Nuclear Engineering and Design 415 (2023) 112713

Contents lists available at ScienceDirect

Nuclear Engineering and Design

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

Optimization of IEA-R1 reactor core parameters using the particle swarm algorithm

Thiago A. dos Santos [a,*], Frederico A. Genezini [a], Giovanni L. de Stefani [b]

[a] Nuclear and Energy Research Institute, IPEN, Brazil
[b] Universidade Federal do Rio de Janeiro, UFRJ, Brazil

ARTICLE INFO

Keywords:
Loading pattern optimization
Particle swarm
Research reactors

ABSTRACT

This work aims on the development of a FORTRAN 90 code to solve the Loading Pattern Optimization Problem (LPOP) in the IEA-R1 research reactor at the Nuclear and Energy Research Institute (IPEN/CNEN-SP) in São Paulo, Brazil. The code integrates the Particle Swarm Optimization (PSO) method, with the current reactor calculation methodology. The objective function seeks to maximize the effective multiplication factor (k_{eff}), minimize the peak power factor (PPF), and achieve the most uniform neutron flux distribution possible. A comparison with the parameters of the reactor's current configuration, and the code successfully finds solutions meeting the standard problem requirements. The new configuration enhances peak power (6.93%) and variance (9.62%), slightly increases neutron flux (0.48%), and marginally reduces in k_{eff} (0.45%). Additionally, it lowers the maximum fuel cladding temperature (0.7%), contributing to reactor safety.

1. Introduction

In a nuclear reactor, designing a core to meet operational needs is one of the greatest challenges in reactor physics. This problem becomes more complex during the reload operation, when new fuels are inserted into the core together with others already used. The search for an optimal configuration of the core in question is known as the Loading Pattern Optimization Problem (LPOP).

However, this is an NP-hard combinatorial optimization problem, and solving it through brute force becomes infeasible (Akbari et al., 2018). Even power reactors with symmetric cores face numerous potential configurations, rendering brute force solutions impractical, with the possible solutions often exceeding 15 factorial (15!). Usual conditions for solving the problem are the maximization of the fuel burning cycle time, the intensity of the neutron flux produced, the flattening of the power distribution and the minimization of the power peak.

Common optimization criteria include maximizing fuel cycle duration, neutron flux intensity, power distribution uniformity, and minimizing power peaks.

In research reactors like IEA-R1 at IPEN/CNEN, core asymmetry expands the solution space. For example, IEA-R1′s core, with 20 fuel elements, results in approximately 2.4×10^{18} potential configurations. In a computer where each configuration is calculated in one second, the total calculation would take about 77 billion years.

Metaheuristic algorithms, including Particle Swarm Optimization (PSO), have been vital for solving LPOP. Several works use the most varied methods, such as Simulated Annealing (SA) (Kropaczek and Turinsky, 1991; Tran et al., 2021; Lee et al., 2001), Artificial Neural Networks (ANN) (Mazrou and Hamadouche, 2006), Genetic Algorithms (GA) (Tanker and Tanker, 1994; DeChaine and Feltus, 1995; Do and Nguyen, 2007), Artificial Bee Colony Algorithm (ABC) (de Oliveira and Schirru, 2011; Safarzadeh et al., 2011), and other methods (Akbari et al., 2018; Eberhart and Kennedy, 1995; Fowler et al., 1972; Haghighat, 2021; Kashi et al., 2014; Khoshahval et al., 2010; Machado and Schirru, 2002). The PSO, created by Russell Eberhart and James Kennedy (Eberhart and Kennedy, 1995), excels at nonlinear, continuous, and discrete optimization problems. Its creation was based on one of several existing studies in the early 1990 s on the social behavior of animals applied to computing. In this case, it was observed that flocks of animals (e.g. fish, bees and birds) had a competitive advantage for the survival of a given individual if he shared his knowledge with the other members of the group. Thus, in the PSO, an individual makes a comparison between himself and his closest neighbors, to identify the one that is the best and then proceeds to imitate him. It is one of the most classic swarm algorithms, serving as the basis for other similar methods, such as the ABC method, among others. Its relatively simple implementation, combined

* Corresponding author.
E-mail addresses: thiago.santos@ipen.br (T.A. dos Santos), fredzini@ipen.br (F.A. Genezini), giovanni.laranjo@coppe.ufrj.br (G.L. de Stefani).

https://doi.org/10.1016/j.nucengdes.2023.112713
Received 7 August 2023; Received in revised form 6 October 2023; Accepted 24 October 2023
Available online 31 October 2023

with the good results already proven in the literature, make this algorithm a good tool for solving LPOP (Pereira et al., 2007; Waintraub, 2009; Yadav R.S.; Gupta, H.P. Optimization studies of fuel loading pattern for a typical Pressurized Water Reactor (PWR) using particle swarm method, 2011; Khoshahval et al., 2010; Ahmad and Ahmad, 2018) and other problems related to nuclear engineering (Marguilis and Shwageraus, 2021; Liu et al., 2014; Almeida et al., 2019).

The present work solves the LPOP for the IEA-R1 reactor core, using the PSO as the main tool. The objective function to be analyzed sought to maximize the effective multiplication factor (k_{EFF}) and the neutron flux, as well as to minimize the Power Peaking Factor (PPF) and ensure that the latter is as uniform as possible. In the next section the model of the problem is discussed, as well as the methodology used to solve it and its particularities in relation to other similar works. In the third section, the validation of the PSO using the sphere function, the fitness analysis and the comparison of the best solution found with an existing configuration of the reactor determined by the IEA-R1 calculation team, are shown. Finally, in the section four shows the final conclusions and suggestions for further work.

2. Materials and methods

2.1. Model of the problem

The IEA-R1, a water-moderated and cooled open pool research reactor, serves multiple purposes, such as radioisotope production for health and industry as well as research with activated sources and neutron beans. The IEA-R1′s remarkable versatility establishes it as a cornerstone of nuclear research and application in Brazil. RODRIGUES, A. C. I.; Tufic Madi Filho; SILVA, D. G. (RODRIGUES, A. C. I.; Tufic Madi Filho; SILVA, D. G. (2017). STUDY AND PROJECT OF THE NEW RACK WITH BORON FOR STORAGE OF FUEL ELEMENTS BURNED IN THE IEA-R1 RESEARCH REACTOR. In: International Nuclear Atlantic Conference - INAC, 2017).

