WARSAW UNIVERSITY OF TECHNOLOGY

DISCIPLINE OF SCIENCE: PHYSICAL SCIENCE

FIELD OF SCIENCE: NATURAL SCIENCE

Ph.D. Thesis

Wojciech Roman Kubiński, M.Sc.

Optimization of the in-core fuel management in a nuclear reactor core using evolutionary algorithms

Supervisors

Dr hab. inż. Krzysztof Fornalski

Dr inż. Rafał Prokopowicz

WARSAW, 2024

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr. hab. inż. Krzysztof Fornalski and Dr. inż. Rafał Prokopowicz, for their readiness to help, patience, and support in the process of preparing this dissertation. Additionally, I would like to thank Prof. dr. hab. Katarzyna Grebieszkow, who supervised me during the first years of my doctoral studies, and provided immense support during my initial steps as a PhD student. Furthermore, I must give special thanks to Dr. inż. Piotr Darnowski from the Institute of Heat Engineering at Warsaw University of Technology, who has always been my mentor, introducing me to the field of nuclear engineering and supporting my development in nuclear energy knowledge as well as directly contributing to the development of parts of the work described in this dissertation. Additionally, I would like to thank MSc. Tomasz Machtyl, Dr. Zuzanna Marcinkowska, and MSc. Gaweł Madejowski from the National Centre for Nuclear Research, for their help while working on the topics related to the MARIA research reactor and their willingness to dispel my doubts. Finally, I would like to thank the team from the DTI Engineering Business Unit at Framatome, particularly Gianluca Giorgi, Dr. Mathieu Segond, and Dr. Xavier Hubert, for their support and the opportunity to carry out an extremely interesting project, which became a part of this dissertation. Moreover, I would like to thank Dr. Jakub Skrzeczkowski from the Mathematical Institute at the University of Oxford, for valuable opinions and reviewing mathematical formulas within the thesis.

Lastly, but most of all, I would like to thank my family and friends, not only for their constant support and belief but most of all for the inspiration and example of hard work, which was the driving force in preparing this dissertation.

The work described in this dissertation was performed in three institutions: the Faculty of Physics at Warsaw University of Technology, Data Science and Applied Mathematics Pole of Framatome, and the National Centre for Nuclear Research. Part of the Research was funded by the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) program under the project “Application of Machine Learning Methods to Optimize the Fuel Campaign of a PWR Nuclear Reactor” (grant number 1820/370/Z01/2021).

Abstract

Optimization of the in-core fuel management in a nuclear reactor core using evolutionary algorithms

This thesis addresses the use of optimization evolutionary algorithms, primarily genetic algorithms, in the optimization of nuclear reactor fuel cycle management. The work discusses three optimization topics – the first core loading of a PWR-type reactor, the equilibrium cycle of a PWR, and the irradiation scheme of target materials for the production of radioisotopes.

In the topic concerning the first core loading, the issue of arranging fuel assemblies in the core of a Westinghouse PWR with is analyzed. The optimization is performed with the aim of extending the cycle, reducing the unevenness of the flux distribution while maintaining the average enrichment of the fuel assembly and keeping the effective neutron multiplication factor within the appropriate range. The second topic pertains to the equilibrium cycle, which is a universal loading scheme applied each time the core is reloaded. The study aimed to maximize the average burnup of the fuel while maintaining or improving selected safety and economic parameters of core operation – the unevenness factor of power distribution, maximum burnup of the assembly, and neutron leakage. The analysis was based on a test model inspired by the core of the Framatome EPR reactor and was carried out during a six-month visit to the company's headquarters. The last of the topics, concerning the optimization of the irradiation scheme of target materials for the production of ^{99}Mo in the core of the Maria research reactor, aimed to find the optimal loading method, based on theoretical analyses and simulations coupled with a genetic algorithm. The goal of the study was to maximize the efficiency of production and minimize the costs of this process.

Each of the topics demonstrates the application of evolutionary algorithms in various issues of nuclear engineering, related to the management of the cycle in a nuclear reactor and the potential gains resulting from their use. The thesis discusses the technical and theoretical aspects of this topic as well as limitations and, above all, the possibilities for further development, especially in terms of expanding the set of optimized parameters and new optimization methods.

Keywords: Optimization, Evolutionary Algorithms, Nuclear Reactor Core, Nuclear Cycle Management, Radioisotopes Production

Streszczenie

Optymalizacja zarządzania cyklem paliwowym w rdzeniu reaktora jądrowego przy użyciu algorytmów ewolucyjnych

Niniejsza praca dotyczy wykorzystania optymalizacyjnych algorytmów ewolucyjnych, przede wszystkim algorytmów genetycznych w optymalizacji zarządzania cyklem paliwowym reaktora jądrowego. W pracy poruszane są trzy tematy optymalizacyjne – pierwszy załadunek rdzenia reaktora typu PWR, cykl równowagowy reaktora PWR oraz schemat napromieniania materiałów tarczowych do produkcji radioizotopów.

W temacie dotyczącym pierwszego załadunku rdzenia analizowane jest zagadnienie ułożenia kaset paliwowych w rdzeniu reaktora PWR firmy Westinghouse. Optymalizacja jest wykonywana pod kątem wydłużenia cyklu, zmniejszenia nierównomierności rozkładu strumienia, przy jednoczesnym zachowaniu średniego wzbogacenia kasety paliwowej oraz utrzymaniu efektywnego współczynnika neutronów w odpowiednim zakresie. Drugi temat dotyczy cyklu równowagowego, czyli uniwersalnego schematu ładowania który stosowany jest za każdym razem gdy rdzeń jest przeładowywany. Badanie ma na celu maksymalizację średniego wypalenia paliwa, przy jednoczesnym utrzymaniu lub polepszeniu wybranych parametrów bezpieczeństwa i ekonomiki pracy rdzenia. Analiza oparta jest na testowym modelu inspirowanym reaktorem EPR firmy Framatome, a zrealizowana była w ramach 6-cio miesięcznej wizyty w siedzibie firmy. Ostatni z tematów, dotyczący optymalizacji schematu napromieniania materiałów tarczowych do produkcji ^{99}Mo w rdzeniu badawczego reaktora Maria w Narodowym Centrum Badań Jądrowych i skupia się poszukiwaniu optymalnej metody załadunku, opartej na analizach teoretycznych oraz symulacjach sprzężonych z algorytmem genetycznym, w celu maksymalizacji wydajności produkcji oraz minimalizacji kosztów tego procesu.

Każdy z tematów demonstruje zastosowanie algorytmów ewolucyjnych w różnych zagadnieniach inżynierii jądrowej, związanych z zarządzeniem cyklem w reaktorze jądrowym oraz potencjalne zyski wynikające z ich zastosowania. W pracy omówione są techniczne oraz teoretyczne aspekty tej tematyki jak również ograniczenia a przede wszystkim możliwości dalszego rozwoju, zwłaszcza pod kątem rozszerzenia zestawu optymalizowanych parametrów oraz nowych metod optymalizacyjnych.

Słowa kluczowe: Optymalizacja, Algorytmy Ewolucyjne, Rdzeń Reaktora Jądrowego, Zarządzanie Cyklem Paliwowym, Produkcja Radioizotopów

Contents

List of Acronyms

BA Burnable Absorber

BC Boron Concentration

BU Burnup

FDH Enthalpy Rise Hot Channel Factor

BEAVRS Benchmark for Evaluation And Validation of Reactor Simulations

BLX Beginning of Life at Xenon Equilibrium (here at 150 MWd/t)

BOC Beginning of the Cycle

EC Equilibrium Cycle

EOC End of the Cycle

FA Fuel Assembly

FF Fitness Function

FFA Fresh Fuel Assembly

ICFM In-Core Fuel Management

LPO Loading Pattern Optimization

MTC Moderator Temperature Coefficient

PPF Power Peaking Factor

PWR Pressurised Water Reactor

EPR European Pressurised Reactor/ Evolutionary Power Reactor

SA Simulated Annealing

SE Shannon Entropy

EFPD Effective Full Power Days

DNB Departure from Nucleate Boiling

GA Genetic Algorithm

PSA Parallel Simulated Annealing

PSA+GA Hybrid method of the Parallel Simulated Annealing and Genetic Algorithm

Overview of the thesis

Scope of the thesis

This thesis describes three optimization problems of the nuclear in-core fuel management problems and is the result of research conducted at three institutions, between the years 2020 and 2024 – at the Faculty of Physics of the Warsaw University of Technology, Data Science and Applied Mathematics Pole of Framatome, and at the Reactor Research Facility of the National Centre for Nuclear Research. The thesis consists of five sections: Introduction, Review of the Fuel Management Optimization studies, Optimization of the Single Cycle Problem, Optimization of the Multi-Cycle Problem, and Optimization of the irradiation scheme. Below is a brief presentation of the contents of each.

Section 1: Introduction, presents a thorough introduction to in-core fuel management (ICFM), emphasizing the importance of optimizing the arrangement of fuel assemblies within nuclear reactors to enhance performance, ensure safety, and maximize fuel utilization. It explores the complexities involved in managing fuel assemblies through various loading patterns, noting the computational challenges posed by the vast number of potential configurations. The chapter underscores the role of advanced optimization algorithms, such as evolutionary algorithms, in refining these loading patterns to improve reactor operations. Additionally, the Section intrduces the foundational principles of nuclear reactor physics, covering critical topics such as neutron transport equations and their resolution through methods like the Monte Carlo and diffusion approximation.

Section 2: Review of the Fuel Management Optimization studies reviews optimization studies in ICFM, focusing on Loading Pattern Optimization (LPO) and various reactor technologies like PWRs, BWRs, research reactors, and Gen IV reactors. It categorizes the issues into single-cycle and multi-cycle problems, using different methods such as evolutionary algorithms and machine learning. It briefly describes how LPO studies have evolved from simple algorithms in the 1950s to complex methods dealing with multi-cycle operations today, highlighting both the historical progress and current trends in fuel management optimization and it organizes current knowledge. The Section also outlines the challenges in applying these

methodologies to different reactor types, stressing the increased complexity in multi-cycle problems and the application of advanced algorithms in recent studies.

