SA hybrid optimization method.

further drawback is that stochastic approaches have problem-dependent performance. This dependency usually results from the parameter settings in each algorithm. The different parameter settings for a stochastic search algorithm result in high performance variances. In general, no single parameter setting exists which can be applied to all problems. Changing the parameter value will increase the speed of the particles resulting in more exploration (global search) and less exploitation (local search) or on the other hand, more exploitation and less exploration (Premalatha and Natarajan, 2010). Thus finding the best value for the parameter is not an easy task and it may differ from one problem to another. Therefore, from the above it can be concluded that the PSO performance is problem-dependent. The problem dependent performance can be addressed through hybrid mechanism. Hybrid refers to combining different approaches to benefit from the advantages of each approach.

Simulated Annealing (SA) is locating a good approximation to the global minimum of a given function in a large search space. Each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter T called the temperature that is gradually decreased during the process.

The applied algorithm incorporates the SA in PSO when the gbest particle stagnates, the number of iterations taken for stagnation checking is 5.

As can be seen in Fig. 1, the algorithm for PSO-SA when gbest particle stagnates can be summarized as below:

1. Initialize the population positions and velocities
2. Initialize temperature
3. Evaluate the fitness of the individual particle using FEMPT code
4. Finding best previous particles (Pbest)
5. Finding global best individuals (gbest)
6. Modify velocities and particles position based on pbest and gbest position
7. If gbest position is not changed over a period of time

7-1 Find a new position using temperature
7-2 Accept the new position as gbest position with probability $\exp(-\Delta E/T)$ even though current position is worse
8. Update temperature (reduce T)
9. Terminate if the condition is met ($t > t_{max}$)
10. Go to step 3

And finally the solution approaches the global minimum.

5. Neutronic calculations

The Boltzmann transport equation in its various integral-differential forms can be used to solve problems arising in reactor physics. For realistic problems of practical interest, one has to resort to numerical models and computational techniques (Ackroyd, 1997).

Apart from their geometrical flexibility, the finite element methods give the angular flux everywhere in the system. In addition, the error bounds on some of the local and global characteristics of the system can be predicated in a systematic manner. TWODOG (Splawski, 1981), FELTRAN (Issa et al., 1986), MARC (Fletcher, 1986), EVENT (De Oliveira, 1986) and FEMPT (Abbasi et al., 2011) are some of the examples of the computer codes used to solve transport problems using finite element methods. These codes are capable of solving multi group transport problems for steady state or time dependent, isotropic or anisotropic scattering problems in multi-dimensions.

In this work, for the generation of macroscopic group constants, the WIMSD5 code has been employed (Winfrith, 1985). In addition, FEMPT code (Abbasi et al., 2011) is used to calculate effective multiplication factor and assembly-wise normalized power throughout the core.

5.1. Even-parity neutron transport equation

Finite element methods for multi-group problems of neutron transport are based on finite element schemes for Boltzmann equation for an arbitrary energy group. By performing a neutron balance within a small volume element dV and representing the rate of change of number of neutrons as the difference of their production and loss, the neutron transport equation is obtained. If $\varphi_0(r,\Omega)$ is the group angular flux, at the point r and for direction Ω at r, then the Boltzmann equation is

$$\vec{\Omega}\cdot\vec{\nabla}\varphi_0(r,\Omega)+\sigma_t(r)\phi_0(r,\Omega)=\int \mathrm{d}\Omega'\sigma_s(r,\Omega\cdot\Omega')\varphi_0(r,\Omega')+S(r,\Omega) \tag{5}$$

The distributed extraneous source of neutron is $S(r,\Omega)$, the total cross section is $\sigma_t(r)$, and $\sigma_s(r,\Omega\cdot\Omega')$ is the directional scattering cross-section at r for scattering from the direction Ω' into the direction Ω.

In even-parity method angular flux, scattering cross sections and sources divided to even and odd parity (Ackroyd et al., 1987). It is well known (Davis, 1968) that Eq. (5) can be cast into the pair of parity equations

$$\vec{\Omega}\cdot\vec{\nabla}\varphi_0^-(r,\Omega)+C\varphi_0^+(r,\Omega)=S^+(r,\Omega) \tag{6}$$

$$\vec{\Omega}\cdot\vec{\nabla}\varphi_0^+(r,\Omega)+G^{-1}\varphi_0^-(r,\Omega)=S^-(r,\Omega) \tag{7}$$

The even and odd parity variables are defined as

$$\varphi_0^{\pm}(r,\Omega)=\frac{1}{2}[\varphi_0(r,\Omega)\pm\phi_0(r,-\Omega)] \tag{8}$$

$$S^{\pm}(r,\Omega)=\frac{1}{2}[S(r,\Omega)\pm S(r,-\Omega)] \tag{9}$$

where C and $G-1$ operators for arbitrary function u (r,Ω) are defined as

$$\begin{aligned} Cu^+(r,\Omega)&=\sum_{n=\text{even}}^{\infty}\frac{(2n+1)}{4\pi}\sigma_n(r)\int P_n(\mu_0)u^+(r,\Omega')\mathrm{d}\Omega' \\ G^{-1}u^-(r,\Omega)&=\sum_{n=\text{odd}}^{\infty}\frac{(2n+1)}{4\pi}\sigma_n(r)\int P_n(\mu_0)u^-(r,\Omega')\mathrm{d}\Omega' \end{aligned} \tag{10}$$

5.2. Boundary conditions for parity equations

By manipulating the above equation, at the vacuum surface, S_b, the boundary condition for even parity equation becomes (Ackroyd, 1978)

$$\varphi_0^+(r,\Omega)+G\left[S^-(r,\Omega)-\vec{\Omega}\cdot\vec{\nabla}\varphi_0^+(r,\Omega)\right]=0 \quad \text{for } \vec{\Omega}\cdot\vec{n}<0 \tag{11}$$