IEA-R1 core operates at a maximum rated power of 5 MW and has the shape of a parallelepiped in a 5x5 arrangement, consisting of fuel elements (FE), surrounded by irradiation elements devices and reflectors. The core is assembled on a grid plate, which is suspended by an aluminum structure coupled to a movable bridge. Fig. 1 shows a representation of a reactor core configuration. It's can be noted that in the central position of fuel elements assembly, in black, there are two channels for high neutron flux irradiation; in red the control and security bars. The highlighted number serves as the identifier for the fuel assembly's placement within the core, while the burnup is located in the lower left corner, and the position on the grid plate is indicated in the lower right corner.

The LPOP resolution for the IEA-R1 aims to optimize the configuration by an objective function (f), which primarily focuses on maximizing k_{EFF} and neutron flux (φ), while simultaneously minimizing PPF and flux variance (Var). This optimization seeks to achieve the most uniform neutron flux distribution, as defined in equation 1.

$$f = \frac{k_{eff}.\phi}{PFF.Var}10^{-13} \quad (1)$$

The equation 1 was empirically defined according to the analysis parameters. The selection of parameters discussed in the preceding paragraph is justified by the imperative to enhance neutron flux, thus maximizing both k_{eff} and flux. Nevertheless, it is essential for this distribution to remain as stable as possible. Hence, the simultaneous minimization of PFF and variance ensures a smaller discrepancy between the maximum, minimum, and, most importantly, the average values of the neutron flux.

The analysis of k_{eff} is conducted by evaluating reactivity (ρ), which quantifies a nuclear reactor's proximity to criticality. This is expressed

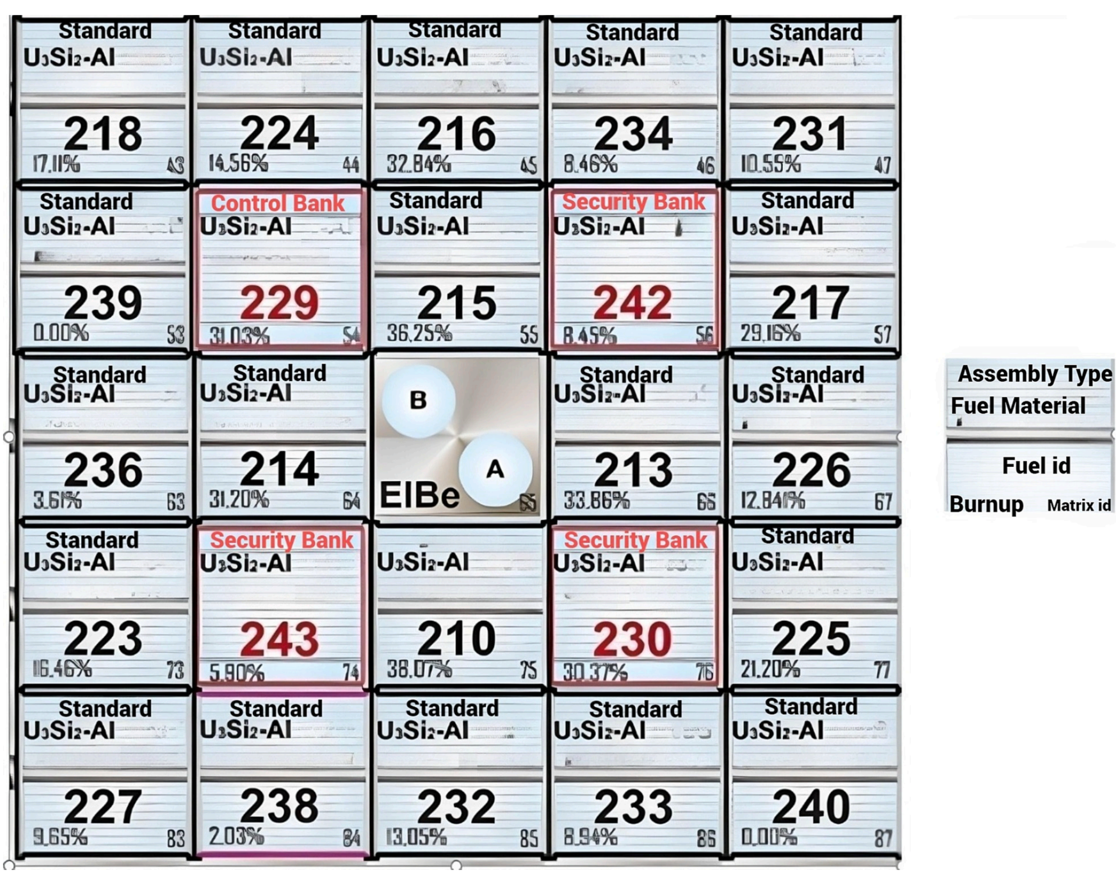

Fig. 1. IEA-R1 reactor core arrangement.

by equation 2, presented below:

$$\rho = \frac{k_{eff} - 1}{k_{eff}} 10^5. \quad (2)$$

The optimization code is seamlessly integrated with the neutronic and thermohydraulic codes employed for assessing the performance and safety of the IEA-R1 during each fuel reload cycle. Specifically, these codes include 2DB (Little and Hardie, 1969), CITATION (Fowler et al., 1972), and COBRA (Chao, j., 1980).

The configuration of the 260th iteration and the novel configurations achieved through PSO optimization using 30 and 50 particles are depicted in Fig. 2.

2.2. Optimization using PSO

In the method, the flock of birds in which the particles are equivalent is represented in a two-dimensional environment. To ensure the existence of a certain individual "personality" of the particles, each one of them is endowed with a distinct and random behavior in which the weighting of these two pieces of information is drawn throughout the process, thus assuming a different value for each one of the particles and varying this value throughout the iterative process. Consequently, as a particle moves within this space, its position is characterized by its coordinates.

Thus, the vector $\vec{x_i}$ is defined as the position of a particle i (i = 1, 2, …,n), where this vector can be of any dimension. The position variation of particle i is defined from the velocity vector, $\vec{v_i}$. The position and velocity vectors are defined as:

$$\vec{v_i}(t) = \omega \vec{v_i}(t-1) + c_1 r_1 (\vec{p_i} - \vec{x_i}(t-1)) + c_2 r_2 (\vec{p_g} - \vec{x_i}(t-1)) \quad (3)$$

$$\vec{x_i}(t) = \vec{x_i}(t-1) + \vec{v_i}(t) \quad (4)$$

where t is defined as a unit time step and hence t-1 denotes the previous step.