Section 3: Optimization of the Single Cycle Problem, describes the first of the discussed problems, that concerns the optimization of the initial fuel loading scheme in the core of a Westinghouse PWR reactor, based on a real American nuclear power plant described in the Benchmark of the MIT Computational Reactor Physics Group [1]. This project was carried out at the Warsaw University of Technology, in cooperation with the Institute of Heat Engineering of WUT and the Faculty of Electronics and Information Technology WUT within the joint project titled "*Application of Machine Learning Methods to Optimize the Fuel Campaign of a PWR Nuclear Reactor*", carried out under the IDUB BEYOND-POB-II grant. In the course of the research, a new adaptive genetic algorithm was implemented to search for the core load optimal in terms of maximizing the cycle length, while maintaining the average enrichment at the reference level, minimizing the power peaking factor, and maintaining the effective multiplication factor of neutrons within a specified range. The paper proposed a new method of variance control, being an original method of selection in the genetic algorithm, yielding better results than the standard methods used. The analysis successfully found solutions characterized by a longer cycle and a more uniform power distribution, demonstrating the correctness of the proposed methods and their potential.

Section 4: Optimization of the Multi-Cycle Problem, concerns the optimization of the Equilibrium Cycle, i.e., a universal loading scheme, exemplified by a reactor core inspired by the EPR reactor, was realized during a 6-months visit to the Framatome headquarters in Paris, which is a leading supplier and operator of nuclear power plants in Europe and worldwide. In cooperation with the Research Department and Data Science and Applied Mathematics Pole, a genetic algorithm and parallel simulated annealing were used to optimize the core shuffling scheme in terms of maximizing burnup while maintaining or improving the unevenness of power distribution, neutron leakage, and maximum fuel assembly burnup. The analysis proposed an extremely simple but original way of presenting the Equilibrium Cycle in matrix form, which significantly simplified the generation and processing of potential schemes. Additionally, in the analyses, the use of Shannon Entropy as a measure to monitor the convergence of the algorithm and its control was proposed, which had not been previously used in cycle optimization

issues. In this part, an important and interesting discussion on core symmetry was undertaken and the advantages and disadvantages of using each were analyzed in terms of potential search for the optimal solution. This part proposed an original hybrid algorithm combining the advantages of efficiency in searching high-dimensional space (known to be a feature of genetic algorithms) and the advantage of avoiding getting stuck in local extremes through broad searching of the search space by parallel simulated annealing. The proposed method presented increased efficiency in selected scenarios, leading to the extension of the cycles by several days, at the same time maintaining or improving other parameters.

Section 5: Optimization of the irradiation scheme, describes the theoretical consideration and application of optimization algorithm for searching for the best irradiation scheme of target materials in the Maria reactor. The study was carried out at the Reactor Research Facility in the National Centre for Nuclear Research and it focused on finding a method of irradiating uranium targets that leads to maximizing the efficiency of the process of producing Mo-99 – a topic extremely important due to the nature of this isotope, being a source of the medical radionuclide Tc-99m, used in 85% of the 30 million medical procedures carried out worldwide annually. The Maria reactor, considered in this work, supplies about 9% of the world's production of this valuable isotope, whose demand is still growing, is greater than the production capabilities. Moreover, cyclic shortages, related to inspections, failures, or the shutdown of old facilities, cause problematic disruptions in radionuclides supplies and availability of medical procedures. Due to the time-consuming and costly installation of new radionuclides-producing facilities, providing more radionuclides is a difficult problem. However, the analysis discussed in this thesis can potentially contribute to some extent to increasing production and partially reducing the problem of Mo-99 availability. The analyses performed showed that optimal irradiation of uranium shields may increase production efficiency and decrease production costs by up to 15%, which is a promising result, worth further continuation.

Role of the thesis

With the growing interest in machine learning methods and artificial intelligence algorithms, topics of intelligent management and optimization are also gaining popularity. At the same time, increasing computational power, unavailable in the past, allows for conducting thousands of simulations, which, combined with an appropriate search strategy, can lead to the efficiency of design processes, improvement of the applied solutions, and even moving beyond the usual schemes to find entirely new, more optimal solutions. The topic of optimization has been addressed in nuclear engineering and applied nuclear physics since the 1970s, alongside the dynamic development of the nuclear sector. However, given the capabilities at the time, this topic was often purely academic contemplation, based on simplified problems, focusing on a narrow range of optimization, and presenting an application of a selected method.

This work serves several functions in its field. Firstly, it fills the gap in the literature, by organizing and presenting the current state of knowledge in the field based on the evaluation of several hundred scientific publications and studies, describing key cases from the very first studies of the 1950s to the latest achievements, in the division into the problem under consideration and the technology analyzed. Furthermore, the work extends the topic of optimization in fuel cycle management to multi-objective optimization of key parameters, providing in-depth analyses of the obtained solutions, applied symmetries, and their influence on the entire cycle and physical phenomena, and also comparing them with actual reference models, validating the correctness of the optimization. Finally, the work discusses topics not only related to the optimization of a given parameter but also includes thorough analyses of the operation of the algorithms, describing their flexibility and potential for nuclear engineering and applied physics applications. It also proposes and examines completely new methods (*Variance Control Method*, application of *Shannon Entropy*, Hybrid Algorithm) that demonstrate improved performance and achieve better results than the classical ones.

In addition, the thesis fits well into the trend and development of the topic, observed since 2021 (see Figure 0.1). It has the potential to significantly contribute to the advancement of the field of engineering and applied nuclear physics, which is increasingly beginning to explore and investigate the possibilities of using machine learning methods, artificial intelligence, and optimization techniques, which in the near future may become the basis for the operation of nuclear systems and conducting research in this field.

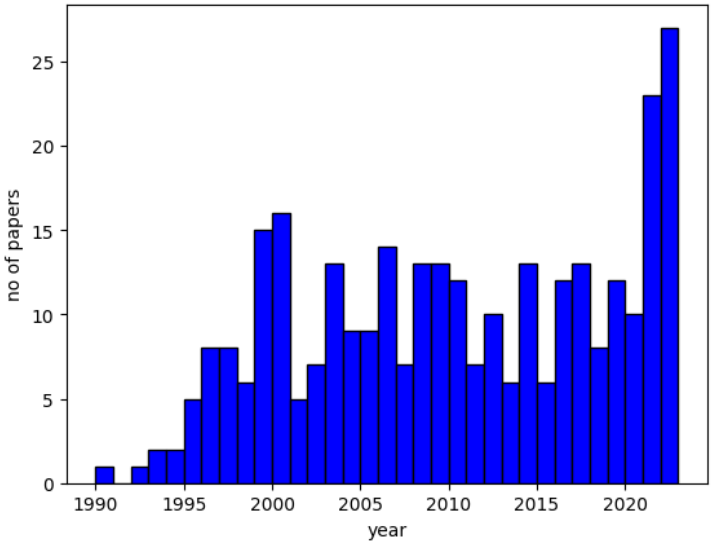


Figure 0.1: The annual number of publications regarding the optimization with genetic algorithms, retrieved from Google Scholar, containing the words "Genetic", "Algorithms", and "Nuclear" in the title, keywords, and/or abstract, manually filtered and validated.

Publications based on the materials presented in the thesis

This thesis summarizes the following journal publications and conference papers, published as part of inter-institutional, and international collaboration:

Journal publications:

1. *Application of novel genetic algorithm in managing the BEAVRS PWR first fuel cycle*, **W. Kubiński**, P. Darnowski, K. Chęć, Annals of Nuclear Energy, 00786, 2020, https://doi.org/10.1016/j.anucene.2021.108153

2. *Application of genetic algorithms in optimization of SFR nuclear reactor design*, W. Żurkowski, P. Sawicki, **W. Kubiński**, P. Darnowski, Nukleonika, 66(4), 2021, https://doi.org/10.2478/nuka-2021-0021

3. *Optimization of the Loading Pattern of the PWR Reactor Core Using Genetic Algorithms and Multi-Purpose Fitness Function*, **W. Kubiński**, P. Darnowski, K. Chęć, Nukleonika, 66(4), 2021, https://doi.org/10.2478/nuka-2021-0022

4. *MARIA Reactor Irradiation Technology Capabilities towards Advanced Applications*, M. Migdał, E. Balcer, ..., **W. Kubiński** et al., Energies, 14(23), 8153, 2021, https://doi.org/10.3390/en14238153

Conference papers:

1. *Optimization of BEAVRS PWR Loading Pattern Using a Novel Genetic Algorithm Based on Population Variance Control*, **W. Kubiński**, P. Darnowski, K. Chęć, 29th International Conference Nuclear Energy for New Europe (NENE), Portoroz, Slovenia, September 7-10, 2020
   https://arhiv.djs.si/proc/nene2020/pdf/NENE2020_0214.pdf

2. *Application of the Artificial Neural Network and Particle Swarm Optimization in determining selected parameters of the nuclear reactor core*, **W. Kubiński**, P. Bojarski, P. Darnowski, European Nuclear Young Generation Forum (ENYGF), 213, 2021,
   http://dx.doi.org/10.13140/RG.2.2.12641.38245

3. *Parallel Simulated Annealing, Genetic Algorithms and Hybrid Method applied to the Multiobjective Optimization of the Nuclear In-Core Fuel Management*, **W. Kubiński**, G. Giorgi, M. Segond, International Conference on Physics of Reactors (PHYSOR), San Francisco, USA, April 21–24, 2024, `https://www.ans.org/meetings/physor2024/`
   https://doi.org/10.13140/RG.2.2.13041.66401

4. *Multiobjective optimization of nuclear fuel multi-cycle management using genetic algorithms and parallel simulated annealing*, **W. Kubiński**, G. Giorgi, M. Segond, Best Estimate Plus Uncertainty (BEPU) Conference, Lucca, Italy, May 19-23, 2024, `https://www.nineeng.com/bepu2024/`
   https://10.13140/RG.2.2.18913.68965

1 Introduction

1.1 Background

1.1.1 In-Core Fuel Management

In-core fuel management (ICFM) is an important part of nuclear reactor operation that pertains to the management and arrangement of fuel assemblies within the reactor core during its operational phase. It involves making precise decisions about the positioning of fuel assemblies to optimize the reactor's performance, maximize fuel utilization, maintain safety margins, and ensure efficient power generation. ICFM involves determining the specific arrangement of fuel assemblies within the reactor for each cycle. This arrangement affects the distribution of heat, neutron flux, and other parameters within the core [2]. ICFM ensures that the core produces the desired amount of energy continuously or within cycles while maintaining safety factors, the inventory of nuclides during the cycle is known, especially at the end of the cycle (EOC), and refueling minimizes the total unit costs of energy production [3].

