

## Research Article

# Multiobjective Core Reloading Pattern Optimization of PARR-1 Using Modified Genetic Algorithm Coupled with Monte Carlo Methods

Nadeem Shaukat <sup>1</sup>, Ammar Ahmad <sup>2</sup>, Bukhtiar Mohsin <sup>2</sup>, Rustom Khan <sup>2</sup>,  
Salah Ud-Din Khan <sup>3</sup> and Shahab Ud-Din Khan<sup>4</sup>

<sup>1</sup>Center for Mathematical Sciences (CMS), Pakistan Institute of Engineering & Applied Sciences, Nilore 45650, Islamabad, Pakistan

<sup>2</sup>Department of Nuclear Engineering, Pakistan Institute of Engineering & Applied Sciences, Nilore 45650, Islamabad, Pakistan

<sup>3</sup>College of Engineering, King Saud University, PO-Box 800, Riyadh 11421, Saudi Arabia

<sup>4</sup>National Tokamak Fusion Program, Nilore 45650, Islamabad, Pakistan

Correspondence should be addressed to Nadeem Shaukat; [nadeem\\_shaukat07@hotmail.com](mailto:nadeem_shaukat07@hotmail.com)

Received 9 January 2020; Revised 3 February 2021; Accepted 8 April 2021; Published 19 April 2021

Academic Editor: Mark Deinert

Copyright © 2021 Nadeem Shaukat et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to maximize both the life cycle and efficiency of a reactor core, it is essential to find the optimum loading pattern. In the case of research reactors, a loading pattern can also be optimized for flux at an irradiation site. Therefore, the development of a general-use methodology for core loading optimization would be very valuable. In this paper, general-use multiobjective core reloading pattern optimization is performed using modified genetic algorithms (MGA). The developed strategy can be applied for the constrained optimization of research and power reactor cores. For an optimal reactor core reloading design strategy, an intelligent technique GA is coupled with the Monte Carlo (MC) code SuperMC developed by the FDS team in China for nuclear reactor physics calculations. An optimal loading pattern can be depicted as a configuration that has the maximum  $k_{\text{eff}}$  and maximum thermal fluxes in the core of the given fuel inventory keeping in view the safety constraints such as limitation on power peaking factor. The optimized loading patterns for Pakistan Research Reactor-1 (PARR-1) have been recommended using the implemented strategy by considering the constraint optimization, i.e., to maximize the  $k_{\text{eff}}$  or maximum thermal neutron flux while maintaining low power peaking factor. It has been observed that the developed intelligent strategy performs these tasks with a reasonable computational cost.

## 1. Introduction

Nuclear fuel management is one of the most significant considerations when managing smaller research reactors. In these applications, the reactors must secure a wide variety of safety restraints while also providing high thermal neutron flux for external experiments. Core optimization can be performed at the assembly level by optimizing fuel composition or at the core level by optimizing the assembly loading pattern. Overall, many aspects of core operation are very sensitive to loading patterns such as power density, burnup, reactivity distribution, and accumulation of fissile

materials. Naive calculations for core loading optimization techniques can be prohibitively expensive due to the computational cost of simulating each arrangement of the reactor core. This is especially true because a design may need to simultaneously optimize multiple parameters such as peaking power, radial leakage, and end-of-cycle core reactivity [1]. This paper presents an intelligent technique using modified genetic algorithms to perform these tasks with reasonable computational cost.

Extensive studies have been done to optimize the in-core fuel loading pattern during the last few decades, resulting in numerous perspective processes. The loading pattern must

ensure the safety and efficiency of a research reactor. Several different techniques were used in the past for core loading optimization; some of them were simulated annealing method [2–4], genetic algorithms (GAs) [5–8], simple tabular search [9], particle swarm optimization [10], and artificial neural network [11–14]. The GA was first introduced by Holland in the 1970s as one of the stochastic optimization techniques inspired by the mechanism of the natural evolution [15]. It has demonstrated to overcome traditional techniques in many complex problems. By using a GA, an in-core fuel management strategy has been developed in the past to optimize more than one parameter [2].

The GA differs from most of the optimization techniques by searching for one group of solutions to another, rather than focusing on one solution to another. This makes it suited uniquely to most of the multiobjective optimization problems. An improvement of new coding procedure for GA is capable of searching automatically for the optimal fuel loading patterns most suitable for the research reactor [15]. The standard bit-based genetic operators in GA are used to optimize the arrangement of assemblies, burnable absorber, and burnt assembly orientations in the pressurized water reactor (PWR) [12]. The application of GA to reloading pattern optimization problems of a PWR allows pattern from different distributions to combine and reproduce. An optimized pattern in GA is better than that obtained from linear programming [14]. The multiobjective GA (MOGA) used three integer-based arrays to perform the multiobjective optimization of fuel loading patterns of PWR. The MOGA makes it possible to identify all the competing objective surfaces in a single optimization run [5]. In comparison of GA to simulated annealing, the GA optimization performed better in the initial global search, but simulated annealing was better for the local search. GAs seem to perform better in solving LP optimization problems [14].

In nuclear design calculation, the effective multiplication factor  $k_{\text{eff}}$  and the neutron flux distributions are considered the most foundational evaluation quantities. It is used broadly to derive many important formulations in nuclear design calculation such as control rod worth and the reactivity coefficient. Optimizing the placement of fuel assemblies in the core is the most effective way to increase the utilization of a research reactor. Therefore, one needs to maximize  $k_{\text{eff}}$  and neutron flux densities inside the reactor core [16–20]. In the present study, loading patterns for PARR-1 have been proposed by maximizing the effective multiplication factor and neutron thermal flux while keeping low power peak.

The main objectives of the fuel management optimization problems considered in the present work are listed below:

- (i) Maximize the  $k_{\text{eff}}$
- (ii) Maximize thermal neutron flux in the flux trap

The study follows the two constraints:

- (i) Keep the local power peaking factor lower than a predetermined value to maintain fuel integrity.

- (ii) The positions of control fuel elements (CFEs) in the core are fixed, but they can swap their position among the CFEs.

## 2. Materials and Methods

**2.1. SuperMC Computer Code.** Practical computer simulations of a nuclear system like reactors use either deterministic or Monte Carlo codes. Due to continuous energy cross-sections and flexible geometrical capabilities, the Monte Carlo (MC) method is distinctive to simulate complicated nuclear systems and is envisioned as a routine method for nuclear design and analysis in the future. The SuperMC computer code, developed by the FDS team in China, is a general-purpose CAD-based MC-based radiation transport code. It can simulate neutrons, photons, and their coupled behavior. It performs transport calculation, geometry and physics modeling, and visualization of results and geometry. It has been developed and verified by using a series of benchmarking cases such as the fusion reactor ITER model and the fast reactor BN-600 model. SuperMC is still in its evolution process toward a general and routine tool for the simulation of nuclear systems [21].

SuperMC computer program exploits the hybrid MC-deterministic method concept and advanced computer technologies. Being general-purpose radiation transport program, SuperMC is designed for high-fidelity simulation of nuclear-system problems such as reactor analysis, radiation shielding and dosimetry, medical physics, and radiation detection. SuperMC code can be applied to transport calculation of various types of particles, depletion and activation calculation including isotope burnup, material activation and shutdown dose, and multi-physics coupling calculation including thermo-hydraulics, fuel performance, and structural mechanics. The powerful bidirectional automatic conversion between general CAD models and calculation models can be easily performed. Results and the process of simulation can be visualized with a dynamical 3D data set and geometry model.