For odd-parity equation the vacuum surface boundary condition is

$$\varphi_0^-(r,\Omega)+C^{-1}\left[S^+(r,\Omega)-\vec{\Omega}\cdot\vec{\nabla}\varphi_0^-(r,\Omega)\right]=0 \quad \text{for } \vec{\Omega}\cdot\vec{n}<0 \tag{12}$$

For a perfect reflecting surface we have

$$\varphi_0^+(r,\Omega)=\varphi_0^+(r,\Omega^*) \quad \text{for } \vec{\Omega}\cdot\vec{n}=-\vec{\Omega}\cdot\vec{n}\neq 0 \tag{13}$$

The even-parity boundary condition for a prescribed surface source $T(r,\Omega)$ is

$$\varphi_0^+(r,\Omega)+G\left[S^-(r,\Omega)-\vec{\Omega}\cdot\vec{\nabla}\varphi_0^+(r,\Omega)\right]=T(r,\Omega) \quad \text{for } \vec{\Omega}\cdot\vec{n}<0 \tag{14}$$

The same boundary conditions in the direction $-\Omega$ and for odd parity equation can be developed.

5.3. Variational treatment of even parity transport equation

As the works of Ackroyd (1997) demonstrate, indefinitely many variational principles can be associated to the even-parity problem. The generalized least square method has been used by Ackroyd and Nanneh (1990) to establish for steady state transport leads to $K^+(\varphi^+)$ principle as follow

$$\begin{aligned} K^+\left(\varphi^+\right)=&\int_v\int_{4\pi}\left\{2\varphi^+S^+ + 2\vec{\Omega}\cdot\vec{\nabla}\varphi^+GS^- - \vec{\Omega}\cdot\vec{\nabla}\varphi^+G\vec{\Omega}\cdot\vec{\nabla}\varphi^+ \right. \\ &\left. -\varphi^+C\varphi^+\right\}\mathrm{d}\Omega\mathrm{d}V + 4\int_{\partial v}\int_{\vec{\Omega}\cdot\vec{n}<0}|\vec{\Omega}\cdot\vec{n}|T\varphi^+\mathrm{d}\Omega\mathrm{d}S \\ &-\int_{\partial v}\int_{4\pi}|\vec{\Omega}\cdot\vec{n}|\varphi^{+2}\mathrm{d}\Omega\mathrm{d}S = 2F_s\left[\varphi^+\right] \\ &-F\left[\varphi^+,\varphi^+\right] \end{aligned} \tag{15}$$

A feature of classical variational principles is that admissible trial functions have to satisfy a continuity condition at the interfaces of regions, such as finite elements, so that the same kind of cohesion holds between the regions of the system for trial functions as imposed for the exact solution.

Eq. (15) could be written for the local element (j) in the form of below

$$K^{+}\left(\varphi^{+}\right)=\sum_{j=1}^{E}K^{+j}\left(\varphi^{+}\right)=\sum_{j=1}^{E}\left\{2F_{s}^{+j}\left[\varphi^{+}\right]-F^{+j}\left[\varphi^{+},\varphi^{+}\right]\right\} \quad (16)$$

By choosing the normalized spherical harmonics as basis functions, which defined by

$$Y_{nm}^{e}(\mu,\omega)=\left[\frac{2n+1}{4\pi}(2-\delta_{m0})\frac{(n-m)!}{(n+m)!}\right]^{1/2}P_{n}^{m}(\mu)\mathrm{Cos}(m\omega)$$
$$Y_{nm}^{o}(\mu,\omega)=\left[\frac{2n+1}{4\pi}(2-\delta_{m0})\frac{(n-m)!}{(n+m)!}\right]^{1/2}P_{n}^{m}(\mu)\mathrm{Sin}(m\omega) \quad (17)$$

The trial function can be introduced as

$$\varphi^{+}(r,\mu,\omega)=\sum_{j=1}^{E}\varphi^{+j}(r,\mu)=\sum_{j=1}^{E}\sum_{e=\mathrm{even}}^{N-1}\sum_{p=1}^{Ne}Y_{nm}^{e}(\mu,\omega)B_{p}^{j}(r)\Psi_{pnm}^{j} \quad (18)$$

where $B_{p}^{j}(r)$ are the spatial part basis shape functions of the local element (j) and node (p), and Ψ_{pnm}^{j} is nodal coefficient and determined by maximizing $K^{+}(\varphi^{+})$.

6. Parallelization of PSO-SA optimization algorithm

The particle swarm optimizations along simulated annealing evolve population of candidate solutions. They take a large amount of computation time. The most time consuming part is the evaluation of "Fitness" corresponding to each individual of the population. This requires full core simulation by FEMPT (Abbasi et al., 2011) calculation (See section 5) from which objective function or "Fitness" can be computed. Thus, these neutronic calculations for different individuals in the population are carried out in parallel. Parallel programming can be implemented in discrete memory, shared memory, and clusters of shared memory architectures. The Message Passing Interface (MPI) provides some functions for process communication and synchronization that can support distributed memory programming (Gropp et al., 2007).