There are different ICFM strategies, the core shuffling can be designed every time the core is reloaded (single reloading problem) or it can be designed in a multi-cycle manner, by designing several subsequent reloadings or as an equilibrium cycle (EC), being a generic scheme that is applied each time the core is reloaded, needed e.g. for safety and design studies [4]. Due to differences between fuel assemblies (different burnup level, enrichment, burnable absorbers concentration) and depending on which assemblies are removed from the core, there is an enormous number of possible reloading schemes, reaching 10^{30} and more [5]. Assuming only 1 second to evaluate each scheme, checking all possible combinations would take many orders of magnitude more time than the estimated age of the universe. As it is impossible to search the space of solutions one by one, usually the design starts from a similar or previous solution adjusted by expert judgment and a number of different neutronic, thermal-hydraulic, and mechanical analyses. One way to speed up the process, potentially go beyond usually used solutions and extend the ICFM strategies is to use meta-heuristic optimization algorithms such as parallel simulated annealing (PSA) or genetic algorithms (GA) for the loading pattern optimization (LPO) problem.

Loading patterns

Typically, commercial power reactors operate based on cycles, meaning the operation of such a reactor is periodically halted (usually every 1 to 2 years), the reactor vessel is opened, and part of the fuel (usually the most burned assemblies) is removed from the core (usually about 1/3), and new fuel assemblies are inserted in their place. The core is then arranged in a new configuration (loading pattern), the reactor vessel is closed, and it operates again for a given period (called a cycle) [6]. Of course, as mentioned, the core can be arranged in many different ways, each of which leads to different cycle properties (power distribution, potential cycle length, safety coefficients, etc.). Loading a new core or reloading it can be based on four main schemes: placing new fuel in the center of the core and moving the remaining assemblies outward (in-out method), placing fresh assemblies at the edge of the core and moving the remaining assemblies towards the center (out-in), placing fresh fuel assemblies (FFA) at the edge but mixing of the inner assemblies in the core in a checkerboard pattern, in order to reduce peaking factors (Out-In-Checkboard) or by mixing these methods and attempting to find a compromise between power distribution and leakage (3LP - low leakage loading pattern) [7].

Loading pattern in-out (Figure 1.1a), leads to low neutron leakage, as the most reactive assemblies (highly enriched in the case of the first core or fresh fuel in the case of reloading) are located in the very center of the core. On the other hand, this causes uneven flux distribution and a significant power peak, especially at the beginning of the cycle. To minimize this effect, burnable absorbers (BA), like gadolinium [7] are used to some extent. These absorbers capture neutrons at the beginning of the cycle, then gradually absorb them less effectively as they burn out. Thus, BA even out the power distribution and extends the cycle because it allows for a larger reactivity excess, which is gradually released during the cycle. However, the use of BA has its limitations. Neutrons absorbed by the BA are unproductively lost. Additionally, residual poisoning occurs later in the assembly, which is the negative reactivity associated with the products formed from the BAs, potentially penalizing such an assembly in later cycles. Therefore, in general fuel management, efforts are made to use the smallest possible amount of BA [7].

Loading pattern out-in (Figure 1.1b), leads to more uniform power distribution and lower peaking factors. However, due to the highly reactive assemblies positioned at the edge of the core, this layout is associated with greater leakage, particularly of high-energy neutrons that have resulted from fission and have not yet been thermalized in the moderator. This causes the reactor vessel to exposed to a high neutron fluency, which can lead to faster degradation,

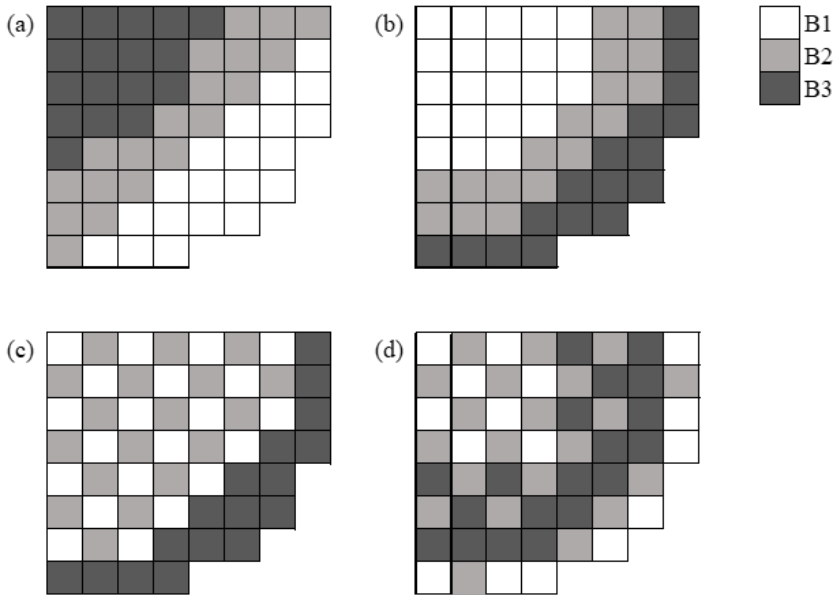


Figure 1.1: Various fuel assembly arrangements in the core: (a) in-out, (b) out-in, (c) out-in-chessboard, and (d) 3LP (low leakage loading pattern). B1, B2, and B3 denote different batches, where B1 assemblies are the least reactive (lowest enrichment in the case of the first core or the highest number of preceding cycles in the case of reloading), B2 assemblies are moderately reactive (medium-enriched or, for example, after one cycle), and B3 assemblies are the most reactive (high enrichment or fresh fuel).

embrittlement, and a shortened lifespan [8].

Loading pattern out-in-chessboard (Figure 1.1c) is an extension of the second method, also involving the loading of fresh assemblies on the edge of the core, but mixing less reactive assemblies (and assemblies with BA) in the inner part to increase burn-up balance. However, this scheme does not reduce neutron leakage.

Low leakage loading pattern - 3LP (Figure 1.1d) seeks a compromise between leakage and even power distribution. This approach essentially combines elements of the other methods, such as utilizing a checkerboard pattern and generally placing more reactive assemblies closer to the core edge, but not at the very periphery, thereby forming a so-called "ring of fire" [9] which provides reactivity to the system. In this type of configuration, the power distribution is relatively flat while neutron leakage is limited. In this configuration, as in all others, BA are also used, particularly arranged in a checkerboard pattern, to limit the initial excess reactivity and even out the power profile.

Each of the mentioned core loading schemes is characterized by different cycle parameters, as described. Some of them lead to more uniform power distribution, and safer and more economical fuel use but increased leakage. Some lead to less leakage but a more uneven power distribution. Besides these parameters, there are many other factors considered in the design of the fuel cycle, such as cycle length, the evolution of boric acid concentration, maximum fuel assembly burnup, critical heat flux, linear power density, enrichment needed, amount of burnable absorbers, nuclear waste production, as well as mechanical properties such as fuel assembly deformation, and bowing, neutron fluency and embrittlement of the vessel and more.

Potential gains of the optimization

Seeking optimal loading schemes can potentially have a significant impact on the safety and economy of nuclear reactor operation. Previous studies suggest that with proper fuel arrangement, it is possible to reduce power peaking factors by several percent [5], thereby increasing the safety margin associated with hot spots and boiling crisis, limit neutron leakage [10, 11], reduce fuel demand, and/or extend the cycle by several days [5, 6, 12]. Assuming a power plant capacity of just 1000 MWe and an electricity price of 0.25 EUR/kWh, extending the cycle by one day only yields a profit of 6 million EUR per unit, of course, reduced by any costs associated with generating this power and the operation of the power plant. Another example can be lowered consumption of fuel. The levelized cost of electricity (LCOE) for nuclear fuel, in the once-through cycle (without reprocessing), was estimated in 2020 at $7/MWh for costs related to mining, enrichment, and conditioning, with an additional approximately $2/MWh for spent fuel removal, disposal, and storage costs [13]. For a power plant with a capacity of 1000 MWe, this gives an annual cost on the order of 78 million USD, assuming continuous operation and no other costs. Considering the production of electrical energy from nuclear power plants in France, which amounted to 360.7 TWh in 2021 [14], reduction of a singlew cent in fuel cost, potentially achieved through better arrangement and lower consumption, corresponds to an additional annual saving of over 3 million USD per year.

These simplified analyses certainly do not account for many factors, including the costs of constructing the power plant itself, loans and financing, and future decommissioning fees. Nonetheless, given the scale of a nuclear power plant and national energy production, they illustrate the order of magnitude of gains and the potential space for optimization that decision-makers face when dealing with the fuel cycle design.

1.1.2 Radionuclides production and their applications

Radioisotopes are integral to a variety of sectors, including medicine, agriculture, industry, and scientific research, playing a key role in national nuclear programs. These programs facilitate the production of radioisotopes, and various radiation sources, significantly enhancing economic sectors such as healthcare and industrial quality assurance. The primary methodologies for synthesizing artificial radioisotopes are through nuclear reactors and particle accelerators. For radioisotope production, nuclear reactors are mostly used, due to their capacity for large-scale irradiation, the ability to irradiate multiple samples at the same time, cost efficiency, and the versatility in the range of radioisotopes they can produce. Particle accelerators are specifically employed to generate isotopes that are either achievable via reactor methods or possess unique properties essential for specific applications [15].

Radionuclides are crucial to medicine and diagnostics, with their use expanding significantly due to advancements in current technology and development of new methodologies. It is estimated that each year, radionuclides are used in over 30 million essential medical procedures worldwide [16], broadly, categorized into diagnostics and therapeutics.

In diagnostics, radionuclides are utilized to invstigate ailments such as heart disease and cancer. This is achieved through the emission of gamma rays from isotopes that are captured by specialised gamma cameras to create detailed body images. Gamma cameras play a crucial role in the diagnostic process for various organs, such as heart, brain, bones, lungs, and thyroid (see Table 1.1). The predominant isotope used in these imaging procedures is Technetium-99m (^{99m}Tc). Additionally, a range of other isotopes are employed, including Thallium-201 (^{201}Tl), Gallium-67 (^{67}Ga), Krypton-81m (^{81m}Kr), Indium-111 (^{111}In), Iodine-123 (^{123}I), Iodine-131 (^{131}I), and Xenon-133 (^{133}Xe).

In therapy, radionuclides that emit alpha particles, Auger electrons, or beta particles are effectively utilized in the treatment of conditions like hyperthyroidism, synovitis, and cancer. Radiofarmaceutics help also to manage pain associated with metastatic cancers. For example, iodine-131 (^{131}I) is used to remove remaining thyroid tissue in treatments of hyperthyroidism or thyroid cancer. Other isotopes like phosphorus-32 (^{32}P), yttrium-90 (^{90}Y), rhenium-188 (^{188}Re), and erbium-169 (^{169}Er) are also employed in the radionuclides treatment [16]. Under development there is also Boron Neutron Capture Therapy (BNCT), a neutron-based technique that enables selective treatment of cancer at the cellular level within tumors. BNCT is particularly effective for treating cancers of the brain, head, neck, and skin. This technique utilizes

the reaction between neutrons and boron to specifically target and destroy cancer cells. Unlike conventional radiotherapy, BNCT offers a fundamentally different approach and holds promise as a more widely adopted option for cancer treatment in the future [17, 18].