Advanced cloud computing framework makes the simulation, which is extremely intensive in computation and data storage, more attractive just as a network service. The modular design and generic interface promotes its flexible manipulation and coupling of external solvers. The architecture of SuperMC is shown in Figure 1.

**2.2. General Description about the Strategy Developed.** The optimization strategy presented in this paper is aimed at producing the best possible core loading pattern for any core by modeling of the core using SuperMC. It is based on the probabilistic approach of genetic algorithm. The detail of the application of the genetic algorithms will be given later in the description of the code. The variables normally available in selecting a fuel loading plan are fuel enrichment of fissile content of the fuel, the number of fresh fuel assemblies to be loaded, the arrangement of the fresh and partially spent fuel assemblies in the reactor, and the techniques used to control the excess reactivity of the reactor during the cycle. The code

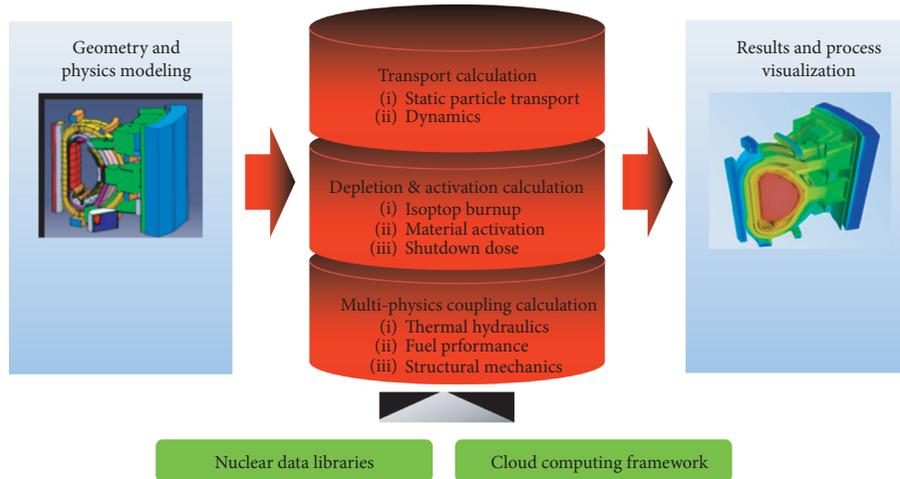


FIGURE 1: Schematic architecture of the SuperMC radiation transport code.

has provision for optimization based on three different factors, i.e., it optimizes the core on the basis of maximum reactivity, which in turn represents longer cycle length; maximum thermal or fast flux in a fixed element, e.g., flux trap or beam tubes for material test reactors; or maximum flux in a particular region of the core.

The optimization of the core is also subject to a constraint on the power peaking factor. The value of power peaking factor must not exceed the designed maximum limit for the reactor. This has been incorporated by reducing the fitness of the loading pattern that violates the constraint so much that chance of its propagation becomes almost zero. The user also has the liberty to select the population size, the total number of generations, crossover fraction, mutation fraction, and the fraction of population of loading patterns that is to be promoted into the next population as being elite.

**2.3. Description of the PARR-1 Core.** The reactor core considered in this study is Pakistan Research Reactor-1 (PARR-1), the swimming pool-type material test research reactor (MTR). PARR-1 has a full power of a 10 MW, utilizing the low-enriched uranium fuel (LEU) of uranium silicide ( $U_3Si_2-Al$ ) fuel contained less than 20% of enriched  $^{235}U$ . The reactor core consists of identical standard fuel elements (SFEs), control fuel elements (CFEs), and an irradiation water box (WB) at the center of the core created by replacing a fuel element. The core immersed in the demineralized light water acts as a coolant, moderator, and reflector. The core is reflected on two sides left and right by the light water, while behind a gamma lead shield of thickness 12.7 cm, light water, and graphite column reflect the third side. Moreover, both graphite and light water reflect the fourth side [22]. The core filled with assemblies with FA types is shown in Table 1. Core configuration modeled by SuperMC and its operating conditions are given in Figure 2 and Table 2, respectively.

**2.4. Description of PARR-1 Fuel Assembly.** The full core consists of  $6 \times 5$  arrays, containing 29 fuel elements and a flux trap in the center and assembled with partial graphite

TABLE 1: Types of fuel assemblies in the PARR-1 core.

FA type	Number of FAs
Standard fuel elements (SFEs)	24
Control fuel elements (CFEs)	5
Water box (WB)	1
Total	30

reflector and one thermal column, all of which are shown in Figure 2. The standard of the fuel element is modeled in three regions. Region 1 is home to  $6.275 \text{ cm} \times 8.10 \text{ cm}$  uranium metal. Regions 2 and 3 are thin claddings of  $0.718 \text{ cm} \times 8.10 \text{ cm}$  aluminum in Figure 3(a). Figure 3(b) represents the control fuel element, which is divided into nine regions. Region 1 located in the central part housing an absorber rod with dimensions of  $6.275 \text{ cm} \times 3.370 \text{ cm}$ . Two-fuel regions 2 and 3 are above and below this absorber region with a dimension of  $0.718 \text{ cm} \times 2.1965 \text{ cm}$ , respectively. Six regions on both sides represent the side plate with dimensions of  $0.718 \text{ cm} \times 2.5335 \text{ cm}$ ,  $0.718 \text{ cm} \times 3.370 \text{ cm}$ , and  $0.718 \text{ cm} \times 3.1965 \text{ cm}$ , respectively. The material composition of the control rod is 80% Ag, 15% In, and 5% Cd. All these five control rods are used as safety rods, one of which with the lowest reactivity worth is used as a power regulating rod [22]. The standard fuel element and control fuel element modeled in SuperMC are shown in Figure 4.

**2.5. Core Neutronic Calculations Using SuperMC.** The core neutronic calculations for PARR-1 to evaluate the objective functions, i.e.,  $k_{\text{eff}}$  and thermal fluxes were preceded using SuperMC. The simulations for the PARR-1 core neutronic calculations were run for 100,000 neutron histories for a total of 120 cycles (20 inactive cycles, 100 active cycles). The 100 independent simulations were run to get the objective function values corresponding to the 100 randomly generated different loading patterns as an initial population of MGA.

**2.6. Modified Genetic Algorithm Characteristics.** To determine, which chromosomes will be chosen as the basis of the

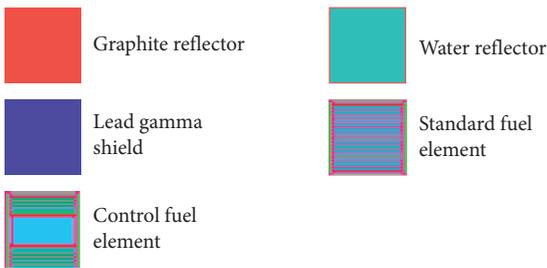
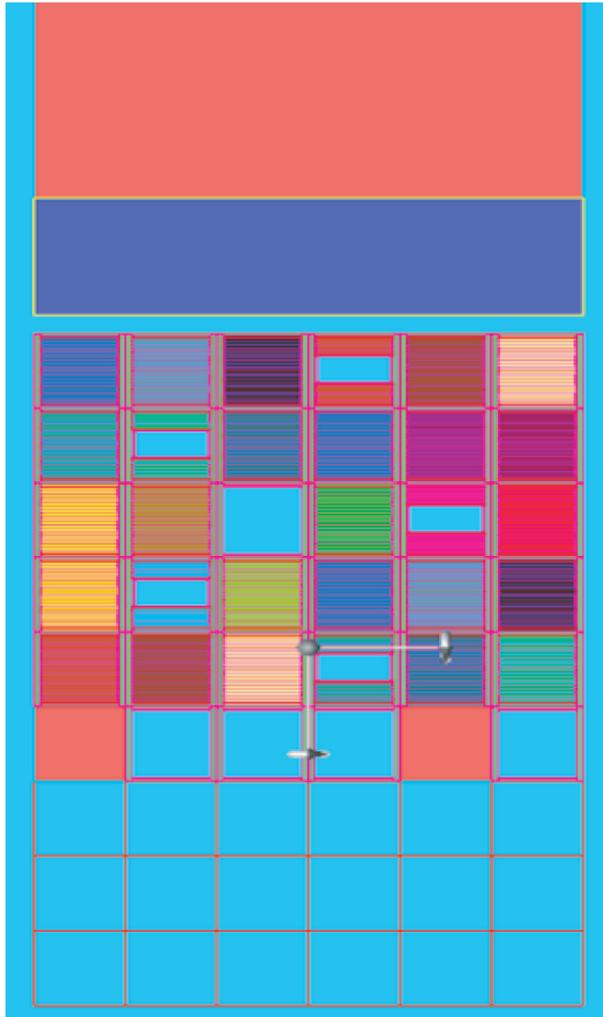