The parallelization is achieved using the MPI library functions (Pacheco, 1997; Message Passing Interface Forum, 2008) as follows. The rank 0 processor is called here as MASTER processor and others are SLAVE processors. The MASTER will generate population containing N individuals. If there are S SLAVE processors, each SLAVE is assigned the job of evaluation fitness of (N/S) individuals by

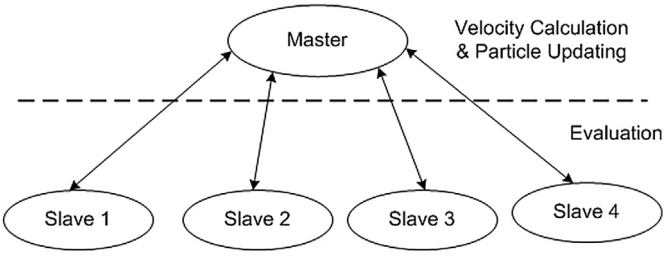

Fig. 2. Parallel PSO-SA hybrid optimization paradigms.

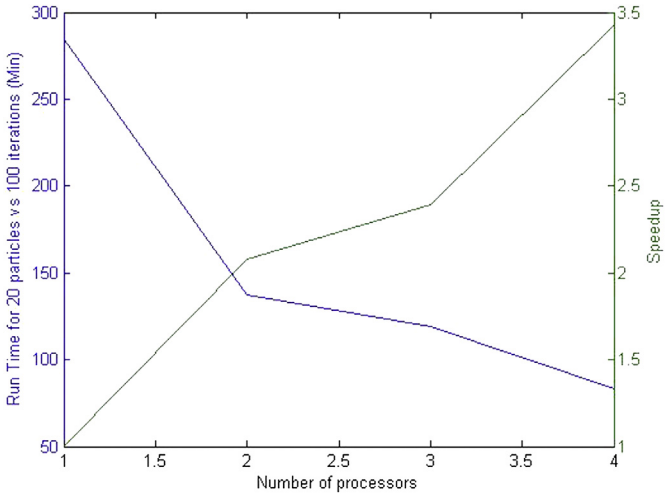

Fig. 3. Execution time and speedup for various numbers of processors.

variational calculations. The results are returned to the MASTER. The MASTER will generate new population and send (N/S) individuals to each slave for fitness evaluation. This is continued for all the generations. Fig. 2 illustrates this approach. The computer code P-PSOSA has been written to execute this using standard message passing interface (MPI) library functions in FORTRAN.

The computation time required on a PC Intel(R) Core™ 2 Quad at 2.5 GHz and 4 GB RAM for different number of processors for 20 particles and 100 iterations is shown in Fig. 3. Computational performance of parallel algorithms is measured by speedup and its efficiency. Parallel speedup is defined to be the ratio of the runtime per iteration in the serial algorithm to the runtime per iteration in the parallel algorithm. Fig. 3 also illustrates the speedups. The parallel efficiency is then computed by dividing the parallel speedup by the number of processors applied to achieve those results. Thus, speedup of P-PSOSA on 4 processors is 3.43 results in 86% efficient.

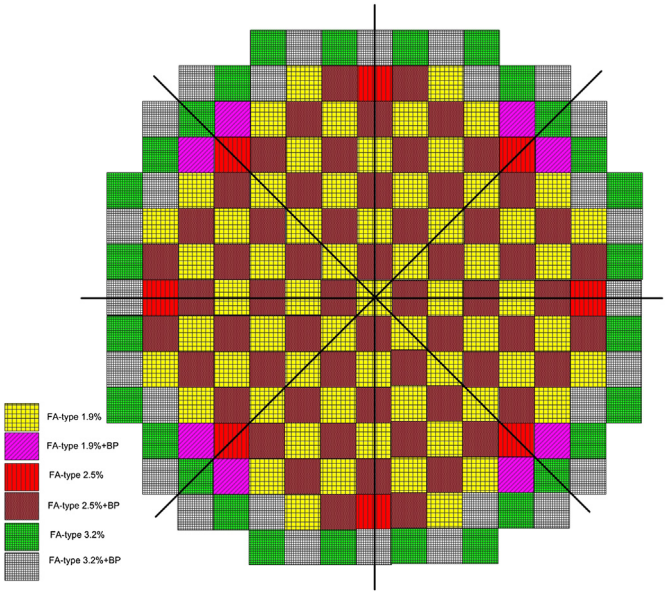

Fig. 4. KWU type reactor core.

Table 1
Design data for the KWU Nuclear Power Plant (Bushehr old design).

Parameters	value
Reactor power (MWe)	1300
Number of fuel assemblies	193
Core active height (cm)	390.0
Equivalent core diameter (cm)	360.5
Enrichment distribution in 1st loading (1.9/2.5/3.2) weight %	69/68/56
Average power density (kW/l)	93
Fuel material	UO_2
Clad material	Zr
Moderator	Water
Fuel pellet diameter (mm)	9.11
Clad outer diameter (mm)	10.75
Fuel rods pitch (mm)	14.35
Clad thickness (mm)	0.72
Pressure in vessel (bar)	158
Inlet temperature ©	291.3
Outlet temperature ©	326.1
Fuel assembly grid	16 × 16
Number of fuel rods per assembly	236
Number of control rod guide thimbles per assembly	20
Burnable poison type	Boron glass tubes
Guide thimble (material/outer dia. (mm)/wall thickness (mm))	S.S./13.72/0.42

7. Description of a test case core

Our test case is the one used by Sadighi et al., 2002. This test case is Bushehr NPP, old KWU design. The KWU reactor has rectangular and 1/8 symmetric shape; it is made of 193 fuel assemblies of the same geometry. Each fuel assembly contains 236 fuel rods and 20 guiding channels for burnable poisons or control rods. The full core model of KWU reactor at the beginning of the cycle (BOC) has been shown in Fig. 4. Table 1 illustrates the design data of the KWU reactor.

Considering the symmetry, the calculations of the reactor core were made using only 1/4 of it and for the optimization process 1/8 of it was considered. This allowed a great simplification for the optimization models and the core simulation, resulting in considerable reduction in the processing time.