Table 1.1: Application of radionuclides in diagnostics [16].

Organ	**Radionuclide**	**Disease**
Lung	^{99m}Tc, ^{133}Xe, ^{81m}Kr	Embolisms, breathing disorders
Bone	^{99m}Tc	Tumours, infection, bone fracture
Thyroid	^{131}I, ^{99m}Tc, ^{123}I	Hyper/hypothyroidism, tumours
Kidney	^{99m}Tc, ^{111}In, ^{131}I	Renal function, embolisms, blood flow, tumours
Brain	^{99m}Tc, ^{123}I, ^{133}Xe	Neurological disorders
Liver, Pancreas	^{99m}Tc, ^{111}In	Tumours
Abdomen	^{99m}Tc, ^{67}Ga	Tumours
Blood	^{99m}Tc, ^{111}In	Infection, blood volume and circulation
Heart	^{99m}Tc, ^{201}Tl, ^{82}Rb	Myocardial function and viability
All	^{67}Ga, ^{99m}Tc, ^{111}In, ^{201}Tl	Tumours

Mo-99/Tc-99 application and production

Technetium-99m-based diagnostic imaging constitutes a significant portion of nuclear medicine practices, with an estimated 85% of all procedures globally utilizing this isotope [19] (see Table 1.1). Nearly all the technetium-99m (Tc-99m) employed in nuclear medicine is generated through the radioactive decay of molybdenum-99 (Mo-99). Molybdenum-99 has a half-life of approximately 66 hours and decays via 88%-branch beta decay that produces Tc-99m. The Nuclear Energy Agency has analyzed the distribution of Mo-99/Tc-99m demand, revealing that mature markets consume about 84% of the global supply, while emerging markets consume the remaining 16%. Projections until 2023 suggest an annual growth rate of 0.5% in mature markets and a more robust growth of 5% in developing markets [19]. The main method for producing this isotope involves the fission of U-235 using enriched uranium targets placed in a neutron field (reactor core). The rate of reaction (effectively the tempo of isotope production) $RR = n \cdot \phi \cdot \sigma$, depends on the number of atoms in the target element (atomic density) n, the

value of the neutron flux density ϕ, and the cross-section σ for the production of Mo-99. The first of these values is determined by the user during sample preparation for irradiation, while the second and third are functions of the position within the core. Both the flux density and the cross-section for a given reaction vary at different locations within the reactor core. The neutron flux density can change by orders of magniture within the core, while the cross-section depends on two factors - the probability of the occurance of the U-235(n, fis) fission reaction (Figure 1.2) and the probability of producing Mo-99 as a result of this fission (Figure 1.3). Both of these factors will depend on the energy of the incident neutron; however, the probability of obtaining a Mo-99 nucleus during fission process (Figure 1.3) is more or less constant, and equals around 6% per fission, for both thermal and fast neutrons [20]. The dependence of the fission cross-section (Figure 1.2) typically follows a characteristic trend of inversely proportional dependence on neutron velocity, $\propto v^{-1}$, from low energies up to thermal energies, then passing through the resonance region for epithermal neutrons, and extending to the unresolved resonances region, where resonances of cross-sections become indistinguishable (transition between 1 and 10 keV).

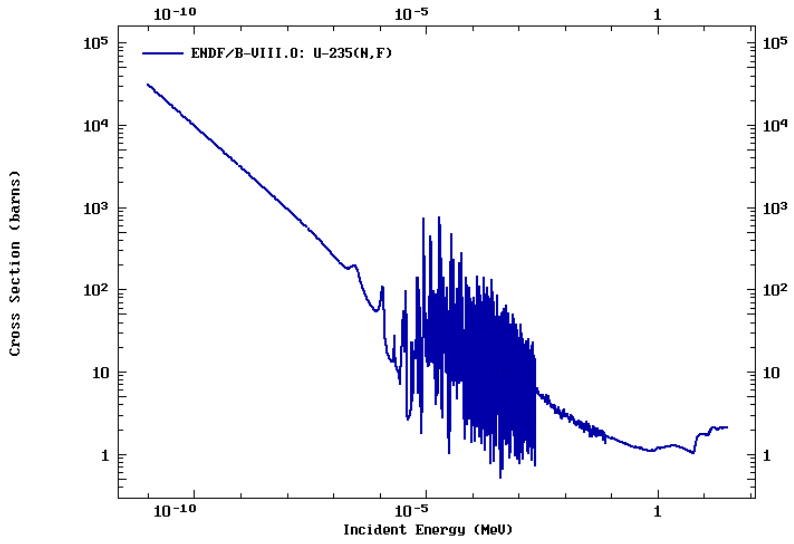


Figure 1.2: Cross section for the fission reaction of $^{235}U(n, fis)$, figure source: ENDF-BVIII.0 nuclear data library [21].

As evident from Figure 1.2, the value of the cross-section, and thus the reaction rate of Mo-99 production, can vary by orders of magnitude, further depending on the flux density value at a given location. Then, the reaction rate can be completely different depending on the location of the sample in the core. Additionally, the actual amount of produced isotope will depend on the duration the sample spends in the core, which, due to spatial limitations,

manouvering and radioactive material handling, can be costly. The overall efficiency of the process and production cost will depend on both the distribution of samples within the core and the proper scheduling of irradiation over time. Even minor modifications causing a shift towards a region with higher cross-section and flux values can significantly increase the production of this element. As mentioned, it is estimated that 85% of diagnostic procedures worldwide, which amounts to about 25 million, utilize Tc-99m. Considering the periodic shortages of Mo-99 observed in healthcare [22–24], seeking optimal production methods is a goal in addressing this gap, and even slight improvements in the process can result in help for hundreds of thousands of patients.

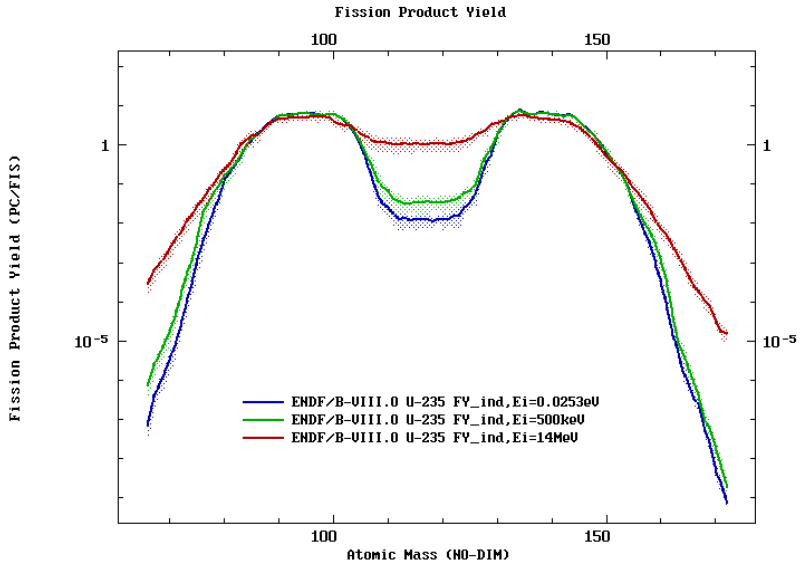


Figure 1.3: ^{235}U fission product yields (with around 6% per fission for ^{99}Mo production), figure source: ENDF-BVIII.0 nuclear data library [21].

1.2 Elements of nuclear reactor physics

1.2.1 Neutron transport equation

The neutron transport equation is the Boltzmann transport equation and is the fundamental formula in nuclear engineering, describing changes in angular flux distribution $\psi(\vec{r},\vec{\Omega}',E',t)$, so the balance of neutrons in space, direction, energy and time [25].

Firstly, neutrons in a nuclear system are produced in the fission process. Neutrons produced directly in this process are called prompt neutrons and their production per unit of time, volume, direction, and energy can be described as:

$$\frac{\chi^p(\vec{r},E)}{4\pi}\left[1-\tilde{\beta}(\vec{r})\right]\int_0^\infty \nu(\vec{r},E')\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',t)dE',$$

where $\chi^p(\vec{r},E)$ is the spectrum of the produced prompt neutrons, $1-\tilde{\beta}(\vec{r})$ is the share of prompt neutrons, $\nu(\vec{r},E')$ is the average number of neutrons produced, $\Sigma_f(\vec{r},E',t)$ is the fission cross-section, and $\phi(\vec{r},E',t)$ is the scalar neutron flux (fission does not depend on the direction).

Neutrons moving through space can be downscattered or upscattered to a given energy and direction, which can be understood as the production of neutrons in a given direction and energy:

$$\int_0^\infty \int_{4\pi} \Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega})\psi(\vec{r},\vec{\Omega}',E',t)\,d\Omega'\,dE',$$

where $\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega})$ is the scattering cross-section responsible for changing energy from E' to E and direction from $\vec{\Omega}'$ to $\vec{\Omega}$.

In the context of the aforementioned reactions (fission, scattering), neutrons with a given energy E and direction $\vec{\Omega}$ are removed from a given point in space, which can be described as:

$$-\Sigma_t(\vec{r},E)\psi(\vec{r},\vec{\Omega},E,t),$$

where $\Sigma_t(\vec{r},E)$ is the total cross-section. Neutrons can also migrate:

$$-\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega},E,t).$$

Finally, neutrons can be produced from precursors, i.e., nuclides that emit neutrons with some delay (in any direction). The term related to this can be written as:

$$\frac{1}{4\pi}\sum_{i=1}^{N_d}\chi_i^d(\vec{r},E)\cdot\lambda_i\cdot C_i(\vec{r},t),$$

where N_d is the number of precursor groups, $\chi_i^d(\vec{r},E)$ is the spectrum of neutrons emitted by the i-th precursor group, λ_i the decay constant of the i-th group, and $C_i(\vec{r},t)$ the concentration of precursors of the i-th group.