The new position of each particle is defined from the sum of three auxiliary vectors. The first one is the velocity vector, while the other two vectors refer to the best position ever found by the particle and the best position ever found by the swarm, defined respectively by $\vec{p_i}$ and $\vec{p_g}$. Obviously, they all originate at the current position of the particle.

The constants c_i and r_i (i = 1.2) are random and contribute to the particle's self-exploration and to its movement towards the swarm's global displacement, respectively. The inertia term, ω, can influence the exploration capacity of the particles, reducing the chance of premature convergence to a local maximum/minimum point. The interval $0.8 < \omega < 1.2$ is defined as a good balance between local and global searches (Luz et al., 2016). Fig. 3 shows the composition of a new particle position - x(t) - from the other vectors mentioned above.

Thus, each particle will have its position adjusted according to the influence of another better positioned particle. Each one of them has a value that shows how close to the ideal result it is, being determined in several ways depending on the problem in question. Fig. 4 describes how the optimization algorithm works in an integrated way with the codes already used in the IEA-R1.

The initial population was created randomly, with the seed provided to the random number generator algorithm determining a set of configurations. Each individual in this population represents a reactor configuration. If these configurations fail to meet the criteria, they are updated using Eqs. (3) and (4). If the particle positions venture beyond the defined solution space, new particles or even an entirely new population are generated.

Because PSO alone cannot handle the mapping required addressing this issue, an additional method is introduced into the algorithm. In this work, the random keys method (RK) (Bean, 1994) was employed. This method maps a vector composed of real numbers into a solution that

a

	3	4	5	6	7
4	EC-218 12,870	EC-224 9,664	EC-216 27,126	EC-234 3,829	EC-231 6,285
5	EC-214 25,713	ECC-229 24,572	EC-215 29,661	ECC-242 2,630	EC-217 23,537
6	EC-235 0,0	EC-207 40,281	EIBE	EC-206 35,420	EC-213 28,088
7	EC-223 11,572	ECC-243 0,0	EC-210 31,270	ECC-230 23,655	EC-225 15,767
8	EC-227 5,548	204 39,815	EC-232 7,646	EC-233 3,928	EC-226 8,343

b

	3	4	5	6	7
4	EC-235 0,0	204 39,815	EC-216 27,126	EC-217 23,537	EC-234 3,829
5	EC-223 11,572	ECC-229 24,572	EC-215 29,661	ECC-242 2,630	EC-224 9,664
6	EC-231 6,285	EC-207 40,281	EIBE	EC-206 35,420	EC-214 25,713
7	EC-227 5,548	ECC-243 0,0	EC-218 12,870	ECC-230 23,655	EC-225 15,767
8	EC-232 7,646	EC-213 28,088	EC-233 3,928	EC-210 31,270	EC-226 8,343

c

	3	4	5	6	7
4	EC-227 5,548	EC-225 15,767	EC-210 31,270	EC-223 11,572	EC-213 28,088
5	EC-206 35,420	ECC-229 24,572	EC-207 40,281	ECC-242 2,630	EC-235 0,0
6	EC-217 23,537	EC-215 29,661	EIBE	EC-214 25,713	EC-216 27,126
7	EC-218 12,870	ECC-243 0,0	EC-234 3,829	ECC-230 23,655	EC-233 3,928
8	204 39,815	EC-232 7,646	EC-224 9,664	EC-231 6,285	EC-226 8,343

Fuel id
Burnup

Fig. 2. a) Configuration 260 (traditional calculation), b) Configuration 260 with PSO optimization (30 particles) and Configuration 260 with PSO optimization (50 particles).

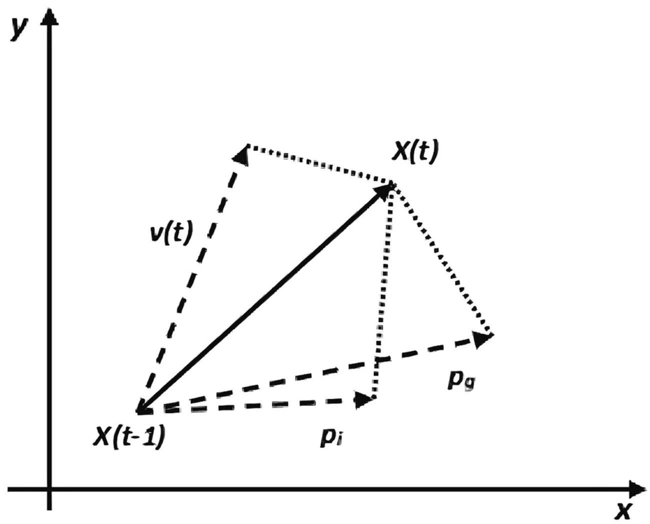

Fig. 3. Representation of the movement of a particle in the PSO.