8. Implementation

8.1. Mapping of the nuclear reactor reload problem on the PSO-SA

There are 31 fuel assemblies in one-eighth of the reactor core and their specification are given in Table 2. As can be seen from Table 2 and Fig. 4, there are six different fuel assemblies. For Instance, from Table 2, it is found that there are 61 fuel assemblies from type 1 in the core and illustrates the enrichment of type 1 is 1.9. It is worth to note that the positions of the nine fuel assemblies as well as the central one have been fixed respectively at periphery and center of the core. Fixing these assemblies is not unusual, because for achieving the constant power density distribution, fuels

Table 2
Various fuel assembly type, used in PWR type reactor.

Fuel assembly type	Enrichment U235	Number of fuel assembly
FA type 1	1.9	61
FA type 1+BP	1.9	8
FA type 2	2.5	8
FA type 2+BP	2.5	60
FA type 3	3.2	28
FA type 3+BP	3.2	28

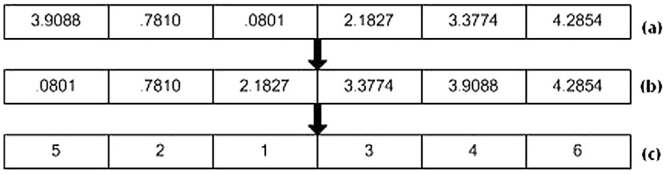

Fig. 5. Decoding a position vector with length equal six (Khoshahval et al., 2010).

with higher enrichments or reactivities should be arranged in peripheral core region. In the other words, it is obvious that in the edge of the core, flux decreases near the reflector, so to prevent flux turning down, it is better to place higher enriched fuels in border of the core. Therefore, one only needs to find position of the rest 22 fuel assemblies in the sector. A straightforward method of loading pattern, LP, representation can be established using a $1 \times d$ ($d = N_1 + N_2 + \ldots + N_n$) position vector,$x_{id}^t$ and fuel type list. The first N_1 positions in the vector indicate the first kind of fuel assemblies, $x_{id}^t(1)$ to $x_{id}^t(N_1)$, and the second N_2 positions specify the second type of fuel assemblies and so forth (Khoshahval et al., 2010).

$$x_{id}^t = \left[\underbrace{vI1, vI2, \ldots, vIN_1}_{\text{Fuel Type } I} , \underbrace{vII1, vII2, \ldots, vIIN_2}_{\text{Fuel Type } II} , \ldots, \underbrace{vm1, vm2, \ldots, vmN_n}_{\text{Fuel Type } m} \right]$$

Each element of the position vector corresponds to the position of a fuel assembly in the core, so it is forbidden each element to take a non-integer or repeated numbers. Real and repeated numbers are decoded to integer and non-repeated values that points to positions of FAs by following approach. In this method, first, the particle position vector is sorted, and then integer and non-repeating values of particles are generated by the location of real number in the sorted vector. This process is depicted in Fig. 5 for a sample particle position vector with six optimized parameters. For instance, since 3.9088 is the first number (Fig. 5a), then the first position in integer and non-repeating position vector (Fig. 5c) is filled by its corresponding position in sorted vector (which is 5 in Fig. 5b). The second number is 0.7810; therefore, in the same way the second position in the position vector is occupied by the corresponding position of 0.7810 in the sorted vector. The process is carried out to fill all position of the vector. This method is used to decode PSO-SA real vectors to integer ones.

Table 3
The P-PSOSA program results for 20 particles along with 100 iteration over 10 experiments with fitness type #1.

Experiment	Fitness value	PPF	K_{eff}
#1	0.79311	1.26610	1.00673
#2	0.78069	1.25972	1.00490
#3	0.77251	1.24979	1.00714
#4	0.81281	1.28022	1.00625
#5	0.76307	1.28234	1.00542
#6	0.75627	1.27245	1.00488
#7	0.79842	1.27464	1.00551
#8	0.76780	1.25822	1.00593
#9	0.81383	1.28429	1.00553
#10	0.80222	1.27799	1.00601
Average	0.78607	1.27058	1.00583
Best	0.75627	1.27245	1.00488
Worst	0.81383	1.28429	1.00553
Std. Dev	0.02085	0.01163	0.00074

Table 4
Comparison of best result attained using Fitness type #1 with main neutronic parameters of KWU reactor.

	Designer	Best Result-Fit
K_{eff}	1.00485	1.00488
PPF_{max}	1.273	1.272
Fitness Function#1	1.22334	0.75627

8.2. Fitness functions

In order to have effective search process in the presence of a constraint, fitness function is needed to be carefully defined (Khoshahval et al., 2011). Moreover, for evaluating different core arrangements in the loading pattern optimization (LPO) problem, we must define a suitable fitness function with respect to considered optimization objectives. In a practical LPO, during the operating cycle, the multiplication factor is usually maximized in order to extract the maximum cycle energy associated with observing constrains such as power peaking factor (PPF) due to safety constraints. We select a multi-objectives fitness function for our optimization problems in order to obtain simultaneously possible maximized K_{eff} along flattened power distribution for a loading pattern in the beginning of cycle (BOC) of our test case. We define a fitness function with the following formulation:

$$FF = \sum_{i=1}^{N} (np_i - 1)^2 + F_p \tag{19}$$

$$F_p = \begin{cases} \dfrac{ppf}{ppf_{\max}} \left(\sum_{i=1}^{N} (np_i - 1)^2 \right) & \text{if } ppf > ppf_{\max} \\ 0 & \text{if } ppf < ppf_{\max} \end{cases} \tag{20}$$

$$\text{Fit\_1} = w_1 FF + w_2 K_{\text{eff}} \tag{21}$$

In which np_i is normalized power of each fuel assembly. N is total number of calculative cells. FF is fitness function w_1 and w_2 are weighting factors which are taken as 0.9 and −0.1 respectively. As can be seen from the Eq. (19) the lower value of FF means flatter power distribution.