FIGURE 2: PARR-1 full-core configuration.

subsequent generation, the reproduction is considered in the simple form of conventional GA. The best chromosome from the last population might be lost from generating populations from only two parents. The selection of crossover or mutation or both operations to generate new chromosome may not reach the best fitness of the objective function value. Thereby, it can be said that the best fitness found through the simple genetic algorithms cracked from the new population may be inferior to the old generations. The objective of the proposed MGA is to avoid this disadvantage. The structure of the MGA is quite similar to the simple GA. However, the MGA has been distinguished from the simple GA in that the new offspring are generated after

TABLE 2: Main operating conditions for PARR-1.

Parameters	Specific operating conditions
Coolant and moderator	Water
Reflector	Water, graphite
Cladding	Aluminum
Fuel rod	Fork type
Fuel material	$U_3Si_2-Al$
Fuel enrichment (w%)	19.75
Uranium density ( $g/mm^3$ )	4.5
Fuel element dimensions ( $mm^3$ )	$76 \times 80 \times 600$
Lattice pitch (mm)	$8.1 \times 7.7$
Shape of fuel plates	Flat
Coolant channel thickness (mm)	2.19
Weight of $^{235}U$ (g)	
Standard fuel element	390
Control fuel element	288
Number of fuel plates	
Standard fuel element	23
Control fuel element	17
Number of dummy plates in CFE	4
Fuel plate dimensions (mm)	
Width	66.4
Inner plate thickness	1.27
Outer plate thickness	1.50
Thickness of cladding (mm)	
Inner plate	0.38
Outer plate	0.495
Fuel meat dimensions (mm)	
Length	600
Width	63
Thickness	0.51
Side plate dimensions (mm)	
Thickness	4.8
Width	80

both the crossover and mutation operations have been applied. Thus the deterioration problem will not take place on the grounds that the best fitness from the new generated offspring will be the fittest or at least the similar with the previous fitness.

There are many operators, parameters, and other settings involved in the genetic algorithms. These ingredients for a GA can be incorporated differently in various problems. The appropriate selection of these ingredients in the evolutionary process plays a vital role in convergence; otherwise, the process will make the GA susceptible to premature convergence. Therefore, the better choice of a specific crossover depending on the nature of the problem can improve the performance of the genetic algorithm.

The genetic parameters like crossover and mutation probability, numbers of individuals and generations should be determined depending on the regarded problem. The population size and the number of generations are particularly considered to be the important parameters of GA. In case of too low number of chromosomes, the crossover operator will have a less number of opportunities to perform crossover. Then, it will explore only a small part of search space. On contrary, if there is too large number of chromosomes, the performance of the GA will be slowed down. Several runs of MGA have been

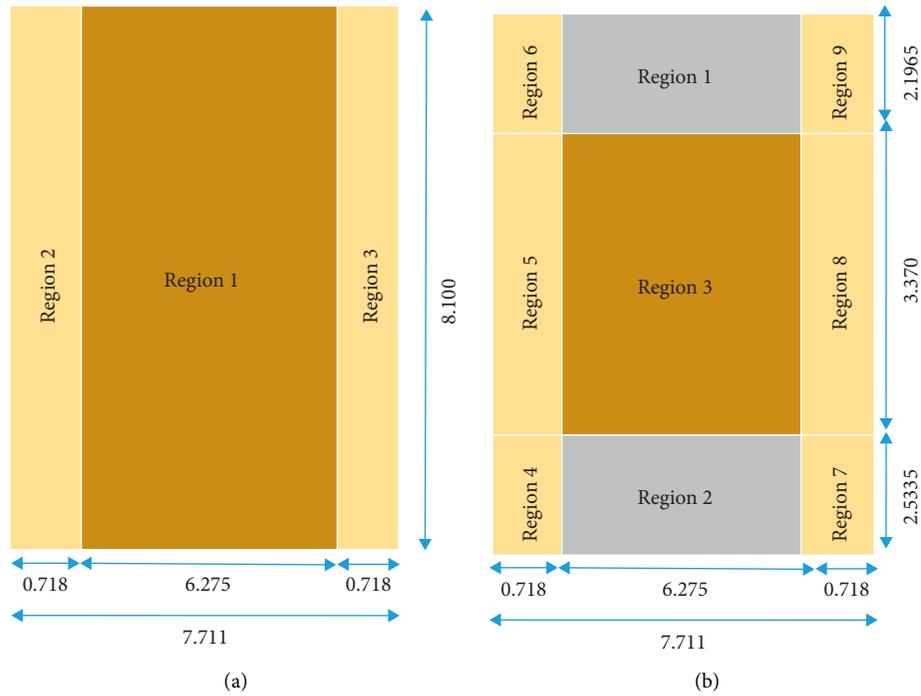


FIGURE 3: The planar model of fuel assembly. (a) Standard fuel element. (b) Control fuel element.

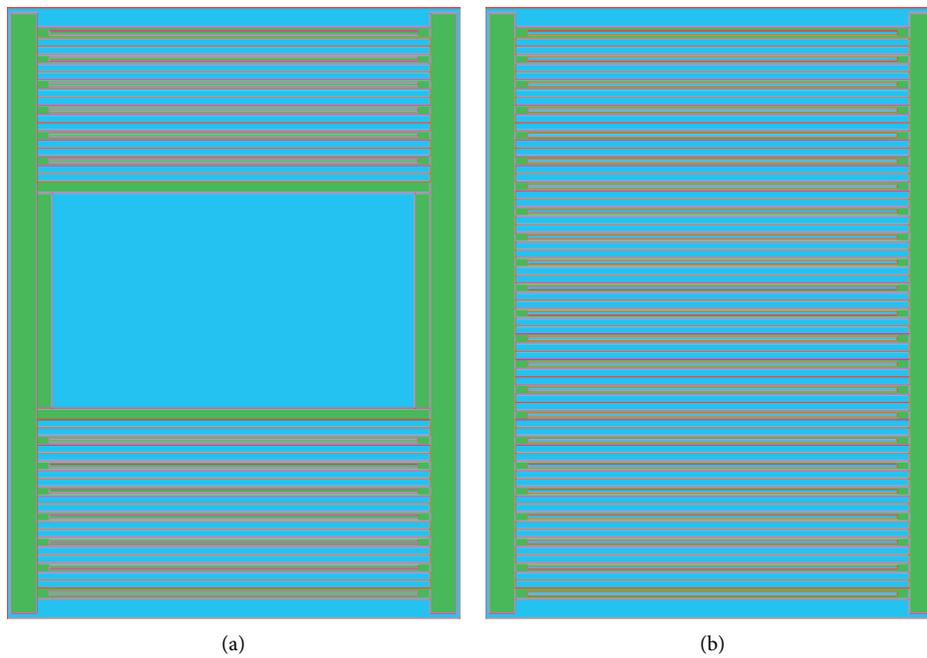


FIGURE 4: SuperMC model of control fuel element (a) and standard fuel element (b).

performed with different values of the above-mentioned genetic parameters. Considering the best solution in the shortest time, the genetic parameters were adjusted as presented in Table 3.