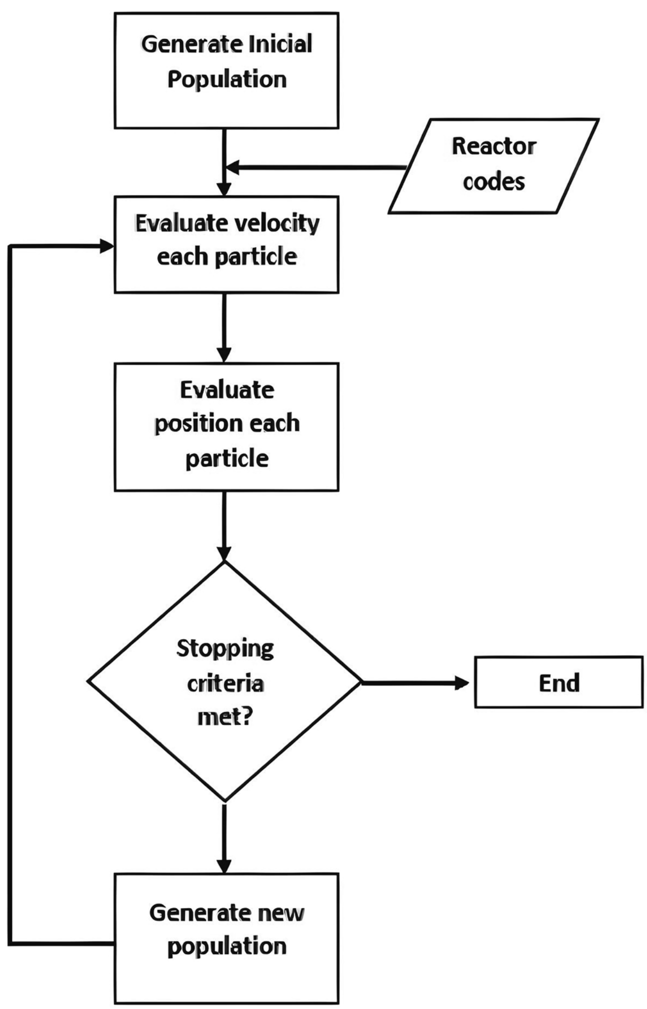

Fig. 4. Flow diagram of the optimization code by PSO.

consists of non-repeating integers. Through this approach, it becomes possible to create feasible solutions for optimization problems. Fig. 5 provides an illustration of how this technique works.

In this work, the RK method plays a crucial role within the PSO algorithm by decoding the position vectors. Its primary function is to transform these vectors into valid solutions that must be subsequently evaluated for fitness. As a result, this method enables the transformation of a continuous search space into a search space containing valid solutions for the problem at hand, which, in turn, facilitates the generation of potential solutions for combinatorial problems.

2.3. Random numbers generator

Random numbers (RNs) are an essential part of any simulation of stochastic processes, and the quality of Artificial Intelligence simulation, whether PSO or another algorithm, relies heavily on the quality and randomness of the random numbers used. To achieve a high degree of randomness, it is essential that random numbers follow a uniform distribution. Therefore, it is necessary to develop methods that can produce sequences of numbers that exhibit randomness, possess a long period before repetition, and do not demand significant computational resources for generation.

In this study, we employ the Algorithmic (or deterministic) method, where we define an algorithm to generate random numbers. This approach, due to its deterministic nature, is commonly referred to as a pseudorandom number generator (PRNG), and the numbers it generates are known as pseudorandom numbers (PRNs). This nomenclature arises from the fact that if you provide the same initial conditions (seed) and define a set of n numbers, you will obtain the same sequence of random numbers each time. The significant advantage of this method is the ability to reproduce computational experiments consistently by using the same seed. This allows for verification of whether the code yields the same results in a given simulation. Therefore, we utilize a congruent generator, the algorithm and explanation of which can be found in (Haghighat, 2021).

The algorithm developed for this study underwent rigorous testing to assess its deterministic randomness. This was achieved by using the configuration described in Fig. 1, with a specified number of particles and interactions, while choosing different seeds. The code was executed twice for each seed. Remarkably, in all 10 tests with 10 different seeds, the algorithm consistently produced the same pairs of configurations.

Additionally, we evaluated the pseudorandom number generator (PRNG) based on its period and average, as detailed in (Haghighat, 2021). The results were then visualized by plotting a 3-tuple distribution of random numbers. As illustrated in Fig. 6, there were no discernible trends or patterns in the generated numbers, indicating that the PRNG exhibited a high degree of randomness and met the desired criteria effectively.

Random numbers (RNs) are a fundamental component of stochastic simulations, including Particle Swarm Optimization (PSO). The quality and effectiveness of artificial intelligence simulations, such as PSO, are heavily reliant on the randomness and quality of the random numbers employed. High-quality random numbers adhere to a uniform distribution, possess a long period before repeating, and can be generated efficiently.

In this study, we adopted a deterministic algorithmic approach known as a pseudorandom number generator (PRNG). PRNGs generate pseudorandom numbers (PRNs) through a predefined algorithm. While PRNGs are deterministic and produce the same sequence of numbers with the same initial conditions (seed), this feature facilitates reproducibility in computational experiments.

We implemented a congruent generator algorithm (as described in (Haghighat, 2021)and assessed its deterministic randomness by conducting multiple simulations with various seeds. Notably, all 10 tests using different seeds consistently yielded identical pairs of configurations, affirming the PRNG's deterministic nature and its capacity for reproducibility.

To assess the quality of the PRNG, we scrutinized its period and average (reference (Haghighat, 2021). The 3-tuple distribution of random numbers, as illustrated in Fig. 6, exhibited no discernible patterns, underscoring the PRNG's suitability for our simulations and its

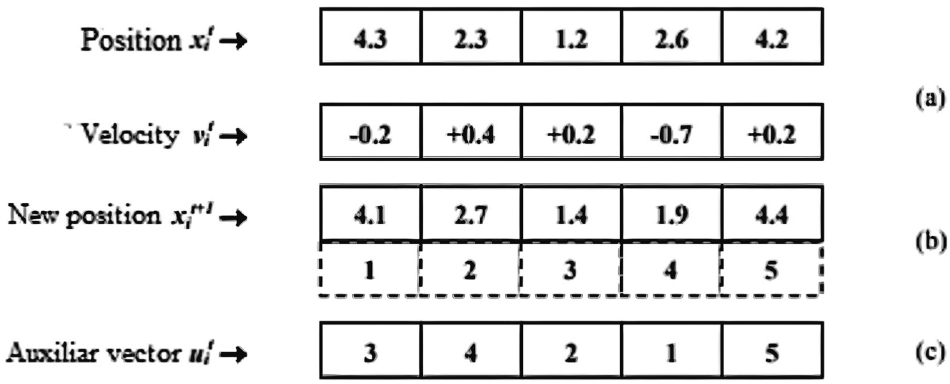

Fig. 5. (a) Position and velocity vectors for a 5-dimensional search space. (b) New positions as random keys. (c) Auxiliary vector containing a valid solution.