In order to reach considered optimization objectives, the possible minimized value of Eq. (21) must be obtained. In the other words, the optimum pattern is the one with highest multiplication factor and flattest power distribution.

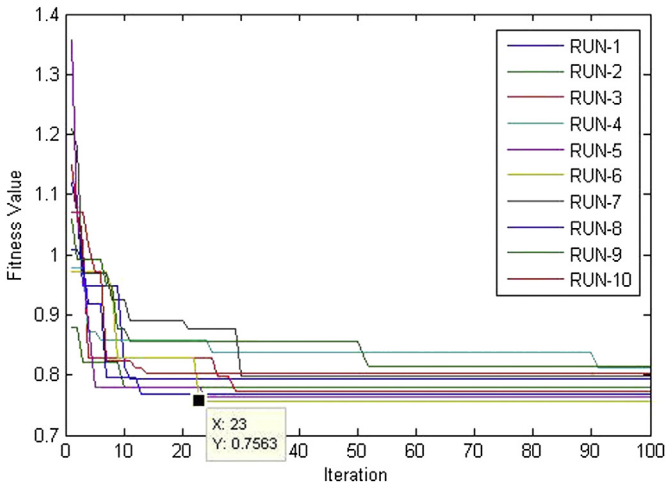

Fig. 6. Convergence of the fitness value versus 100 generation for fitness type #1.

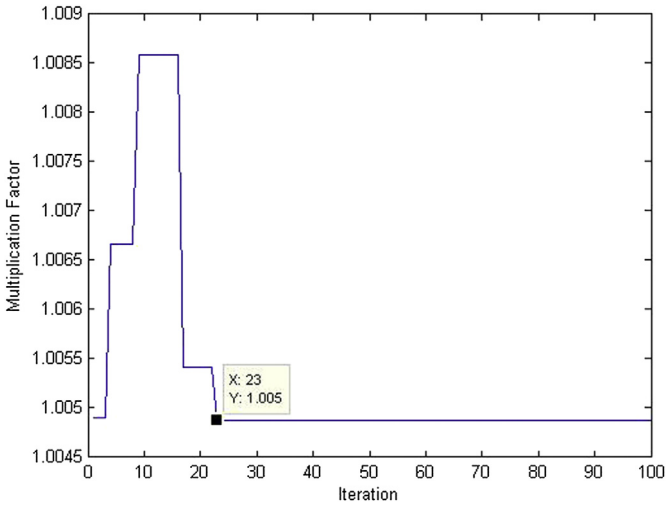

Fig. 7. Multiplication factor value versus iteration for the best result obtained.

In addition, in order to penalize infeasible solutions, a penalty term is used and linearly added to the Eq. (19). PPF is a key parameter and whenever it is greater than ppf_{max} in any FA (this means the power density peaking factor in a position in the core is higher than allowable limit) the accordant loading pattern (arrangement) has lower chance to proceed further in computational procedure (Khoshahval et al., 2010). Therefore, to account for constraint about PPF, the fitness function is augmented by nonnegative penalty term F_p, penalizing constraint violations. The penalty term F_p is proportional to the corresponding violation and zero in case of no violation. In our test case, KWU, the maximum permitted value for PPF is 1.3.

In order to investigate the impact of the fitness function equation on the results of the optimization, we defined another function as follow:

$$\text{Fit\_2} = FF \Big/ K_{\text{eff}} \tag{22}$$

Again optimum pattern is the one with higher multiplication factor and flattest power distribution.

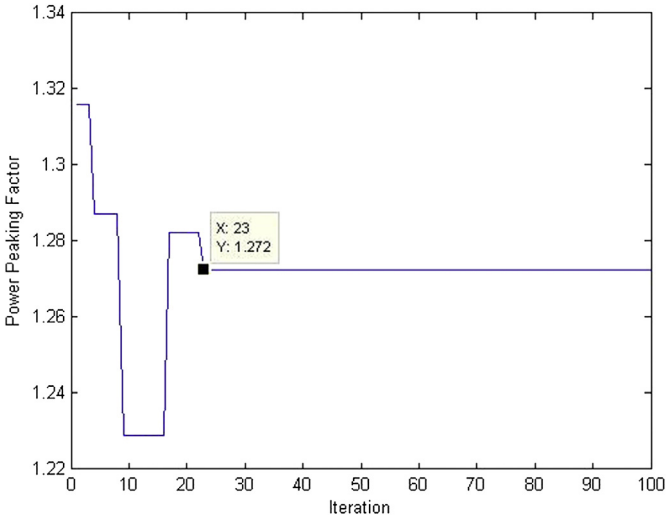

Fig. 8. Power peaking factor value versus iteration for the best result obtained.

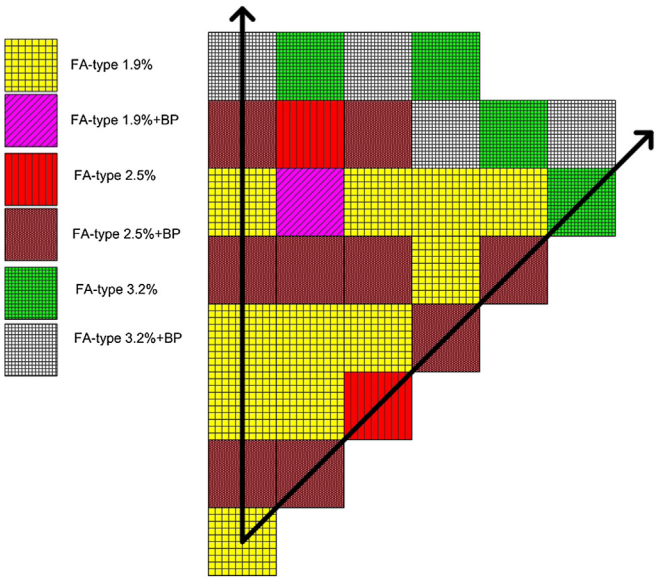

Fig. 9. Optimum arrangement attained by fitness type #1.