Combining all described terms and extending them by the term associated with a potential external neutron source $S(\vec{r},\vec{\Omega},E,t)$, we obtain the integro-differential form of the neutron transport equation:

$$\frac{1}{v(\vec{r},E,t)}\frac{\partial \psi}{\partial t}(\vec{r},\vec{\Omega},E,t)+\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega},E,t)+\Sigma_t(\vec{r},E)\psi(\vec{r},\vec{\Omega},E,t)= \tag{1.1}$$

$$\int_0^\infty\int_{4\pi}\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega})\psi(\vec{r},\vec{\Omega}',E',t)\,d\Omega'\,dE'$$

$$+\frac{\chi^p(\vec{r},E)}{4\pi}\left[1-\tilde{\beta}(\vec{r})\right]\int_0^\infty \nu(\vec{r},E')\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',t)dE'$$

$$+\frac{1}{4\pi}\sum_i^{N_d}\chi_i^d(\vec{r},E)\cdot\lambda_i\cdot C_i(\vec{r},t)+S(\vec{r},\vec{\Omega},E,t).$$

In the case of steady state, time dependencies in Equation 1.1 can be omitted. To simplify analyses and ensure strong equilibrium in the neutron transport equation, a scaling factor k is introduced at the fission term. Additionally, for steady state, the production of neutrons through the i-th group precursors can be written as [25]:

$$\sum_{i=1}^{N_d}\chi_i^d(\vec{r},E)\cdot\tilde{\beta}_i(\vec{r})\int_0^\infty \nu(\vec{r},E')\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',t)dE',$$

where $\tilde{\beta}_i(\vec{r})$ is the fraction of neutrons from the i-th group. Then the steady state neutron transport equation, without an external source, can be written as:

$$\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega},E)+\Sigma_t(\vec{r},E)\psi(\vec{r},\vec{\Omega},E)= \tag{1.2}$$

$$\int_0^\infty\int_{4\pi}\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega})\psi(\vec{r},\vec{\Omega}',E')\,d\Omega'\,dE'$$

$$+\frac{1}{4\pi k}\left(\chi^p(\vec{r},E)\cdot\left[1-\tilde{\beta}(\vec{r})\right]+\sum_{i=1}^{N_d}\chi_i^d(\vec{r},E)\cdot\tilde{\beta}_i(\vec{r})\right)\int_0^\infty \nu(\vec{r},E')\Sigma_f(\vec{r},E')\phi(\vec{r},E')dE'.$$

The above equation is an eigenvalue equation, for which the eigenvector $\psi(\vec{r},\vec{\Omega},E)$ and the eigenvalue k has to be determined. As shown in [25], the largest value of k corresponds to the effective neutron multiplication factor k_{eff}, which is a fundamental parameter describing the state of the nuclear system in which a chain reaction can occur.

1.2.2 Methods for solving the neutron transport equation

An exact solution of the neutron transport equation or set of N_d equations, depending on the number of precurson groups, is possible only in special cases, therefore approximate methods are usually used. There are various methods of this type. Below, the most popular are be presented [25, 26].

Spherical Harmonics Method (P_N Method)

The method consists of representing the neutron flux density in the form of a series of Legendre polynomials (P_m):

$$\psi(\vec{r},E,t,\Omega) = \sum_{m=0}^{\infty} \frac{2m+1}{4\pi} \phi_m(\vec{r},E,t) P_m(\mu), \tag{1.3}$$

$$\phi_m(r,E,t) = \int \psi(r,E,t,\Omega) P_m(\mu) d\mu, \tag{1.4}$$

where $\mu = \cos(\Omega)$, and for Legendre polynomial:

$$P_m(x) = \frac{1}{2^m m!} \frac{d^m}{dx^m}(x^2-1)^m, \text{ for } m = 1,2,\ldots \tag{1.5}$$

In particular, $P_0(x) = 1, P_1(x) = x$. Legendre polynomials are orthogonal and satisfy the relation:

$$\int_{-1}^{1} P_m(\mu) P_n(\mu) \mu \, d\mu = \frac{2}{2n+1} \delta_{mn}, \tag{1.6}$$

where δ_{mn} is the Kronecker delta. The first two expansions of the Legendre polynomials allow for a straightforward physical interpretation: for $m = 0$,

$$\phi_0 = \psi(r,E,t), \tag{1.7}$$

is the total flux density, while for $m = 1$,

$$\phi_1 = 2\pi \int_{-1}^{1} \mu \, \psi(\vec{r},\mu,E,t) \, d\mu = J(\vec{r},E,t), \tag{1.8}$$

is the current in the direction of $\vec{r}$.

Diffusion Approximation

Limiting the spherical harmonics method to the first two terms of the expansion of Legendre polynomials leads to the approximation of the transport equation by the classical diffusion equation. Knowing the particle current in Equation 1.8, **Fick's law** can be written as:

$$J(r,E,t) = -D(r,E,t) \nabla \phi(r,E,t), \tag{1.9}$$

where D is the diffusion coefficient. Additionally, using G energy groups and the group-

averaged values of reaction cross-sections, we can write the **multi-group diffusion equation** [25]:

$$\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\vec{r},t) = \nabla\cdot[D_g(\vec{r},t)\nabla\phi_g(\vec{r},t)] + \sum_{g'=1}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\phi_{g'}(\vec{r},t) \tag{1.10}$$

$$+\left[1-\tilde{\beta}(\vec{r})\right]\chi_g^p\sum_{g'=1}^{G}\nu_{g'}(\vec{r})\Sigma_{f,g'}(\vec{r},t)\phi(\vec{r},t)\sum_{i=1}^{N_d}\lambda_i(\vec{r})\cdot\chi_{i,g}^d(\vec{r})\cdot C_{i,g}(\vec{r},t) - \Sigma_{t,g}(\vec{r},t)\phi(\vec{r},t)$$

where $\phi_g(\vec{r},t)$ are scalar flux values for energy groups g, similarly χ_g^p, $C_{i,g}(\vec{r},t)$ are parameters for the given group and $\Sigma_{x,g}(\vec{r},t)$ are the values of reaction cross-sections averaged for group g. In this way, a system of equations is obtained in which we are looking for the values of scalar fluxes $\phi_g(\vec{r},t)$.

Discrete Ordinates Method (S_N Method)

The discrete ordinates method, also known as the S_N method, is based on replacing the angular variable, Ω of the angular neutron flux $\psi(\vec{r},E,\Omega,t)$ with a finite set of discrete values, Ω_m, called discrete ordinates. It requires integrating the neutron transport equation over homogeneous regions. The regions are defined by dividing the unit sphere into N Sections parallel to the equator, then, each region is further divided into sectors in $4,8,...,2N-4,2N,2N,2N-4,...,8,4$, manner, resulting in a total of $N(N+2)$ sectors. This method significantly simplifies the equation, but it can lead to the so-called "*ray effect*" (called after visible ray-like paths) which is the excessive discretization of the space further away from the center of the unit sphere [25].

Method of characteristics (MOC)

The method of characteristics is based on solving the neutron transport equation along specific paths, known as characteristics, for which the flux is assumed to be constant [25].

Monte Carlo (MC) method

The Monte Carlo method determines the parameters of a system based on data collected from representative sampling. For nuclear system and neutron transport, it involves simulating and tracking the behavior of a given number of indicudal particles under specified conditions. The simulation starts from emission of the particle from a source, and simulating its history by sampling reactions like scattering and absorption, based on the probabilities disribed by scross-

sections from nuclear data libraries (e.g. ENDF [21], TENDL [27]). The libraries are based on experiments, measurements and complicated evalutaions. First, using appropriate numerical codes (e.g. MCNP [28], Serpent2 [29]), geometry, materials and their properties are defined. Then, for each simulated particle, the code draws its free paths and reactions that occur, and repeats this proces until the particle is absorbed or escaped the system. Below, the basic ideas underlying Monte Carlo codes are briefly presented [29, 30].

- Sampling of the neutron free path: The probability that a neutron will interact with a nucleus during its travel through a small distance dx is given by $dP = \Sigma_t dx$, where Σ_t is the total macroscopic cross section. If we define $P_0(x)$ as the probability that a neutron travels a distance x without any interaction with the medium, then the reduction of this probability is equal to:

$$dP_0 = -P_0(x)dP = -P_0(x)\Sigma_t dx.$$

Solving this equation, we obtain:

$$P_0(x) = e^{-\Sigma_t x}.$$

The probability that a neutron will travel a distance x and then interact with a nucleus in the next distance of dx is:

$$P_0(x)dP = P_0(x)\Sigma_t dx = \Sigma_t e^{-\Sigma_t x},$$

thus, the probability distribution of the free path is:

$$f(x) = \Sigma_t e^{-\Sigma_t x}.$$

Knowing the probability distribution, we can sample the free path lengths of neutrons using a pseudorandom number generator and, for example, the inverse transform method: The cumulative distribution function (CDF) has the form:

$$F(x) = \int_0^x \Sigma_t e^{-\Sigma_t \tilde{x}} d\tilde{x} = 1 - e^{-\Sigma_t x}.$$

Inverting this, we obtain:

$$x = -\frac{\ln(1-\xi)}{\Sigma_t}, \quad \xi \in (0,1).$$

Since sampling from the uniform distribution of numbers $1-\xi$ and ξ is equivalent, we can write:

$$x = -\frac{\ln(\xi)}{\Sigma_t}.$$

- Sampling of reaction type: If we denote $\pi_i(E)$ as the probability of the i-th reaction occurring as a result of a neutron of energy E colliding with a nucleus, by drawing a number $\xi \in (0,1)$ from a uniform distribution and using the condition [30]:

$$\sum_{i=1}^{s} \pi_i(E) \leq \xi < \sum_{i=1}^{s+1} \pi_i(E).$$

we can determine that reaction s has occurred.

- Behavior after collision: Depending on the type of reaction, the behavior of the neutron is sampled differently. For instance:

In the case of fission, based on experimental data, it is determined how many delayed and prompt neutrons are emitted. Since the emission time of delayed neutrons has an exponential distribution, its sampling is carried out similarly to the free path:

$$t = -\frac{\ln(\xi)}{\lambda_j}.$$

The energy of emitted neutrons is sampled from distributions available in tables based on experimental data. In the case of scattering, which can be either elastic or inelastic, we apply the conservation laws of energy and momentum, from which the post-collision velocities can be determined:

$$v_{0c} = \sqrt{v_c^2 + \frac{2AQ}{(A+1)m}}, \quad V_{0c} = \sqrt{V_c^2 + \frac{2Q}{A(A+1)m}}.$$

where v_c and V_c are the velocities of the nucleus and neutron before the collision, respectively, and v_{0c} and V_{0c} are the velocities after the collision. Here, $A = \frac{M}{m}$ is the ratio of the atomic mass of the nucleus to the neutron, and Q is the energy (heat) released from the

system. For elastic scattering $Q = 0$, whereas for inelastic scattering $Q < 0$. The value of Q is generated from a distribution derived from data. In cases where more than one neutron is emitted, there are more variables than equations; therefore, sampling from additional probability distributions is applied. The emission of particles other than neutrons is not considered significant from the perspective of neutron transport and their behavior is not analyzed.