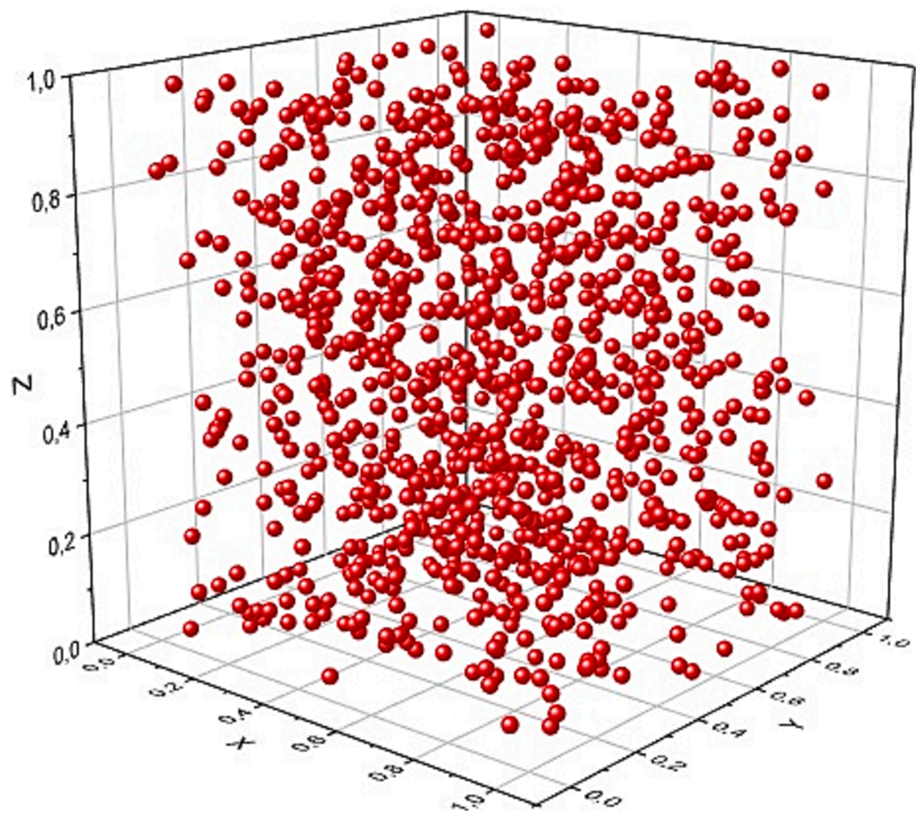

Fig. 6. 3-tuple distribution of random numbers.

capacity to generate truly random numbers.

3. Results and discussions

3.1. PSO validation via a test function

For code validation, we employed the sphere function (Dixon and Szego, 1978), a symmetric, convex, unimodal, and continuous function. This function is versatile as it imposes no restrictions on the number of variables, and its complexity escalates with the increase in the number of parameters under consideration. Mathematically, the sphere function is defined as follows:

$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{4}$$

During this phase of the study, we systematically varied both the number of particles and the number of iterations to evaluate the code's optimization capabilities. The algorithm performed as anticipated, converging toward the origin of the coordinate system as the values of the particles and, more importantly, the number of interactions increased. Simulations were conducted with larger values, even reaching up to 100,000 interactions per particle. It's worth highlighting that increasing the number of iterations had a more substantial impact on improving results compared to increasing the number of particles.

3.2. Fitness analysis

As an initial analysis, our aim was to observe the behavior of the objective function to ensure the convergence of its solutions. To achieve this, we conducted calculations with varying numbers of particles and iterations. Specifically, we examined cases where the number of particles varied, and in others, the number of iterations was adjusted. For this analysis, we considered a set of particles (p) ranging from [20,30,40,50,60,70], and iterations (t) from [500,1000,1500]. These values were chosen with the consideration of computational time, aiming to keep it within a reasonable timeframe for typical computers. Fig. 7 illustrates the behavior of the objective function concerning the number of particles.

Based on Fig. 7, the premature convergence for cases with more particles used is more evident, as mentioned in the literature (Luz et al., 2016). This justifies the use of data between 20 and 50 particles, since the objective function for 60 and 70 particles is considerably smaller. Thus, Fig. 8 shows the fitness behavior in the interval $20 < p < 50$ as a function of the number of iterations.

It is evident that as the number of iterations increases, regardless of the number of particles used, there is an increase in the fitness function value. However, it's noteworthy that the influence of iterations was more pronounced for p = 50.

Consequently, the practice of evaluating within the interval $20 \leq p \leq 50$ and $500 \leq t \leq 1500$ was adopted, consistently using the same seed for generating the random numbers that initialize the particle population. From this analysis, the best fitness configuration was compared with a configuration previously determined by the IEA-R1 reactor calculation group (as shown in Fig. 2).

As depicted, the objective function yields the best results for 30 and 50 particles, consistently maintaining a direct proportionality to the combination of the number of particles and iterations. This places greater emphasis on increasing the number of iterations, akin to what was observed in the tests with the sphere function.

3.3. LPOP solution

Based on the discussion in section 3.2, the optimal values for the objective function were identified in cases where p = 30; t = 1500 and p = 50; t = 1500.

The obtained solutions were then compared with the 260 configuration of the reactor, which was calculated by the reactor's physics team. This configuration is not only one of the most recent but also involves fewer burning steps, resulting in time savings in the calculations. Table 1 provides a comprehensive comparison of each parameter analyzed by fitness for the cases involving 30 and 50 particles, as well as for the 260 configuration.

While it's true that setting p = 30; t = 1500 yields a slightly higher fitness value compared to p = 50; t = 1500, the latter configuration appears to offer advantages in most parameters. This difference is particularly pronounced when examining the area graphs presented in Fig. 9(a) and (b), which compare the solutions obtained from the optimization code with the 260 configuration (as shown in Fig. 2). These visualizations emphasize the benefits of the p = 50; t = 1500 configuration in various parameters.

It's evident that the area occupied by the calculated configuration is significantly larger than that of the 260 configuration, confirming it as the superior solution, even though it has a slightly lower objective function value than the p = 30, t = 1500 case. Consequently, the configuration with 50 particles emerges as the best result obtained, making it the focus of this part of the study.

The reactivity gain between the configuration determined by the calculation group (configuration 260) and the configuration derived from the optimization code using p = 50 and t = 1500 amounts to −451.73 pcm. Substantial improvements are also observed in PFF gains (6.93 %) and variance (9.62 %). The calculated neutron flux holds a slight advantage over the original configuration (0.45 %).