Optimization of PARR-1 using MGA and SuperMC is the novelty in this paper. The core of PARR-1 has never

modeled before using SuperMC code and never been optimized using MGA in the past. The name-modified GA has been used because it is different in some aspects from a traditional GA. In a conventional GA, reproduction in subsequent generation is considered to choose the chromosomes for next generations. The best chromosomes from

TABLE 3: Initial population sensitivity with varying number of individuals.

Number of individuals	Objective function value averaged over 100 independent runs of MGA	
	$k_{\text{eff}}$ (std. dev.)	Maximum power peaking factor (std. dev.)
50	1.037871 ( $\pm 0.00023$ )	1.6781 ( $\pm 0.00029$ )
100	1.039297 ( $\pm 0.00010$ )	1.6982 ( $\pm 0.00013$ )
150	1.039296 ( $\pm 0.00011$ )	1.6981 ( $\pm 0.00013$ )
200	1.039297 ( $\pm 0.00012$ )	1.6983 ( $\pm 0.00013$ )
250	1.039297 ( $\pm 0.00011$ )	1.6981 ( $\pm 0.00011$ )
300	1.039298 ( $\pm 0.00011$ )	1.6982 ( $\pm 0.00012$ )
350	1.039296 ( $\pm 0.00011$ )	1.6982 ( $\pm 0.00013$ )
400	1.039297 ( $\pm 0.00010$ )	1.6981 ( $\pm 0.00011$ )
450	1.039298 ( $\pm 0.00011$ )	1.6982 ( $\pm 0.00012$ )
500	1.039297 ( $\pm 0.00012$ )	1.6982 ( $\pm 0.00012$ )

the last population may get missed by generating population only from two parents. The selection of crossover or mutation or both operations to generate new chromosomes may not converge to the best fitness value of the evaluation function. This disadvantage of conventional GA can be avoided by the application of proposed modified GA. In the modified GA, the new descendants are generated by the application of both crossover and mutation operations. Thus the missing of the chromosome having better fitness value than the previous can be avoided in this way or at least the offspring will have the similar fitness than the subsequent generation.

The flow scheme of the designed strategy is given as follows:

- (1) The total number of assemblies; different types of assemblies; their indices; the optimization method; and limits on the constraints, the optimization function, and the assemblies are saved along with the parameters required for genetic algorithms from the file "CORES.inp."
- (2) The Python code was developed to generate the SuperMC readable input file with the changing core loading pattern for each simulation.
- (3) SetCoreLP.txt is an input file for Python to set the core loading pattern to generate the INPUT.txt readable for SuperMC.
- (4) All the possible locations of each assembly type are noted.
- (5) The indices assigned by the user to each assembly are saved in a one-dimensional array that serves as the chromosome for the genetic algorithm.
- (6) Initial population of different such chromosomes containing these indices at different locations are generated.
- (7) For each chromosome, a 2-D core map is generated in which the position of each assembly depends on the position of its corresponding index in the chromosome.
- (8) This core is used for the objective function calculation by SuperMC.
- (9) The value of the optimization function is saved.

- (10) The power peaking factor is calculated, and the chromosomes resulting in a peaking factor greater than the limiting value of the core are placed on least priority.
- (11) Parents for crossover are selected using the selection function, which assigns high probability to the chromosomes with high fitness.
- (12) A new population is generated using genetic algorithms and the user-defined fractions of different ways of generation like mutation, crossover, and elite count.
- (13) The steps numbered 5 to 10 are repeated for new population.

This process continues until the maximum allowed number of iterations has been reached or the given tolerance level is met.

The flow chart in Figure 5 depicts the whole process.

### 3. Numerical Results and Discussion

The optimization of loading pattern for the PARR-1 core was performed subject to a constraint on the power peaking factor. The fitness function is defined for obtaining the best configuration of the reactor to obtain the maximum  $k_{\text{eff}}$  under safety considerations of the maximum allowable power peaking factor of 2.089 [16]. This has been incorporated by reducing the fitness of the loading pattern that violates the constraint so much that chances of its propagation become almost zero. Initially, 100 randomly generated loading patterns were simulated as an initial population. The total number of generations, crossover fraction, mutation fraction, and the fraction of population of loading patterns that was to be promoted into the next population as being elite were set for the calculation of the best loading pattern and are presented in detail in Table 4. The total number of assemblies, types of assemblies, their indices, limit on the power peaking constraint, the optimization function, and the assemblies were input along with the parameters required for GA. All the possible locations of each assembly type were noted. The indices assigned to each assembly were saved in a one-dimensional array that served as the chromosome for the GA. The initial