1.2.3 Activity and Reaction Rate

The reaction rate α, expressed in units of inverse time, defines the averaged occurrence frequency of a given nuclear reaction per target nuclide atom. It can also be interpreted as the probability of a reaction occurring on a single atom per unit of time. The reaction rate defined by formula [31]:

$$\alpha = \int_0^\infty \sigma(E) \cdot \phi(E)\, dE. \tag{1.11}$$

Due to the simplicity of its definition, it is commonly used for conducting comparative tests with numerical codes modeling neutron transport under experimental conditions. The reaction rate is the simplest function for numerical reproduction of the response of the measuring system, and can be obtained in the activation method, using activation detector.

On the other hand, activity, which is the number of decays of a given nuclide per second (Bq), is defined as the product of the number of nuclei (N^*) and the decay constant λ [32]:

$$A = \lambda \cdot N^*. \tag{1.12}$$

If the nuclides N^* are created by activation of the target nuclides N_a, by irradiation with neutron flux $\phi(E,t)$, with corresponding cross-section $\sigma(E)$ that describes probability of activation, the final activity of the sample would be [31]:

$$A = \int_0^\infty \int_0^\infty \lambda \cdot N_a \cdot \sigma(E) \cdot \phi(E,t) \cdot e^{-\lambda t}\, dt\, dE, \tag{1.13}$$

After integration over irradiation time t_r, activity of the sample:

$$A(t_r) = N_a \cdot (1 - e^{-\lambda t_r}) \cdot \alpha, \tag{1.14}$$

and after cooling time of t_c:

$$A = A(t_r) \cdot e^{-\lambda t_c}, \tag{1.15}$$

By transforming Equation 1.14 and applying the cooling factor we can write a formula for determination of the reaction rate for known activity, and times t_r and t_c:

$$\alpha = \frac{A}{N_a \cdot (1 - e^{-\lambda t_r}) e^{-\lambda t_c}}, \tag{1.16}$$

where the number of target nuclide atoms is given by:

$$N_a = N_m \cdot m, \tag{1.17}$$

where m is the detector mass, N_m is the number of detector's nuclei per unit mass. Finally we can write:

$$\alpha = \frac{A}{N_m \cdot m \cdot \eta}, \tag{1.18}$$

where $\eta = (1 - e^{-\lambda t_r}) e^{-\lambda t_c}$ is a time coefficient that depends on the irradiation and cooling times. The reaction rate uncertainty can be estimated with the formula [32]:

$$\left(\frac{\Delta\alpha}{\alpha}\right)^2 = \left(\frac{\Delta A}{A}\right)^2 + \left(\frac{\Delta m}{m}\right)^2 + \left(\frac{\Delta\eta}{\eta}\right)^2. \tag{1.19}$$

In case of the activation measurements at the MARIA reactor, the activity uncertainty directly comes from activity measurements. The masses of the detectors are small and are measured on an analytical balance with precision $\Delta m = 0.01\ mg$ The source of uncertainties in the term η are the times: activation $\Delta t_r = 30\ s$ and cooling $\Delta\, t_c = 600\ s$. Decay constants λ are known with very high accuracy and can be neglected [32].

1.2.4 Selected parameters of nuclear reactor core operation

The section below briefly describes selected, most relevant parameters and issues of reactor physics discussed in further parts of the work.

Reactivity

Reactivity of a nuclear system is a measure of the deviation of this system from criticality. In general, it a quantitative description of the ability of a nuclear system to sustain a chain reaction of nuclear fission [33]. Criticality, expressed by the multiplication factor k, indicates the state of a system: critical with $k = 1$, subcritical $k < 1$ and supercritical $k > 1$. In the simples definition k can be understand as a relative change in neutron population in the system. It is

can be also interpreted as the higest eigenvalue of the steady-state neutron transport equation (Equation 1.2) [25]. Reactivity (ρ) is relative deviation of k from unity:

$$\rho = \frac{k-1}{k}. \tag{1.20}$$

This value is expressed in units of percent delta $\Delta k/k$ or often in dollars (\$), where one dollar is the value of the share of delayed neutrons. Reactivity control plays a crucial role in reactor core operation.

Power Peaking Factors

The Power Peaking Factor (PPF) is defined as the ratio of the maximum heat flux q'' in the core to the average heat flux in the core. It is the parameter that describes quantitatively the uniformity of the heat sources (also neutron flux) in the core. It can be divided into radial and axial parts:

$$P_{xyz} = P_{xy} \cdot P_z, \tag{1.21}$$

where P_{xy} is a ratio of the average heat flux of the hot channel to the average heat flux of all channels:

$$P_{xy} = \frac{\int_H q''(r_{HC})dz}{\frac{1}{N_c}\int_H q''(r)dz}, \tag{1.22}$$

and P_z is the ratio of the maximal heat flux of the hot channel to the average heat flax of the hot channel:

$$P_z = \frac{max[q''(r_{HC})]}{\frac{1}{H}\int_H q''(r)dz}, \tag{1.23}$$

then, P_{xyz} is can be expressed as:

$$P_{xyz} = \frac{max[q''(r_{HC})]}{\frac{1}{HN_c}\sum_{N_c}\int_H q''(r)dz}, \tag{1.24}$$

where N_C is the number of cooling channels, H is the active height of the core, and r_{HC} is the location of the hot channel. The hot channel is defined as the channel with the highest heat flux and enthalpy rise [34].

Hot Channel Enthalpy Rise Factor

Hot Channel Enthalpy Rise Factor $F_{\Delta H}$ is another parameter describing the non-uniformity of

the power distribution, taking into account not only the power output but also the coolant flow and conditions prevailing in the core [34]. It is defined as the ratio of the maximum enthalpy rise, that is, in the hot channel, to the average enthalpy rise per channel:

$$F_{\Delta H} = \frac{\Delta H_{\text{hot}}}{\Delta H_{\text{avg}}}. \tag{1.25}$$

$F_{\Delta H}$ is related to the total heat power in the channel and the mass-flow rate of the coolant passing through the channel and is an indicator showing how far one is from boiling [7].

Neutron Leakage

In this study, the leakage of neutrons is also considered in the form of a percentage deviation of k_{eff} from the value k_∞, which is the percentage difference in the effective neutron multiplication factor between an infinite system and a real system:

$$L = \frac{k_\infty - k_{eff}}{k_\infty}. \tag{1.26}$$

This value can be derived directly from the six factor formula [33], which is a fundamental equation in nuclear engineering that defines the neutron life cycle in a nuclear system and effecive neutron multiplication factor:

$$k_{eff} = \eta \cdot f \cdot p \cdot \varepsilon \cdot P_{\text{FNL}} \cdot P_{\text{T}}, \tag{1.27}$$

where:

- η - is the reproduction factor (the average number of neutrons produced per absorbed),
- f - is the thermal utilization factor (the ratio of the number of thermal neutrons absorbed in the fuel to those absorbed in all materials),
- p - is the resonance escape probability (probability of avoiding absorption in the resonance region, before thermalization),
- ε - is the fast fission factor (the number of fast neutrons to the number of neutrons from thermal fission),
- P_{FNL} - is the probability that a fast neutron will not escape the system,
- P_{T} - is the probability that a thermal neutron will not escape the system.

Because $k_\infty = \eta \cdot f \cdot p \cdot \varepsilon$ [33], leakage in the definition in Equation 1.26 corresponds to:

$$L = 1 - P_{\mathrm{FNL}} \cdot P_{\mathrm{T}}, \tag{1.28}$$

so the probability that a neutron (neither thermal nor fast) will escape the system.

1.3 Optimization Algorithms

1.3.1 Evolutionary algorithms

Evolutionary Algorithms are meta-heuristic, black-box optimization methods that draw inspiration from biology and the theory of evolution [35]. Next to Differential Evolution, Genetic Programming, Differential Evolution, Genetic Programming, and Evolutionary Strategies, the most representative of this group of algorithms are Genetic Algorithms, which are the focus of this work.

In **Genetic Algorithms (GA)**, the optimized system is usually coded into a vector (by analogy called chromosome), that consists of numbers or strings (called genes). The set of genes (genotype) reflects directly the optimized system and its parameters to be optimized. The algorithm operates on a set of chromosomes (called population) that is being processed until the best (or close-to-best) solution is found. Despite many versions of the GA, for most of them, the population at each step of the algorithm (generation) is usually being subjected to three main genetic operators: selection (with Fitness Function), crossover, and mutation. Each of the operators is a simple but important mechanism that reflects the natural processes of evolution.

Selection and Fitness Function

Selection imitates the competition in nature, where the strongest specimens are more likely to survive, with Fitness Function (FF) being the way to assess each of them. There are different types of selection. The most basic is the roulette method, in which each specimen is being selected for further processing with probability proportional to the value of its FF [35]. Next to the roulette method, tournament and ranking methods are the most commonly used [36].

Crossover

Crossover reflects breeding and transmission of genetic material (mainly the strongest specimens will create their offspring), thus it determines the direction of the population's evolution.

It is based on the exchange of set of genes between chromosomes (parents). In this operator, a random point or points are selected, defining the subsets of genes that are exchanged between parents to create offspring [35]. There are many versions of the crossover operator, and are based on the different number of exchange points, mapping, ordering or method of parents selection [36]. In the basic version of the crossover, two randomly selected specimens exchange parts of their chromosomes at a random exchange point. For the basic crossover operation $\times_k$, two randomly chosen individuals exchange chromosome parts at random point k:

$$[x_1, x_2, ..., x_k, ..., x_n] \times_k [y_1, y_2, ..., y_k, ..., y_n] \rightarrow [x_1, x_2, ..., y_k, ..., y_n], [y_1, y_2, ..., x_k, ..., x_n]. \quad (1.29)$$

Mutation

The mutation is an operator that mirrors random processes in genetics and allows searching a broader space of possible solutions. The mutation can be based on the random change of the gene or exchange of the gene in the chromosome [36], its probability can be constant or it can change in time [37]. In the simplest version, for the permutation problem, mutation operation $M_{k,l}$ can be a simple swap between elements k and l:

$$M_{k,l}([x_1, , ..., x_k, ..., x_l, ..., x_n]) \rightarrow [x_1, , ..., x_l, ..., x_k, ..., x_n]. \quad (1.30)$$

As GA are heuristic methods, there is never an absolute certainty of finding a global solution (or set of solutions). The algorithm may asymptotically approach the global extremum (but it is possible that it would not achieve it in a finite time/steps), it can be stuck or oscillate around a local optimum or it can jump between equally-good ones. In the case of multi-criteria optimization, the final result is in fact dependent on the form of the Fitness Function and the relation between optimized parameters as they usually are interdependant. Thus, the outcome of the GA operation is the whole family of the solutions (so-called Pareto Front) and a single best solution does not exist [38].