Table 5
The P-PSOSA program results for 20 particles along with 100 iteration over 10 experiments with fitness type #2.

Experiment	Fitness value	PPF	K_{eff}
#1	1.00700	1.27761	1.00409
#2	1.00560	1.26533	1.00568
#3	1.00290	1.25479	1.00778
#4	1.03444	1.25774	1.00757
#5	0.98091	1.27505	1.00561
#6	1.00620	1.24078	1.00816
#7	1.00276	1.29183	1.00565
#8	0.96584	1.26007	1.00708
#9	0.91897	1.24503	1.00650
#10	0.96535	1.23664	1.00715
Average	0.98900	1.26049	1.00653
Best	0.91897	1.24503	1.00650
Worst	1.03444	1.25774	1.00757
Std. Dev	0.03233	0.01747	0.00126

9. Optimization results

9.1. Fitness function type #1

The developed program, P-PSOSA, has been executed 10 times with 20 initial particles/populations along with 100 iterations/generations for the KWU reactor. Due to stochastic behavior of the algorithm one needs to run the program several times, experiments, in order to evaluate the method. In order to make the INFM[1] optimization problem real (which we encounter in the industry), the number of assemblies of each type is not conserved in the optimized patterns. However the attained number of assemblies of each type in the optimized patterns is so similar to the designer proposed pattern which also shows the ability of the P-PSOSA method. The best, the average and the worst fitness function are shown in Table 3. As can be seen from Table 3 the best result (best fitness) is 0.75627 with multiplication factor equal to 1.00488 and power peaking factor equal to 1.278. In Table 4 the results of optimized pattern are compared with designer proposed loading pattern (Sadighi et al., 2002). It is noticeable that the proposed method gives us relatively better fitness function, higher multiplication factor and lower power peaking factor compared with designer of KWU. Therefore the proposed method has good reliability for finding the optimum core pattern. Convergence of the fitness value versus 100 generation is depicted in Fig. 6.

The best fitness value versus iteration number is also shown in Fig. 6. One can see from Fig. 6 that the best result has converged after 23 iteration. In addition, we have illustrated the trend of the multiplication factor and power peaking factor change versus iteration for the best result in Fig. 7 and Fig. 8 respectively. Fig. 9 represents the optimal LP with lowest fitness proposed by this investigation using fitness function type #1 for the first cycle of the reactor core.

[1] In core fuel management.

Table 6
Comparison of best result attained using Fitness type #2 with main neutronic parameters of KWU reactor.

	Designer	Best result-Fit2
K_{eff}	1.00485	1.00650
PPF_{max}	1.273	1.245
Fitness Function#2	1.46381	0.91897

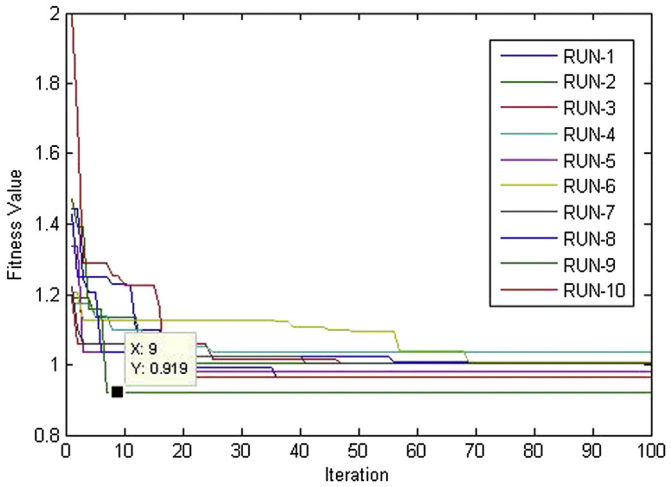

Fig. 10. Convergence of the fitness value versus 100 generation for fitness type #2.

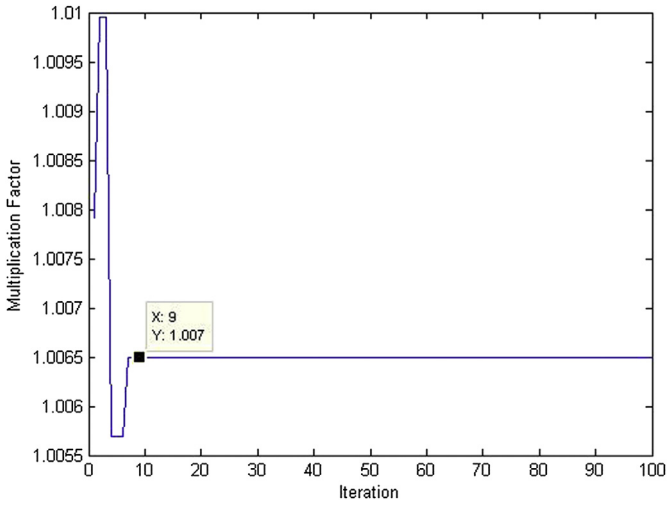

Fig. 11. Multiplication factor value versus iteration for the best result obtained.