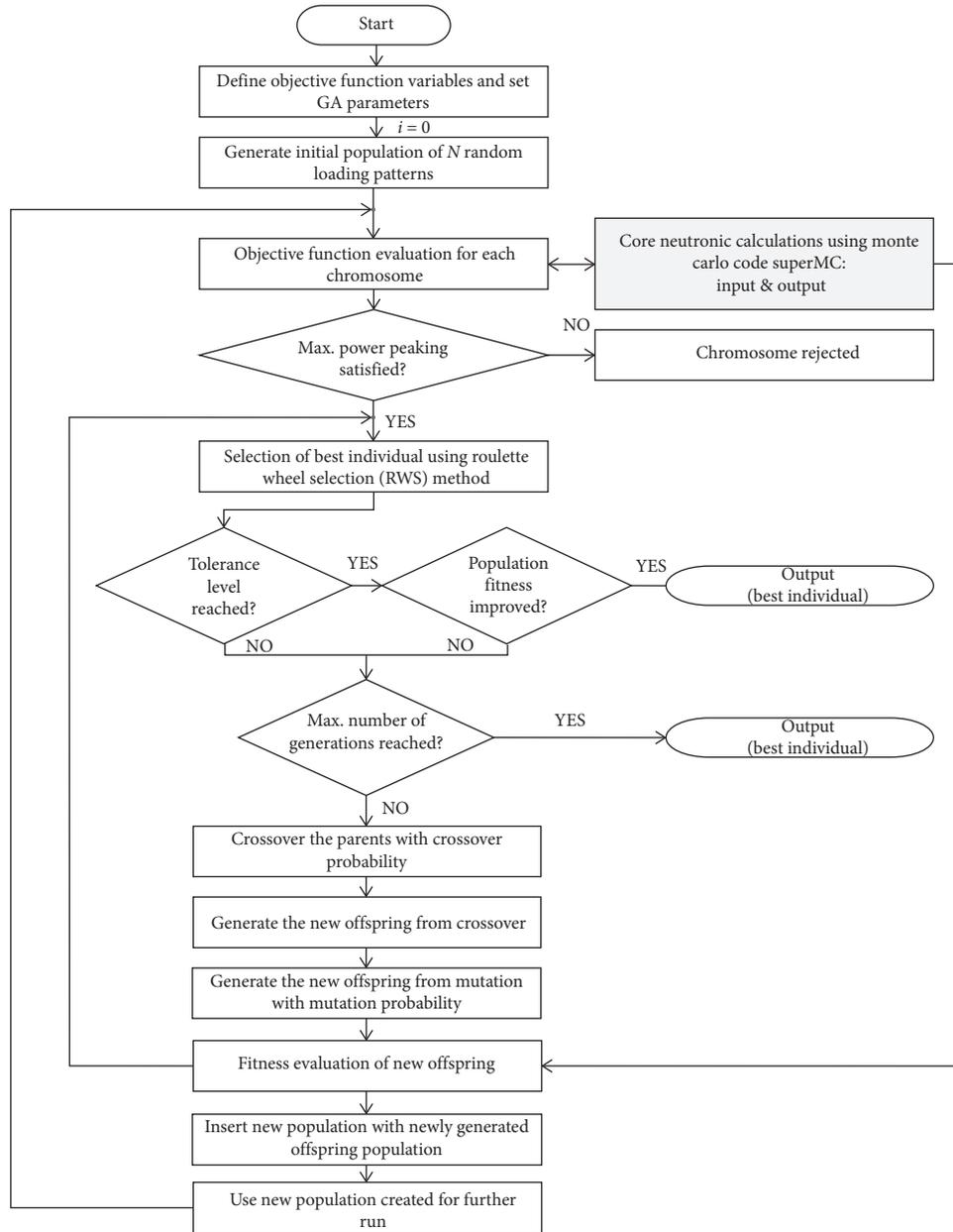


FIGURE 5: The flow chart of the modified genetic algorithm.

TABLE 4: Parameters used in MGA optimization.

Cell types	FAs in PARR-1 core
Number of chromosomes in each generation	100
Number of generations	100
Fitness limit	12
Crossover fraction	0.8
Elite count	0.3
Number of crossovers	48
Mutation fraction	0.2

population is generated of such different chromosomes containing these indices at different locations. For each chromosome, an input file is generated using the Python

script, which is readable for SuperMC in which the position of each assembly depends on the position of its corresponding index in the chromosome. The objective function values for  $k_{eff}$  and thermal neutron fluxes in the flux trap were obtained using SuperMC. The values of the optimization functions were saved. The power peaking factor was calculated. The chromosomes resulting in a peaking factor greater than the limiting value in the core are placed on the least priority. Parents for crossover are selected using the function to assign high probability to the chromosomes with high fitness. A new population was generated using GAs and the defined fractions of different ways of generation like mutation, crossover, and elite count. A total of 100 generations were simulated in a similar way to get the optimal solution.

TABLE 5: Stability study of MGA in the first 10 independent simulations.

Independent run	$k_{\text{eff}}$	Maximum power peaking factor
1	1.039076	1.6981
2	1.038366	1.6972
3	1.037965	1.7035
4	1.039371	1.693
5	1.039073	1.7071
6	1.038369	1.6964
7	1.039374	1.7004
8	1.038607	1.6998
9	1.038067	1.6901
10	1.039285	1.7001
Total mean	1.03929	1.6984
Variance	0.0000003	0.00002

TABLE 6: Group generations having the same optimal  $k_{\text{eff}}$  values.

Group of generations	$k_{\text{eff}}$ averaged over group generations (SD)	Maximum power peaking factor (SD)	Generations
1	1.038386 ( $\pm 0.00026$ )	1.6889 ( $\pm 0.00029$ )	1–2
2	1.038351 ( $\pm 0.00023$ )	1.7019 ( $\pm 0.00028$ )	3–4
3	1.039018 ( $\pm 0.00027$ )	1.7136 ( $\pm 0.00031$ )	5–13
4	1.038775 ( $\pm 0.00025$ )	1.7140 ( $\pm 0.00028$ )	14–41
5	1.038907 ( $\pm 0.00024$ )	1.7013 ( $\pm 0.00029$ )	42–67
6	1.038659 ( $\pm 0.00031$ )	1.6947 ( $\pm 0.00029$ )	68–100

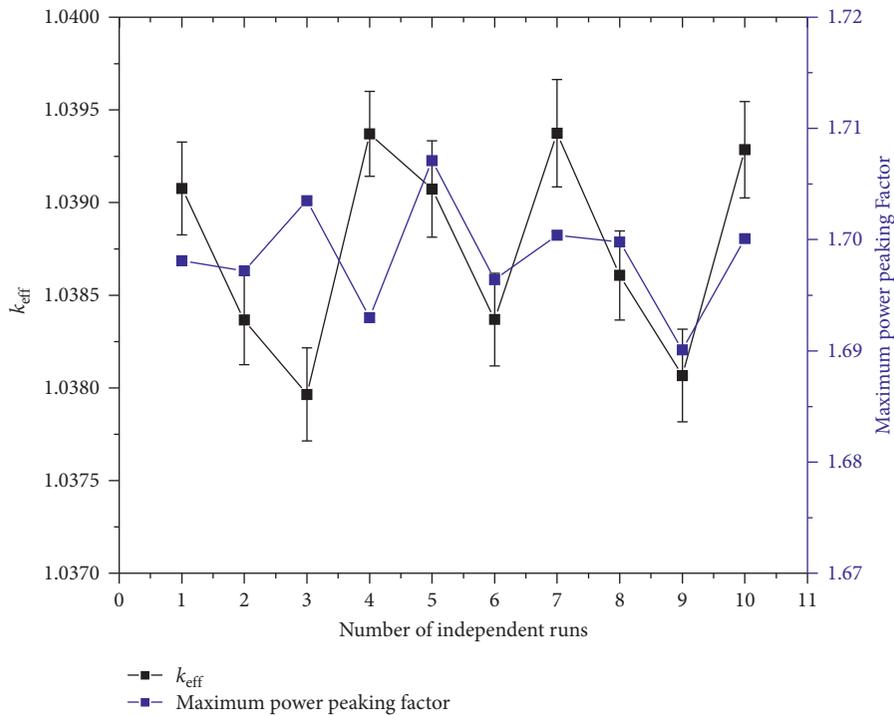


FIGURE 6: Stability study of MGA.

3.1. *Sensitivity of the Initial Population Size.* To check the effect of the number of initial population size on the results, different initial population sizes of 20, 50, 100, 150, and 200 were used. Table 3 shows the sensitivity of initial population.

100 numbers of independent runs were made, and it is observed that 100 numbers of individuals give optimized results. It is also observed that any increase in population size would increase computational cost without increasing

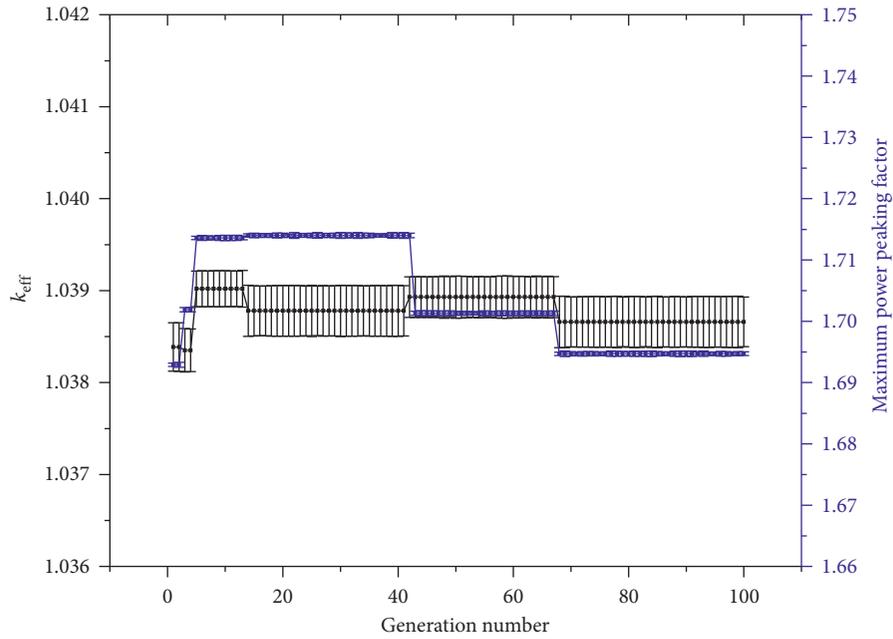


FIGURE 7: Comparison of converged optimal solution of the maximum  $k_{eff}$ .