1.3.2 Parallel Simulated Annealing

PSA, a parallelized version of Simulated Annealing (SA), is a heuristic method for seeking the global minimum within a broad search space. It is inspired by the annealing process in materials science, where controlled cooling creates crystals with minimal defects, achieving a state of minimal energy [39]. In classic SA, a solution x_i at step i with energy E_i undergoes

stochastic perturbation through random exchanges, resulting in a new state x_{i+1} with energy E_{i+1}. Acceptance or rejection of the new state is determined by certain probability [39], often defined as:

$$p_A(E_i, E_{i+1}, T_g) = \begin{cases} 1, & \text{when } E_{i+1} < E_i, \\ e^{-(E_{i+1}-E_i)/T_g}, & \text{otherwise.} \end{cases} \tag{1.31}$$

This implies that a worse solution can be accepted [40], with probability $p_A(E_i, E_{i+1}, T_i)$, depending on the energy difference and the temperature parameter T_g. The optimization strategy involves gradually cooling down the system by decreasing T_g, allowing exploration of a larger search space initially and convergence as the temperature approaches zero. This prevents premature convergence and getting stuck at a local extreme. In this study, temperature reduction follows a geometric decrease with the number of generations g: $T_g = T_{\text{init}}/g$.

1.3.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a biology-inspired method of searching for a global extremum with the use of a swarm of agents (particles) distributed over a multidimensional space of solutions. One type of PSO method is the firefly algorithm, which involves the use of particles with characteristics and principles of operation inspired by the behavior of fireflies. In this method, each of the particles (initially randomly placed in the space of solutions) is characterized by a certain brightness (analogically to its biological equivalent). This brightness is proportional to the quality of the solution that characterizes the given particle (its position). There are many possible variants of this algorithm, but in the simplest version, each of the particles moves towards the brightest particle in its neighborhood. It moves at a certain speed (and inertia) through the solution space, and at the same time, it can find a better position and become the brightest particle. In this way, particles search the solutions space, potentially approaching the global optimum [41, 42].

When we consider the k-dimensional solution space, each of the particles can be described as a vector:

$$\hat{\mathbf{x}} = [x_1, x_2, \ldots, x_k].$$

Then the distance between particles i and j can be defined as the Cartesian distance between two points:

$$r_{ij} = \sqrt{\sum_{m=1}^{k} (x_{j,m} - x_{i,m})^2}.$$

Assuming that the position of particle i at time t is $\hat{\mathbf{x}}_i^t$, and particle j is the brightest one in its neighborhood, the position of particle i at time $t+1$ will change as follows [41]:

$$\hat{\mathbf{x}}_i^{t+1} = \hat{\mathbf{x}}_i^t + \beta_0 \cdot e^{-\gamma \cdot r_{ij}} (\hat{\mathbf{x}}_j^t - \hat{\mathbf{x}}_i^t) + \alpha \varepsilon_i^t,$$

where α, β_0 and γ are the parameters of the algorithm, ε_i is the random factor (inertia) of the particle, and r_{ij} is the distance between the particle i and the brightest particle j.

1.3.4 Non-dominated Sorting Genetic Algorithm

The Non-dominated Sorting Genetic Algorithm (NSGA) and its further improved version NSGA-II [43], is based on a genetic algorithm, with a modified selection operator to enhance efficiency for multi-objective optimization. The idea of NSGA-II is based on sorting solutions based on their so-called domination, which means exhibiting better properties in terms of a selected parameter. In other words, a solution is non-dominated if it demonstrates the best properties relative to at least one optimized parameter. In this way, the algorithm operates on entire families of solutions that are equally good, and the solution is a whole set of optimized solutions. An important role in this algorithm is also played by the sorting distance, which promotes the diversity of solutions, helping the algorithm avoid local extremes and converge to a single solution.

1.3.5 Reinforcement Learining

Reinforcement learning (RL) is realized using a Markov Decision Process (MDP) in which the agent moves within the environment and iteratively learns from its actions [44]. At each step t, the agent observes a state $s_t \in S$ and based on policy $\pi(a|s)$, chooses an action $a_t \in A(s_t)$. The policy represents the likelihood of selecting action a at state s. At each step, the agent receives rewards $R_{t+1} \in \mathbb{R}$ and moves to a new state s_{t+1}.

RL algorithms are a wide class of algorithms. Examples can be algorithms based on Deep-Q-Networks (DQN) [45], which approximates the reward, Proximal Policy Optimization (PPO) [46] that uses gradient method promoting stable learning rate, or Soft Actor Critic (SAC) [47], which optimizes a stochastic policy.

2 Review of the Fuel Management Optimization studies

This section describes the main issues of ICFM and related problems. It outlines their key features, challenges, along with the historical studies and current state of research. In this chapter, ICFM issues are presented in division of different types of loading apporach (Single-Cycle and Multi-Cycle) and different technology (PWR, BWR, research reactors, and Generation IV reactors).

2.1 Type of the problem

2.1.1 Loading Pattern Optimization

One of the most common applications of optimization methods such as evolutionary algorithms, for In-Core Fuel Management problems, is Loading Pattern Optimization (LPO). The first appraches to LPO began concurrently with the intensive development of nuclear industry in the 1950s and 1960s. In 1954, analytical considerations were done by Goertzel, on the optimal fuel distribution in the core, leading to the minimization of the critical mass of uranium and the flattening of the neutron flux [48]. In 1968, Wall and Fenech [49] applied dynamic programming aiming at optimization of 30 years fuel campain of 1000 MWe PWR, minimizing the overall unit power generation cost. Tabak, in 1968 [50] minimized usage of U-235 and maximized Pu-239 production, using linear and quadratic programming. A year later, Melice [51] maximized reactivity of the reloading profile with constraint on power-peak factor, using trial and error method, dynamic programming and proposed modified version.

In the 1970s and 1980s, the topic gained popularity, resulting in works based on analytical analyses [52], linear programming [53, 54], dynamic programming [55], perturbation theory [54, 56], and eventually also on heuristic methods [57, 58]. However, due to the technology of the time and limited computational power, these problems were usually simplified, concerning a homogenized core with defined regions, using simplified calculations, focusing on the optimization of one or two parameters. Nonetheless, they definitively demonstrated the potential of optimization methods in nuclear engineering.

In the 1990s, the topic continued to develop and heuristic algorithms gained popularity. Due to the complexity and multidimensional nature of LPO problems, along with increasing computational resources, heuristic algorithms, including simulated annealing [59, 60], genetic algorithms [61–63], and tabu search [64], began to be dominant in the field. Since then, they

have been systematically analyzed in research, consistently demonstrating performance and efficiency that are nowadays, still not easily surpassed, despite the proposals of many other algorithms, including more recent machine learning-based techniques [44, 65].

Generally, LPO can be more precisely divided into the following problems: First Core, where the core is loaded with fresh fuel for the first time, Single Reload, where the assembies are partially refueled and rearranged between two cycles, Multiple Reload where a series of reloads is considered, and finally Equilibrium Cycle - a general loading scheme used every time the core is reloaded. Although the categorization seems very clear, it is not always easy to assign a given study to one of these categories. Often, authors analyze general problems, dividing the core into several zones of more reactive and less reactive fuel assemblies. These can correspond to the First Core problem, where we deal with fresh fuel with varying degrees of enrichment, and the different number of BA. At the same time, this can be interpreted as a Single Reload, where the more reactive and less reactive fuel assemblies correspond to assemblies that have spent more or fewer number of cycles in the core.

The remaining part of the studies is divided into *Single Cycle* problems concerning the First Core and the Single Reload, and *Multi-Cycle* problems concerning Multiple Reloads and Equilibrium Cycle. The goal is to present and compare these issues in a more detailed context, enabling a deeper understanding of trends, issues raised, as well as applied solutions and achievements.

2.1.2 Single Cycle Problems

First Core Loading (FCL) is the simplest of the ICFM problems and involves loading when the core is started for the first time. There is no assembly movement in this problem, only their positioning in the core. The FCL optimization problem can be simply defined as a seeking for the best configuration of a vector containing fuel assemblies FA_i at positions $P_i \in \{FA_1, FA_2, ..., FA_l\}$:

$$[FCL] = [P_1, P_2, ..., P_i, ...P_n]. \tag{2.1}$$

If we consider n fuel assemblies in the core (or selected core symmetry) and l number of available types of fuel assemblies, we obtain l^n possible solutions. For example, considering 1/8 symmetry of the core and about $n = 36$ assemblies to arrange with $l = 6$ different types of FFA, we obtain $6^{36} \approx 10^{28}$ possible solutions.

Single Reload (SR) involves selecting the assemblies that we plan to remove from the core, choosing new ones to replace them, and designing a new scheme composed of the new assemblies and those from the previous cycle. SR problem can be understood as optimizing the mapping of the fuel assemblies FA_i that remain in the core and their new positions $P_i \in \{FA_1, FA_2..., FA_n\}$ in the next cycle, $i \in \{1, n-k\}$, and the k positions $P_j \in \{FFA_1, FFA_2..., FFA_l\}$ of fresh fuel assemblies FFA_j, where $j \in \{n-k+1, n\}$ and can be written as:

$$[SR] = [P_1, P_2, \ldots, P_{n-k}, P_{n-k+1}, \ldots, P_n]. \tag{2.2}$$

In the simplest case, when the inventory of fuel assemblies is fixed, the optimization problem is a a simple permutation of a vector of n integers. But in a general case, considering n assemblies in the core, k as the batch size, i.e., the number of assemblies we replace, and l as the number of possible assemblies, then:

a. Choosing k from n assemblies is a combination $\binom{n}{k}$ and can be done in $\frac{n!}{k!(n-k)!}$ possible ways.

b. Choosing k positions for the new assemblies is again a combination $\binom{n}{k}$ and can be done in $\frac{n!}{k!(n-k)!}$ possible ways.

c. Choosing k assemblies from l types is variations with repetitions and can be done in l^k possible ways.

d. Arranging $n-k$ assemblies from the previous cycle in $n-k$ positions is a standard permutation which can be done in $(n-k)!$ possible ways.

Combining all four steps, we obtain: $\left(\frac{n!}{k!(n-k)!}\right)^2 \cdot (n-k)! \cdot l^k = \left(\frac{n!}{k!}\right)^2 \frac{l^k}{(n-k)!}$ possible configurations. Considering a case similar to the previous one ($n = 36$, and $l = 6$) and a batch size of fresh fuel assemblies (FFA), about 1/3 of the core, which is $k = 12$, we obtain approximately 10^{51} possible reloads.