The power density distributions of the two configurations are visualized in 2D and 3D in Fig. 10. It's important to note that the arrangement of the configurations exhibits some differences, with the peaks in the 260 configuration being more isolated, while those in the calculated configuration coalesce in the central regions, forming a kind of "plateau." Furthermore, concentrating power in the central regions, which is a characteristic of IEA-R1 configurations and is still present in the configurations calculated by the optimization code, results in reduced neutron leakage and, consequently, lower fuel consumption.

Regarding the thermal operational limit of the fuel element, defined as 95 °C, the analyzed configuration not only adheres to these values but also achieves lower temperatures compared to configuration 260, as demonstrated in Fig. 11. The maximum temperature obtained through the optimization code was 88.3 °C, which signifies a reduction of 6.7 °C compared to the data provided by the benchmark configuration. This indicates that the optimized configuration maintains safe thermal conditions within the desired limits.

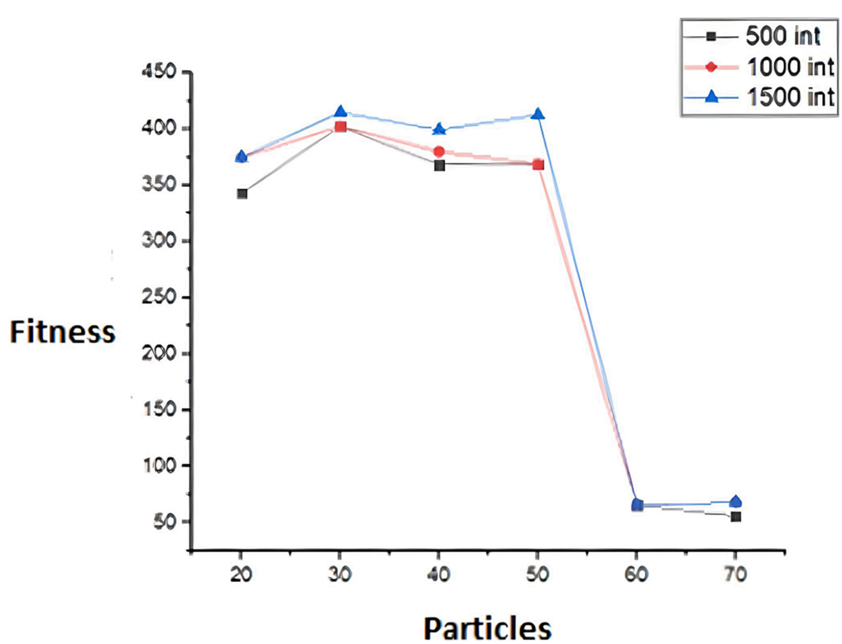

Fig. 7. Behavior of the fitness function in relation to the number of iterations for 20, 30, 40 and 50 particles.

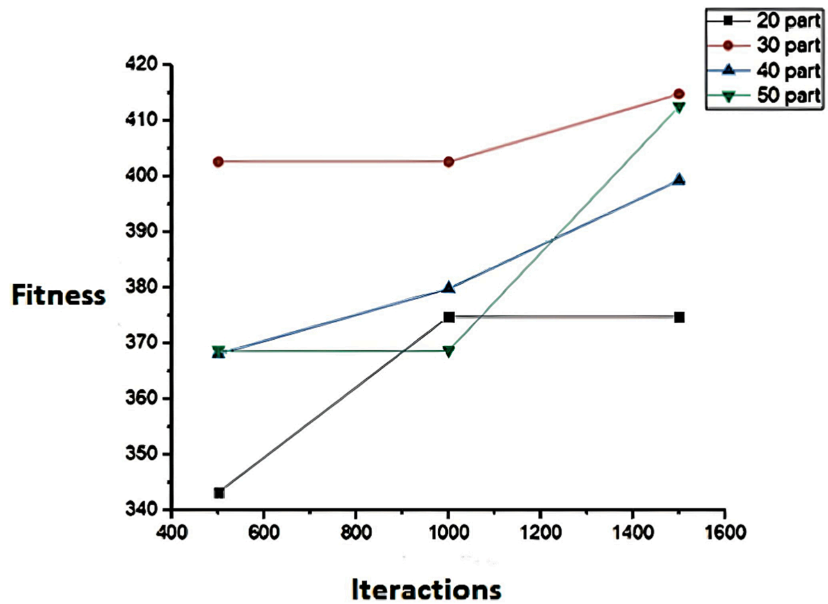

Fig. 8. Behavior of the fitness function in relation to the number of particles for 500, 1000, 1500 iterations.

Table 1
Comparison between the objective function parameters for the cases of 30 and 50 particles and the 260 configuration.

	config. 260	30 particles	50 particles
ρ (BOC)	5916.89	6161.44	5465.15
PFF	1.429	1.539	1.33
Flux (10^{14})	1.1558	1.1594	1.161
Var	0.0343	0.033	0.031

4. Conclusions

Based on the results obtained, it was possible to reach the following conclusions:

- The optimization code effectively solves the POR for IEA-R1, producing configurations that offer advantages over the original configuration.
- While improvements were achieved, it's important to consider the relative importance of different parameters in fitness evaluation. The data suggests that k_{eff} plays a more significant role in fitness than other factors. This is why the p = 50, t = 1500 configuration, despite having a slightly lower objective function value, performs better in terms of other parameters.
- The code demonstrates its effectiveness in finding positioning patterns that differ from the current loading pattern, even leading to the maximum peak being located in the outermost region of the core.
- The best solutions not only meet but sometimes improve thermo-hydraulic safety standards, even with increased neutron flux.
- The development of this code establishes an initial benchmark for solving this type of problem in the context of IEA-R1. There is potential for further optimization using different objective functions that assign weights to various parameters and by employing alternative metaheuristics or hybrid approaches.

In summary, the optimization code represents a valuable tool for enhancing the configuration of IEA-R1, offering improvements in terms of reactor performance while maintaining safety standards. Further refinements and exploration of different optimization approaches may lead to even more favorable results.