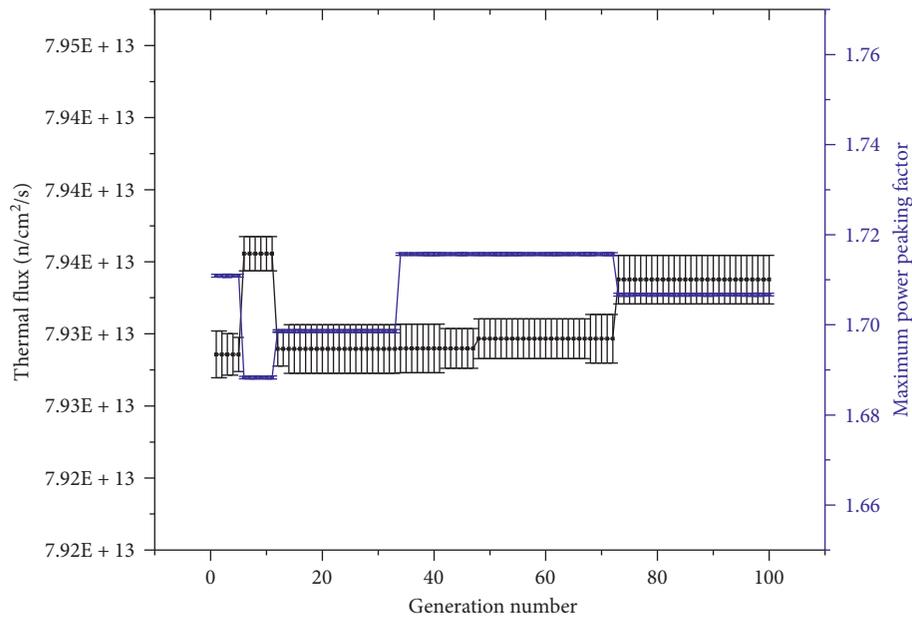


FIGURE 8: Comparison of converged optimal solution of thermal flux.

the accuracy of results and would be unnecessary. Based on this sensitivity analysis, initial population size of 100 is selected for further analysis in this research.

3.2. *Stability Analysis of MGA.* The test of the algorithm stability is one of the most important tests that should be conducted. The objective to check the stability of the algorithm is the issue that the algorithm gives the same answers, and the uniqueness of the optimized answer should be tested. To this end, GA was run 100 times independently and it was observed that the results are in 95% confidence

interval. The results of the first 10 repeated runs of the algorithm along with the mean value and the variance are shown in Table 5 and Figure 1.

Results shown in Table 6 and Figure 6 apparently represent a small difference among results of repeated simulations. The 0.0000003 variance in  $k_{eff}$  and the 0.00002 variance in maximum power peaking factor are very low, which shows high stability of the modified genetic algorithm in different runs for 100 repetitions.

The convergence of the objective function values corresponding to the best loading patterns in each generation

TABLE 7: Group generations having the same optimal thermal flux values.

Group of generations	Thermal flux averaged over group generations (SD)	Maximum power peaking factor (SD)	Generations
1	$7.93086 \times 10^{13}$ ( $\pm 0.00025$ )	1.7109 ( $\pm 0.00027$ )	1–5
2	$7.93645 \times 10^{13}$ ( $\pm 0.00022$ )	1.6883 ( $\pm 0.00029$ )	6–11
3	$7.93116 \times 10^{13}$ ( $\pm 0.00026$ )	1.6986 ( $\pm 0.00029$ )	12–33
4	$7.93119 \times 10^{13}$ ( $\pm 0.00023$ )	1.7157 ( $\pm 0.00028$ )	34–47
5	$7.93173 \times 10^{13}$ ( $\pm 0.00025$ )	1.7067 ( $\pm 0.0003$ )	48–72
6	$7.93501 \times 10^{13}$ ( $\pm 0.0003$ )	1.7032 ( $\pm 0.00027$ )	73–100