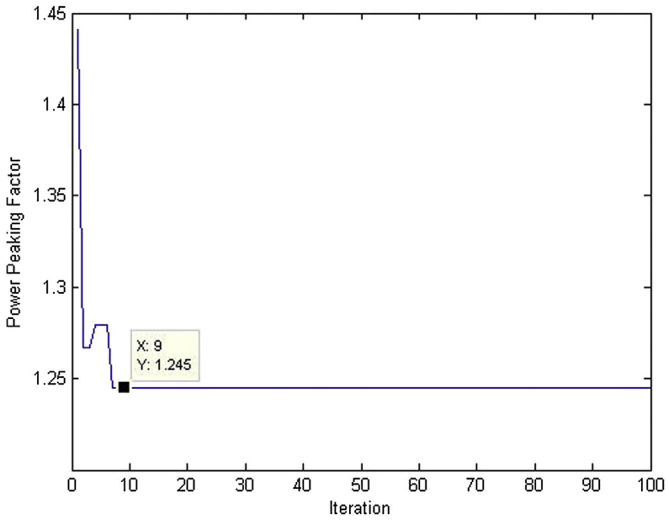

Fig. 12. Power peaking factor value versus iteration for the best result obtained.

9.2. Fitness function type #2

In order to analyze the effect of the fitness function equation on the results of the LP optimization problem, we applied Eq. (22) too. Again, the best, the average and the worst fitness function are shown in Table 5. As can be seen from Table 5 the best fitness is 0.91897 with multiplication factor equal to 1.0065 and power peaking factor equal to 1.245. In Table 6 the results of optimized pattern are compared with designer proposed loading pattern (Sadighi et al., 2002). The Table 6 shows that the fitness function type #2 has improved the results in comparison with designer in the aspect of multiplication factor and power peaking factor. In addition comparing the best result attained by fitness function type #1 and fitness function type #2 show that the second fitness function has given better result (higher multiplication factor along with lower power peaking factor), although it should be noted the standard deviation of the fitness function type #1 is lower than the standard deviation type #2. Thus generally, it can be said that the average results of the fitness function type #1 is better that the one with the fitness function type #2.

Convergence of the fitness value versus 100 iteration (generation) is shown in Fig. 10. One can see from Fig. 10 that the best result has converged at iteration #9.

The multiplication factor and power peaking factor variation versus iteration for the best result has depicted in Fig. 11 and Fig. 12 respectively. Fig. 13 represents the optimal LP with lowest fitness proposed by this paper using fitness function type #2.

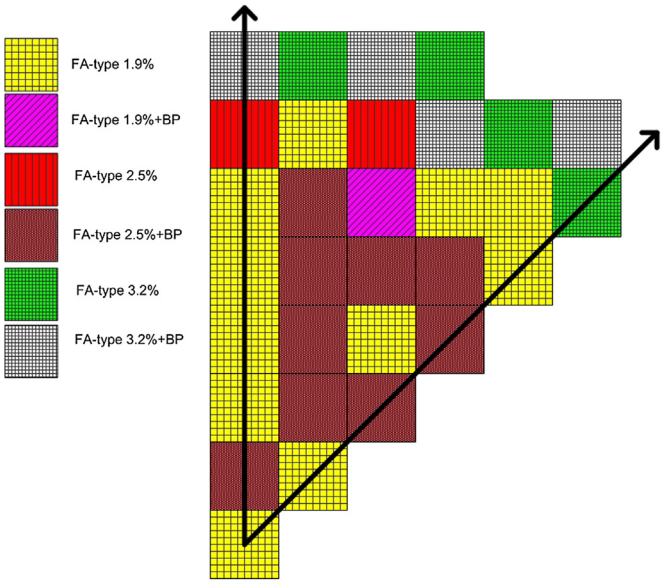

Fig. 13. Optimum arrangement attained by fitness type #2.

10. Conclusion

Test results so far indicate that the parallel PSO-SA algorithm is capable of finding suitably optimized loading pattern for the given PWR design. Moreover, common procedure for core treatment equation is implementation of finite difference form of diffusion equation, P_1, or Nodal approach. In this paper, however, we used variational method of approximating the solution of the even-parity transport equation. The variational method improves the exactness of the predictions of core parameters. Although, it increases running time, using high speed PCs or parallelization can compensate. The P-PSOSA code developed in this study has been applied to a PWR core. In addition, in order to analyze the effect of the objective function on the results of the LP optimization problem, we applied two different fitness function equations. It is concluded that both used equation can be used for multi-objective fuel management optimization problem. It can be also said that the average results of the fitness function type #1 is better that the one with the fitness function type #2.