Declaration of Competing Interest

The authors declare that they have no known competing financial

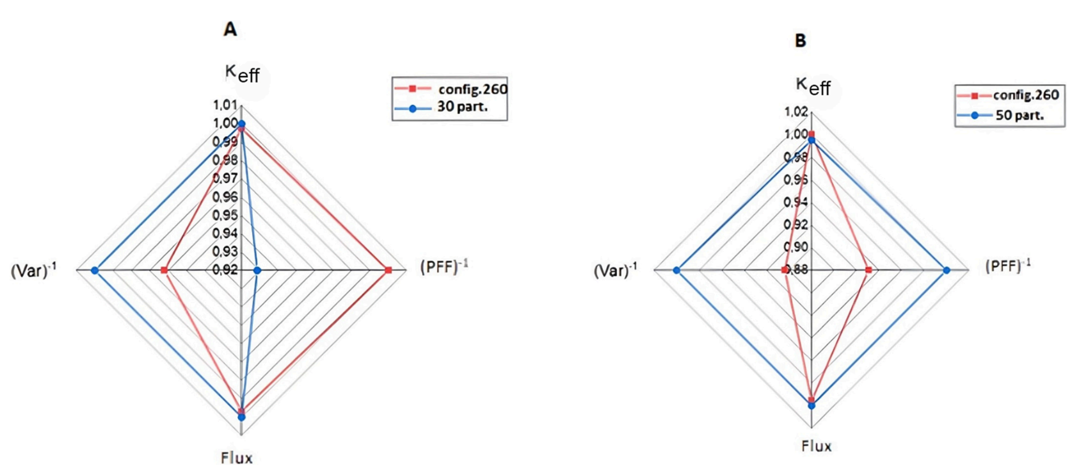

Fig. 9. Comparison by areas between (a) p = 30; t = 1500 and config. 260. (b) p = 50; t = 1500 and config.260.

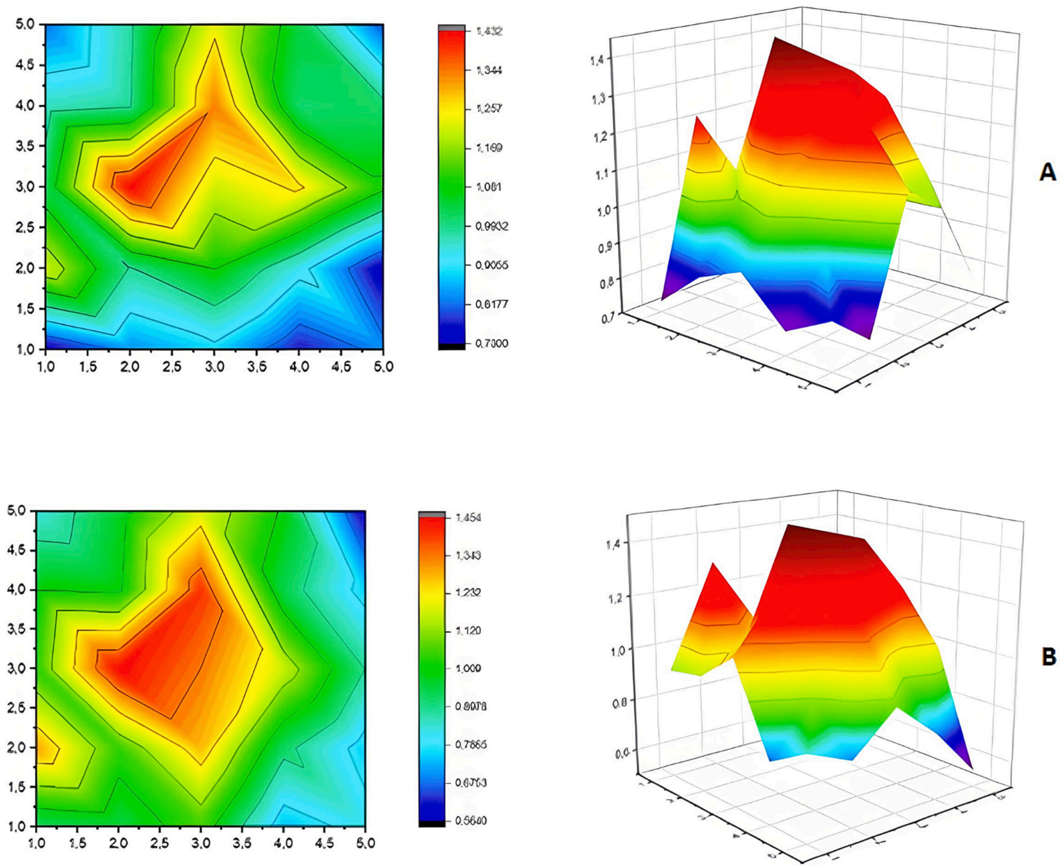

Fig. 10. Power Density Distribution in 2D and 3D for (a) config. 260 (traditional calculation) (b) p = 50; t = 1500 (I.A. optimization).

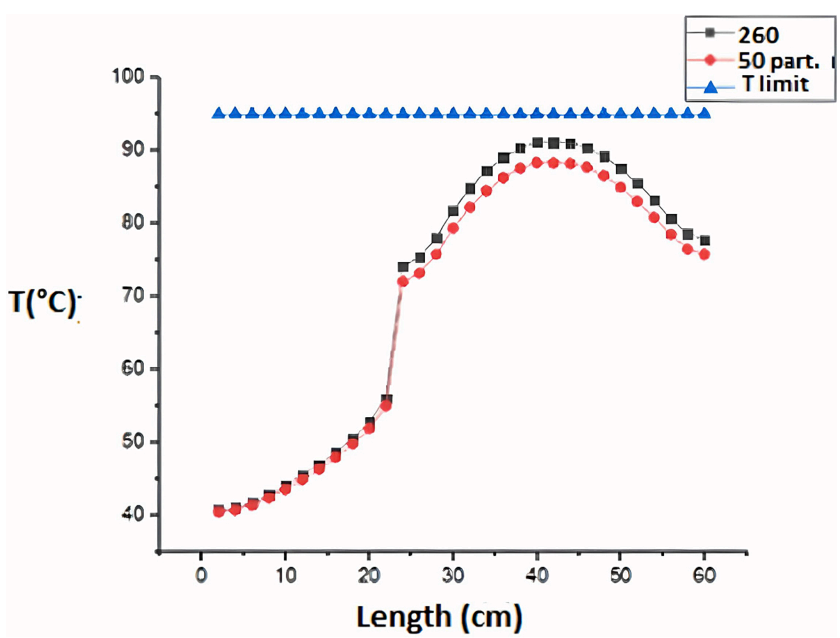

Fig. 11. Power comparison between the temperature distributions in the highest power density EC in the 260 configurations (traditional calculation) and calculated with 30 particles and 1500 iterations and the threshold temperature.

interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Ahmad, A.; Ahmad, S. Optimization of fuel loading pattern for a Material Test Reactor using Swarm Intelligence. (2018) *Prog. Nucl. Energy*. **103**, 45-50.