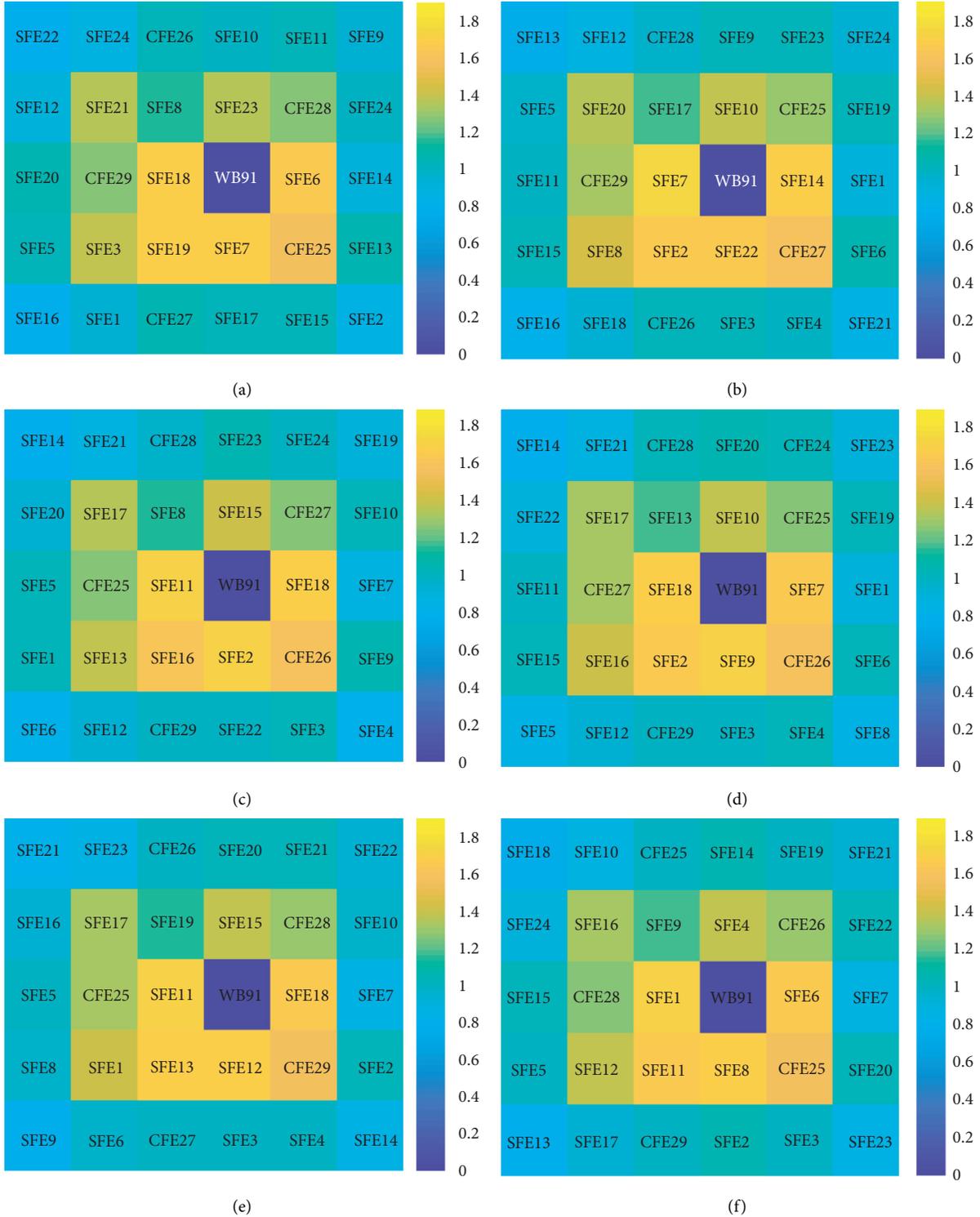


FIGURE 9: Optimal loading patterns for the group generations having the same  $k_{\text{eff}}$ . (a) Group 1:  $k_{\text{eff}} = 1.038386$  ( $\pm 0.00026$ ). (b) Group 2:  $k_{\text{eff}} = 1.038351$  ( $\pm 0.00023$ ). (c) Group 3:  $k_{\text{eff}} = 1.039018$  ( $\pm 0.00027$ ). (d) Group 4:  $k_{\text{eff}} = 1.038775$  ( $\pm 0.00025$ ). (e) Group 5:  $k_{\text{eff}} = 1.038907$  ( $\pm 0.00024$ ). (f) Group 6:  $k_{\text{eff}} = 1.038659$  ( $\pm 0.00031$ ).

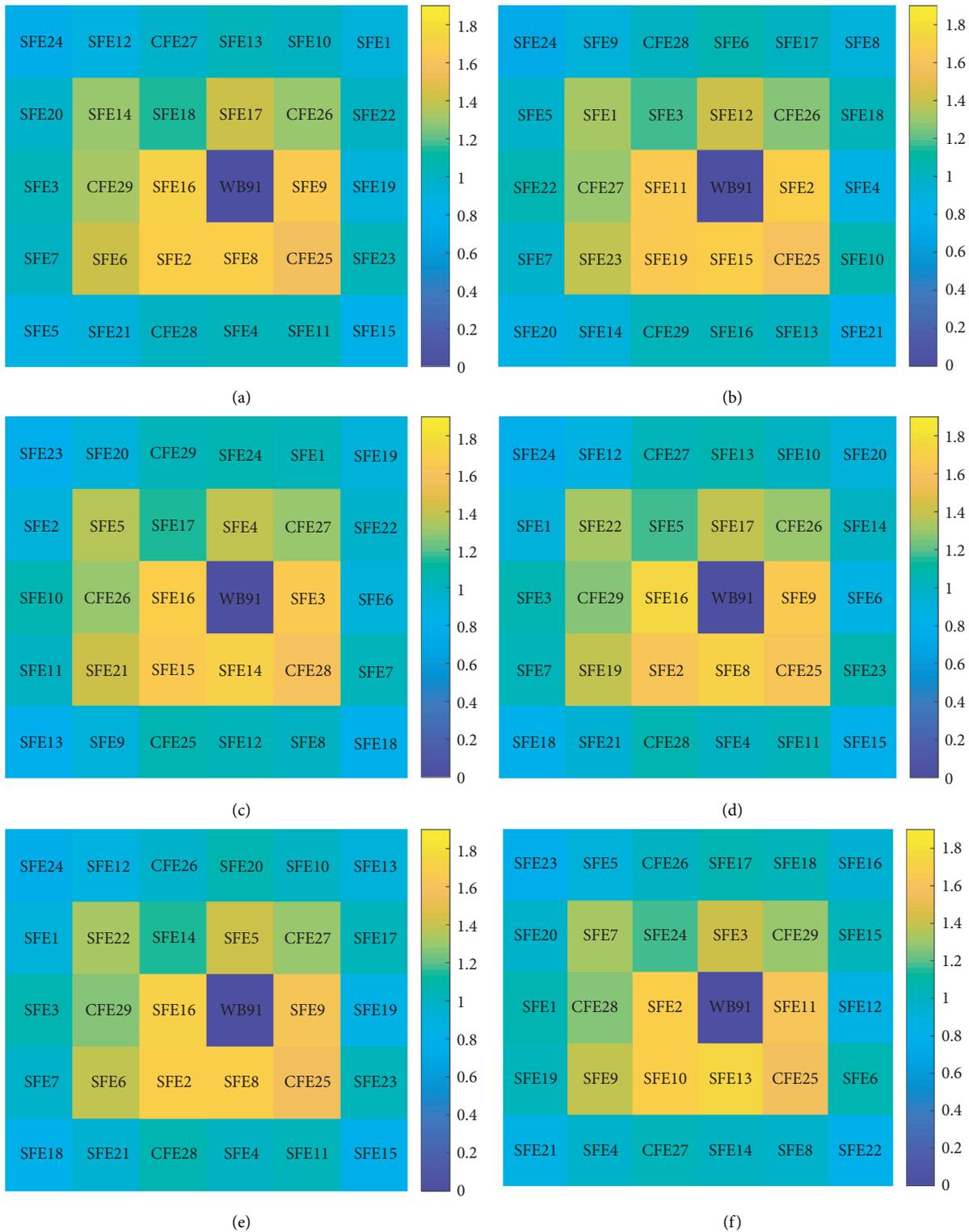


FIGURE 10: Optimal loading patterns for the group generations having the same thermal flux. (a) Group 1:  $\phi_{th} = 7.93086 \times 10^{13} (\pm 0.00025)$ . (b) Group 2:  $\phi_{th} = 7.93645 \times 10^{13} (\pm 0.00022)$ . (c) Group 3:  $\phi_{th} = 7.93116 \times 10^{13} (\pm 0.00026)$ . (d) Group 4:  $\phi_{th} = 7.93119 \times 10^{13} (\pm 0.00023)$ . (e) Group 5:  $\phi_{th} = 7.93173 \times 10^{13} (\pm 0.00025)$ . (f) Group 6:  $\phi_{th} = 7.93501 \times 10^{13} (\pm 0.0003)$ .

with the maximum  $k_{eff}$  value and maximum thermal neutron flux is illustrated in Figures 7 and 8, respectively. The maximum power peaking factors for the cores during the convergence are also shown in Figures 7 and 8 for

corresponding objective functions. After a certain number of generations, the solution was converged to the most optimal value. After the completion of 67 generations, the value of  $k_{eff}$  was converged to 1.038659. For the thermal neutron flux,

TABLE 8: Summary of computational time to get the best optimal solution.

No. of CPUs	No. of histories per cycle	No. of independent simulations	Computational time for each independent simulation (CH)	Total computational time for 100 simulations (CH)	Computational time for GA calculations (CH)	Total computational time to get optimal solution (CH)
50	100,000	100	~1.22	~122	~0.1	~122.1

it started to converge to the optimal solution at the 73rd generation with flux of  $7.93501 \times 10^{13}$ . The error bars in the corresponding figures indicates the  $\pm 2$  standard deviation (SD) intervals of the objective function values, where the SDs are calculated by the square root of the sample variance from the group generation-wise  $k_{\text{eff}}$  values and the thermal fluxes of the GA calculations.