References

Abbasi, M., Zolfaghari, A., Minucher, Yousefi, M., 2011. An adaptive finite element approach for neutron transport equation. Nucl. Eng. Des. 241, 2143–2154.
Ackroyd, R.T., 1978. A finite element method for neutron transport—I. Some theoretical considerations. Ann. Nucl. Energy 5, 75–94.
Ackroyd, R.T., 1997. Finite Element Methods for Particle Transport Applications to Reactor and Radiation Physics. Research Studies Press Ltd, England.
Ackroyd, R.T., Nanneh, M.M., 1990. Hybrid variational principles and synthesis method for finite element transport calculations. Ann. Nucl. Energy 17, 603–634.
Ackroyd, R.T., Fletcher, J.K., Goddard, A.J.H., Issa, J., Williams, M.M.R., Wood, J., 1987. Some recent developments in finite element methods for neutron transport. Adv. Nucl. Sci. Technol. 19, 381–483.
Axmann, J.K., 1997. Parallel adaptive evolutionary algorithms for pressurized water reactor reload pattern optimizations. Nucl. Technol. 119, 276.
Chao, Y.A., Hu, C.W., Suo, C.A., 1986. A theory of fuel management via backward diffusion calculation. Nucl. Sci. Eng. 93, 78.
Davis, J.A., 1968. Transport error bounds via PN approximation. Nucl. Sci. Eng. 31, 166–176.
De Oliveira, C.R.E., 1986. An arbitrary geometry finite element method for multigroup neutron transport with anisotropic scattering. Prog. Nucl. Energy 18, 227–236.
Downar, T.J., Kim, Y.J., 1986. A reverse depletion method for pressurized water reactor core reload design. Nucl. Technol. 73, 42–54.
Erdog, A., Geçkinli, M., 2003. A PWR reload optimisation code (XCore) using artificial neural networks and genetic algorithms. Ann. Nucl. Energy 30, 35–53.
Fadaei, A.H., Setayeshi, S., 2009. A new optimization method based on cellular automata for VVER-1000 nuclear reactor loading pattern. Ann. Nucl. Energy 36, 659–667.
Fletcher, J.K., 1986. Recent development of the transport theory code, MARC/PN. Prog. Nucl. Energy 18, 75–84.
Freeman, J.A., Skapura, D.M., 1991. Neural Networks Algorithms Application and Programming Techniques. Addison-Wesley Publishing Company.
Gropp, W., et al., September 14, 2007. MPICH2 User's Guide, Version 1.0.6. Mathematics and Computer Science Division, Argonne National Laboratory.
Issa, J.G., Riyait, N.S., Goddard, A.J.H., Stott, G.E., 1986. Multigroup application of the anisotropic fem code FELTRAN to one, two, three-dimensions and R-Z problems, progress in nuclear energy. Prog. Nucl. Energy 18, 251–264.
Khoshahval, F., Zolfaghari, A., Minuchehr, H., Sadighi, M., Norouzi, A., 2010. PWR fuel management optimization using continuous particle swarm intelligence. Ann. Nucl. Energy 37, 1263–1271.
Khoshahval, F., Minuchehr, H., Zolfaghari, A., 2011. Performance evaluation of PSO and GA in PWR core loading pattern optimization. Nucl. Eng. Des. 241, 799–808.

Kim, H.G., Chang, S.H., Lee, B.H., 1993. Optimal fuel loading pattern design using an artificial neural network and a fuzzy rule-based system. Nucl. Sci. Eng. 115, 152–163.
Kim, Y.J., Downar, T.J., Sesonske, A., 1987. Optimization of core reload design for low-leakage fuel management in pressurized water reactors. Nucl. Sci. Eng. 96, 85.
Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M., 1983. Optimization by simulated annealing. Science 220 (4598), 498–516.
Lee, J.H., Lee, H.C., Kim, C.H., 1999. Parallel computing fuel assembly loading pattern optimization by the simulated annealing method. Trans. Am. Nucl. Soc. 80, 228.
Liu, S., Cai, J., 2012. Studies of fuel loading pattern optimization for a typical pressurized water reactor (PWR) using improved pivot particle swarm method. Ann. Nucl. Energy 50 (1), 117–125.
Machado, L., Schirru, R., 2002. The Ant-Q algorithm applied to the nuclear reload problem. Ann. Nucl. Energy 29, 1455–1470.
Meneses, A.A.M., Machado, M.D., Schirru, R., 2009. Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor. Prog. Nucl. Energy 51, 319–326.
Message Passing Interface Forum, 2008. MPI: a Message-Passing Interface Standard Version 1.3. University of Tennessee, Knoxville.
Pacheco, P.S., 1997. Parallel Programming with MPI. Morgan Kaufmann Publishers Inc., San Francisco, California.
Premalatha, K., Natarajan, A.M., 2010. Combined heuristic optimization techniques for global minimization. Int. J. Adv. Soft Comput. Appl. 2 (1).
Sadighi, M., Setayeshi, S., Salehi, A., 2002. PWR fuel management optimization using neural networks. Ann. Nucl. En. 29, 41–51.
Smuc, T., Pevec, D., Petrovic, B., 1994. Annealing strategies for loading pattern optimization. Ann. Nucl. Energy 21, 325–336.
Stevens, J.G., 1995. A Hybrid Method for In-core Optimization of Pressurized Water Reactor Reload Design. Ph.D. Thesis. Purdue Univ.
Splawski, B.A., 1981. TWODOG: a Two-dimensional One-group Finite Element Program for the Solution of Neutron Transport Equation.
Stillman, J.A., Chao, Y.A., Downar, T.J., 1989. The optimum fuel and power distribution for a pressurized water reactor burnup cycle. Nucl. Sci. Eng. 103, 321–333.
Stout, R.B., 1973. Optimization of In-Core Nuclear Fuel Management in a Pressurized Water Reactor. PhD Thesis. Oregon State University.
Terney, W.B., Williamson Jr., E.A., 1982. The design of reload cores using optimal control theory. Nucl. Sci. Eng. 82, 260–288.
Van den Bergh, F., Engelbrecht, A.P., 2004. A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput., 225–239.
Van Laarhoven, P.J.M., Aarts, E.H., 1987. Simulated Annealing: Theory and Applications. D. Reidel Publishing Company, Holland, Dordrecht.
Wall, I., Fenech, H., 1965. The application of dynamic programing to fuel management optimization. Nucl. Sci. Eng. 22, 285–297.
Winfrith, 1985. LWR-WIMS, a Computer Code for Light Water Reactor Calculations. AEE, UK. AEEW-R 1498.
Yamamuto, A., 1997. Comparison between equilibrium cycle and successive multi cycle optimization methods for in-core fuel management of pressurized water reactors. In: Proceedings of joint International Conference on Math, Methods and supercomp, for Nuclear. Application. American Nuclear Society, Inc, New York, pp. 769–781. Saratoga springs.