Akbari, R., Abbasi, M., Faghihi, F., Mirvakili, S.M., Mokhatari, J., 2018. A novel multi-objective optimization method, imperialist competitive algorithm, for fuel loading pattern of nuclear reactors. Prog Nucl. Energy. 108, 391–397.

Almeida, B.S., Leite, V.C.P., swarm optimization: A powerful technique for solve engineering problems., 2019. *In Swarm Intelligence: Recent advances, new perspectives, and applications*; Del Ser, J., Villar, E., Osaba, E., IntechOpen: London. United Kingdom 2, 31–48.

Bean, J.C., 1994. Genetic Algorithms and Random Keys for sequencing and optimization. ORSA J. of Comp. 6, 154–160.

Chao, j., 1980. COBRA-3C/RERTR: A Thermal-hydraulic subchannel code with low pressure capabilities. Argonne National Laboratory, USA.

de Oliveira, I.M., Schirru, R., 2011. Swarm intelligence of artificial bees applied to In-Core Fuel Management Optimization. Ann. Nucl. Energy. 38, 1039–1045.

DeChaine, M.D., Feltus, M.A., 1995. Nuclear fuel management optimization using genetic algorithms. Nucl. Technol. 111, 109–114.

Dixon, L.C.W., Szego, G.P., 1978. The global optimization problem: an introduction. Towards Global Optimization 2, 1–15.

Do, B.Q., Nguyen, L.P., 2007. Application of a genetic algorithm to the fuel reload optimization for a research reactor. Appl. Math. Comput. 187, 977–988.

Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. (1995) *Proceedings of the Sixth International Symposium on MHS'95*, Nagoya, 04-06 October 1995.

Fowler, T.B.; Vondy, D.R.; Cunnigham, G.W. (1972) Nuclear reactor core analysis code: CITATION. Oak Ridge National Laboratory, ORNL-TM-2496, Rev.2, Suppl. 3. Oak Ridge, USA.

Haghighat, A., 2021. Monte Carlo Methods for Particle Transport, 2° Edition. Taylor & Francis Group LLC.

Kashi, S., Minuchehr, H., Poursalehi, N., Zolfaghari, A.B., algorithm for the fuel arrangement optimization of reactor core., 2014. Ann. Nucl. Energy 64, 144–151.

Khoshahval, F., Zolfaghari, A., Minuchehr, M., Sadighi, M., Norouzi, A., 2010. PWR fuel management optimization using continuous particle swarm intelligence. Ann. Nucl. Energy 37, 1263–1271.

Kropaczek, D., Turinsky, P.J., 1991. In-core nuclear fuel management optimization for pressurized water reactors utilizing simulated annealing. Nucl. Technol. 95, 9–32.

Lee, H.C., Shim, H.J., Kim, C.H., 2001. Parallel computing adaptive simulated annealing scheme for fuel assembly loading pattern optimization in pwr. Nucl. Technol. 135, 39–50.

Little, W., Hardie Jr, R.W., 1969. 2DB User's manual – Revision I. In: Battelle Pacific Northwest Laboratory, BNWL. USA, p. - 831..

Liu, C.Y.; Yan, C.Q.; Wang, J.J. Hybrid particle swarm optimization algorithm and its application in nuclear engineering. (2014) *Ann. Nucl. Eng.* 276-286.

Luz, E.P., Becceneri, J.C., Stephany, S., Velho, H.F., Neto, A.J., 2016. Otimização por Enxame de partículas (Particle swarm optimization). In: Neto, A.J. (Ed.), Inteligência Computacional Aplicada a Problemas Inversos De Transferência Radiativa. UERJ, Rio de Janeiro, pp. 139–153.

Machado, L., Shirru, R., 2002. The Ant-Q algorithm applied to the nuclear reload problem. Ann. Nucl. Energy 29, 1455–1470.

Marguilis, M., Shwageraus, E., 2021. Optimization of AGR-Like FHR fuel assembly using multi-objective particle swarm algorithm. J. Nucl. Eng. 2, 35–43.

Mazrou, H., Hamadouche, M., 2006. Development of a supporting tool for an optimal fuel management in research reactors using artificial neural networks. Nucl. Eng. Des. 236, 255–266.

Pereira, C.M.N.A., Schirru, R., Lapa, C.M.F., Medeiros, J.A.C.C., Waintraub, M., Meneses, A.A.M., Baptista, R.P., 2007. Particle swarm optimization applied to a nuclear engineering problems. Int. J. Nuclear Knowledge Manage. (online) 2, 313–332.

RODRIGUES, A. C. I.; Tufic Madi Filho ; SILVA, D. G. (2017). STUDY AND PROJECT OF THE NEW RACK WITH BORON FOR STORAGE OF FUEL ELEMENTS BURNED IN THE IEA-R1 RESEARCH REACTOR. In: International Nuclear Atlantic Conference - INAC 2017, Belo Horizonte - MG, Brasil. Nuclear Energy for National Projects.

Safarzadeh, O., Zolfaghari, A., Norouzi, A., Minuchehr, H., 2011. Loading pattern optimization of PWR reactors using artificial bee colony. Ann. Nucl. Energy 2218–2226.

Tanker, E.; Tanker, A. (1994) Aplication of a genetic algorithm to core reload pattern optimization. *Proceedings of the International conference in reactor physics on reactor computations*, Tel Aviv, 23 January 1994.

Tran, V., Phan, G., Hoang, V., Ha, P., Yamamoto, A., Tran, H., 2021. Evolutionary simulated annealing for fuel loading optimization of VVER-1000 reactor. Ann Nucl. Energy. 151, 107938.

Waintraub, M., 2009. Algoritmos paralelos de otimização por enxame de partículas em problemas nucleares. Federal University of Rio de Janeiro, Rio de Janeiro. Ph.D. Thesis,.

Yadav, R.S., Gupta, H.P., 2011. Optimization studies of fuel loading pattern for a typical Pressurized Water Reactor (PWR) using particle swarm method. Ann. Nucl. Energy 38, 2086–2095.