Tables 6 and 7 show the groups of generations having the same optimum value of  $k_{\text{eff}}$ , thermal fluxes, and the maximum value of power peaking factors of the core, respectively. The best optimum loading patterns for the group of generations having the same value of  $k_{\text{eff}}$  and thermal fluxes are shown in Figures 9 and 10, respectively. The colormap shows the normalized radial power distribution in the core.

Summary of the computational time to get the best optimal loading patterns is given in Table 8. The parallel simulations of the Monte Carlo code for the calculation of objective function values were run on 50 CPUs. It takes approximately 122 core-hours (CH) to complete 100 independent simulations on 50 CPUs. To find the best optimal loading pattern for PARR-1 core, the GAs take around ~0.1 core-hours (CH) on the master node with system specifications of Intel (R), Core (TM) i7-7700 CPU @ 3.60 GHz 3.60 GHz. Therefore, the total computational time to get optimal solution was approximately 122.1 CH.

#### 4. Conclusions

This study has demonstrated a powerful multiobjective optimization method for the nuclear fuel loading optimization problem of the MTR-type research reactor PARR-1. The SuperMC code has been successfully applied for the neutronic calculations of PARR-1 core and combined with the modified GA to evaluate the objective function values. The results of optimization illustrate the efficiency of the MGA search for the optimal fuel loading pattern to obtain its maximum effective multiplication factor and the maximum thermal neutron fluxes in the flux trap region while keeping the power peaking factor under design limits. It is worth mentioning the results that the optimum value of  $k_{\text{eff}}$  is 1.038659 and the optimal thermal neutron flux in the trap is  $7.93501 \times 10^{13}$  n'/cm<sup>2</sup>/sec. Moreover, based on the present research it can be concluded that MGA coupled with SuperMC can be efficiently used for the loading pattern optimization of PARR-1. Also, the best optimal loading pattern can be obtained in around ~122.1 core-hours on the previously mentioned computer system details.

#### Data Availability

Due to confidential matters, data might not be shared.

#### Additional Points

(i) Optimal core reloading patterns were generated for Pakistan Research Reactor-1. (ii) SuperMC has been used for the core neutronic calculations. (iii) A powerful multi-objective optimization method GA was coupled with Monte Carlo method. (iv)  $k_{\text{eff}}$  and thermal neutron flux were considered as an objective function for optimization. (v) The local power peaking factor was kept lower than a pre-determined value. (vi) The best reloading patterns were recommended by using the implemented strategy.

#### Disclosure

This work was also performed under the research contract between Key Lab of Neutronics and Radiation Safety, INEST, CAS, China, and PIEAS, Pakistan.

#### Conflicts of Interest

There are no conflicts of interest.

#### Acknowledgments

The authors sincerely appreciate funding from Researchers Supporting Project number (RSP-2021/58), King Saud University, Riyadh, Saudi Arabia. The authors thank the members of the FDS team for their great help for this research.

#### References

- [1] D. J. Kropackzek and P. J. Turinsky, "Incore nuclear fuel management optimization for pressurized water reactors utilizing simulated annealing," *Nuclear Technology*, vol. 95, 2017.
- [2] A. Yamamoto and H. Hashimoto, "Application of temperature parallel simulated annealing to loading pattern optimizations of pressurized water reactors," *Nuclear Science and Engineering*, vol. 136, no. 2, pp. 247–257, 2000.
- [3] D. J. Kropackzek and R. Walden, "Large-scale Application of the constraint annealing method for pressurized water reactor core design optimization," *Nuclear Science and Engineering*, vol. 193, no. 5, pp. 1–14, 2019.
- [4] T. K. Park, H. G. Joo, C. H. Kim, and H. C. Lee, "Multi-objective loading pattern optimization by simulated annealing employing discontinuous penalty function and screening technique," *Nuclear Science and Engineering*, vol. 162, no. 2, pp. 134–147, 2009.
- [5] G. T. Parks, "Multiobjective pressurized water reactor reload core design by nondominated genetic algorithm search," *Nuclear Science and Engineering*, vol. 124, no. 1, pp. 178–187, 2017.

- [6] A. Thakur and U. Kannan, "Fuzzy logics as an integral part of evolutionary algorithms," *Nuclear Science and Engineering*, vol. 193, no. 10, pp. 1160–1171, 2019.
- [7] P. W. Poon and G. T. Parks, "Application of genetic algorithms to in-core nuclear fuel management optimization," vol. 2, pp. 777–786, in *Proceedings of the Joint International Conference on Mathematical Methods and Super-computing in Nuclear Applications*, vol. 2, pp. 777–786, American Nuclear Society, Karlsruhe, Germany, April 1993.
- [8] J. L. C. Chapot, F. Carvalho Da Silva, and R. Schirru, "A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem," *Annals of Nuclear Energy*, vol. 26, no. 7, pp. 641–655, 1999.
- [9] C. Martín-del-Campo, J. L. François, and L. B. Morales, "Boiling water reactor fuel assembly axial design optimization using tabu search," *Nuclear Science and Engineering*, vol. 142, no. 1, pp. 107–115, 2017.
- [10] A. Ahmed and S. U. I. Ahmad, "Optimization of fuel loading pattern for a material test reactor using swarm intelligence," *Progress in Nuclear Energy*, vol. 103, pp. 45–50, 2018.
- [11] A. Yamamoto, "Application of neural network for loading pattern screening of in-core optimization calculations," *Nuclear Technology*, vol. 144, no. 1, pp. 63–75, 2003.
- [12] H. Mazrou and M. Hamadouche, "Development of a supporting tool for optimal fuel management in research reactors using artificial neural networks," *Nuclear Engineering and Design*, vol. 236, no. 3, pp. 255–266, 2006.
- [13] E. F. Faria and C. Pereira, "Nuclear fuel loading pattern optimisation using a neural network," *Annals of Nuclear Energy*, vol. 30, no. 5, pp. 603–613, 2003.
- [14] A. Erdogan and M. Geckinli, "A PWR reload optimization code (XCORE) using artificial neural networks and genetic algorithms," *Annals of Nuclear Energy*, vol. 30, pp. 35–53, 2003.
- [15] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [16] N. Shaukat, S.-U.-I. Ahmad, A. Majeed, N. Ahmad, and B. Mohsin, "Optimization of core reload pattern for PARR-1 using evolutionary techniques," *Nuclear Engineering and Design*, vol. 240, no. 10, pp. 2831–2835, 2010.
- [17] W. Hongchun, "Pressurized water reactor reloading optimization using genetic algorithms," *Annals of Nuclear Energy*, vol. 28, no. 13, pp. 1329–1341, 2001.
- [18] B. Q. Do and L. P. Nguyen, "Application of a genetic algorithm to the fuel reload optimization for a research reactor," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 977–988, 2007.
- [19] Y. P. Mahlers, "Core loading pattern optimization for research reactors," *Annals of Nuclear Energy*, vol. 24, no. 7, pp. 509–514, 1996.
- [20] E. Tanker and A. Z. Tanker, "Application of a genetic algorithm to core reload pattern optimization," in *Proceedings of the International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analysis*, vol. 1, Portland, OR, USA, January 1995.
- [21] Y. Wu, J. Song, H. Zheng et al., "CAD-based Monte Carlo program for integrated simulation of nuclear system SuperMC," *Annals of Nuclear Energy*, vol. 82, pp. 161–168, 2015.
- [22] S.-U.-I. Ahmad and N. Ahmad, "Effect of updated WIMSD libraries on neutron energy spectrum at irradiation site of Pakistan research reactor-1 using 3D modeling," *Annals of Nuclear Energy*, vol. 32, no. 5, pp. 521–548, 2005.