In the 1990s, the subject of Single Reload was repeatedly raised. Parks (1996) proposed a Non dominated Genetic Algorithm (Section 1.3.4) to address multiobjective optimization for PWR, effectively balancing power peaking, fresh fuel, and BA inventories, demonstrating robust optimal loading patterns [63]. Zhao, Knight, Nissan, and Soper (1998) developed FUELGEN, a system for LPO using GA, which outperformed existing methods in optimizing cycle length and safety [66]. In the early 2000s, Pereira and Lapa (2003, 2008) explored an island parallel GA, based on several parallel realizations of GA that can exchange specimens, achieving

superior results in computational efficiency over standard GA [67, 68]. Kobayashi and Aiyoshi (2003) optimized a BWR LPO using an improved GA that maintained diversity and optimized k_{eff} and PPF [69]. Sacco and Pereira (2006) demonstrated the effectiveness of the Particle Collision Algorithm (inspired by the collision process of particles) and the Great Deluge Algorithm (based on an increasing level of the objective function value that allows accepting a solution), showcasing their potential over traditional methods [70]. Alim, Ivanov, Yilmaz, and Levine (2008) implemented a specialized GA for simultaneous optimization of loading pattern and BA placement [71]. Martín-del-Campo and Palomera-Pérez (2009) applied a flexible GA for LPO in BWR, increasing the energy output compared to the reference core [72]. Zameer et al. (2014) explored LPO for a 300 MWe PWR using SA, GA, and a hybrid of the two. The study focused on flattening the power profile, employing novel crossover and mutation strategies in the GA to overcome limitations observed in SA, such as premature convergence to a local minimum. The hybrid approach significantly improved optimization, demonstrating a balance between computational effort and solution quality [73]. Binh, Huy, and Hai (2014) proposed a binary mixed integer coded GA for the multi-objective LPO of a research reactor. They simultaneously maximized the k_{eff} and minimized the PPF [74]. Rahmani (2017) applied GA to optimize the LP of VVER-1000 reactors, also achieving desired safety and performance thresholds [75]. Li, Huang, Wang, and Ding (2021) developed a hybrid teaching-learning GA, which combined method inspired by the interaction between the teacher and students in a class with GA [76, 77]. So et al. (2021) proposed an adaptive GA for a 1000 MWe PWR. Their methodology adjusted crossover and mutation probabilities based on the fitness functions, aiming to minimize the maximum radial PPF [78]. Shaukat et al. (2021) developed a multiobjective optimization strategy for the PARR-1 research reactor, employing a modified GA coupled with Monte Carlo code [79]. Wan, Lei, and Li (2022) explored an innovative optimization method for LPO using a combination of improved convolutional neural networks and GA. Their method enhanced the evaluation speed and accuracy of reloading patterns [80]. Lima-Reinaldo and François (2023) utilized GA for LPO of the ALLEGRO experimental fast reactor. They introduced specialized crossover operators (Partially Mapped Crossover and Order Crossover), demonstrating a significant improvement in the operational cycle length and overall core performance [81].

2.1.3 Multi Cycle Problems

Single-cycle optimization is straightforward to implement. In practice, it requires generating a vector containing types of fuel (either fresh or partially spent) and translating it into a full core model. However, optimizing one reload can lead to optimizing a given cycle at the expense of another [60, 82] , that is, improving the characteristics of one cycle may worsen the properties of subsequent cycles. It is evident that with a batch size $k \approx \frac{1}{3} \cdot n$, the assemblies placed in cycle N will affect not only that specific cycle but also cycles $N+1$ and $N+2$, and then assemblies placed in cycle $N+1$ will impact cycle $N+2$ and $N+3$, and so on. For example, one can imagine that placing large amounts of BA might help to flatten the power distribution in cycle N, but in cycle $N+1$ it could lead to an undesirable decrease in reactivity due to residual poisoning, which is the accumulation of unwanted by-products from the burning processes of BA that remain in the assembly. Also, a more optimal placement of assemblies that results in higher final burnup of the assemblies may lead to the need for additional reactivity in subsequent cycles [82].

Therefore, to truly optimize a cycle, one must consider it from the perspective of the entire fuel campaign, or at least a certain reload horizon, such as several cycles. In the quest for truly optimal schemes, one problem to consider is the Equilibrium Cycle, which is a universal shuffling scheme, the same scheme that is applied every time the core is reloaded, so it reaches equilibrium after the first $\frac{n}{k}$ cycles, i.e., the same cycle properties every time. However, in practice, many operators of nuclear power plants modify the cycles and adjust them to current requirements or issues that may arise, Single Reload or Multiple-Reload optimization is often used because it is most practical as it answers the question of how to currently (or for the next N cycles) load the core. The Equilibrium Cycle optimization then becomes more of a tool for searching for general optimal patterns that may serve as a basis for reloading, or be used to perform calculations at the design and safety analysis stage.

Equilibrium Cycle (EC) is a general scheme of FA reloading that is applied every time the core is reloaded (Figure 2.1). In contrast to a single reload problem where each fuel assembly can be placed at almost every selected position, EC needs to meet several requirements to avoid too low or too high number of cycles that each assembly stays in the core. As the core follows the same refueling scheme after every cycle, it is necessary to ensure that each assembly:

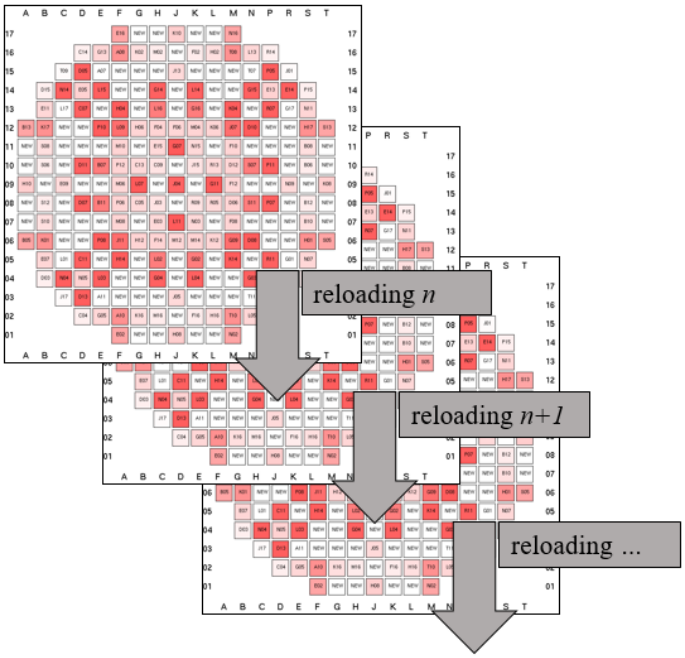


Figure 2.1: General scheme of the Equilibrium Cycle.

a) is not placed in a loop (it comes back to the original position), so it is never removed from the core, eg. it is moved from position *A* to position *B* and then again to *A* then to *B* and so on (Figure 2.2a);

b) does not stay at the same position (so it gets stuck in the core), e.g. it is moved from position *A* to the same position *A* (Figure 2.2b);

c) is not placed in a path that is too long, so it stays in the core for too many cycles e.g. it is moved from position *A* to *B, C, D,* etc. and stays in the core for more than the defined (usually three) cycles (Figure 2.2c);

d) is not exchanged between different symmetry zones, e.g. it is not moved from the diagonal (yellow part in Figure 2.2d) to the octagonal zone (gray part in Figure 2.2d), as the diagonals are mirrored four times in the core whereas 1/8th is mirrored eight times. That would result in the lack of assemblies needed to fill 1/8th. In case of opposite exchange (1/8th to the diagonal), four assemblies would not be used at all while creating the whole core. This also concerns the single reload problem.

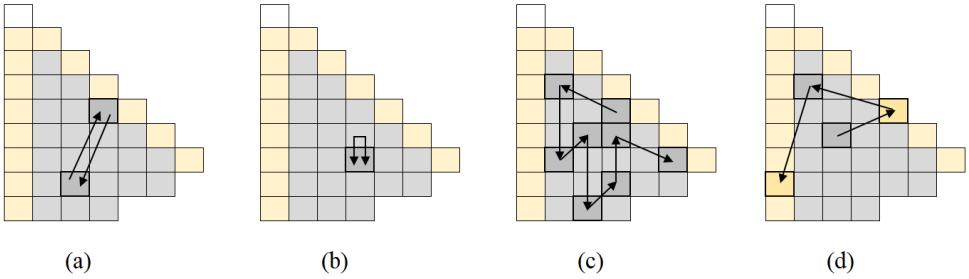


Figure 2.2: Restrictions on the movement of fuel assemblies in the equilibrium cycle: (a) loop, (b) same position, (c) too long path, (d) exchange between different symmetries.

If we consider the batch size of k FFA placed each time in the core and we assume that each FA can stay in the core maximum m cycles, we proposed in this study to represent EC in a very simple way, in the form of a $k \times m$ matrix that consists of paths of fuel assemblies:

$$[EC]^{m\times k} = \begin{bmatrix} P_{FA_1,1} & \dots, & P_{FA_k,1} \\ \vdots & \ddots & \vdots \\ P_{FA_1,m} & \dots, & P_{FA_k,m} \end{bmatrix}^{m\times k} . \tag{2.3}$$

In this problem, we consider that each fuel assembly spends m cycles in the core. Assuming there are n assemblies in the core, the batch size is defined as $k = \frac{n}{m}$. To estimate the number of potential solutions, we assume that each assembly FA_k from batch k is unique. We then need to define k trajectories for each assembly, specifying their positions in successive cycles, denoted as $\{P_{FA_k,1}, P_{FA_k,2}, \dots, P_{FA_k,m}\}$. In this scenario, the equilibrium cycle can be represented by an $m \times k$ matrix containing $m \cdot k$ unique positions. Thus, there is $(m \cdot k)! = (m \cdot \frac{n}{m})! = n!$ possible configurations for the EC. Reconsidering the simplified problem for 1/8 of the core and $n = 36$ assemblies, we obtain approximately 10^{41} possible solutions.

The idea of optimizing the EC originated in the 1970s. Sekimizu (1975) introduced a novel approach to understanding the combined impact of fuel shuffling and control rod programming on equilibrium fuel cycle performance, in a theoretical and deterministic study. Sekimizu applied a three-region cylindrical reactor model and Haling principle [83] to minimize power distribution effects on reactor performance [52]. Yamamoto and Kanda (1997) compared the EC and successive multicycle optimization methods for PWR. Using SA, they found EC optimization to be more computationally efficient and more straightforward compared to successive multicycle optimization [60]. Mahlers (2002) proposed an algorithm combining SA and succes-

sive linear programming to optimize EC for PWR. This approach treated uranium enrichment and BA loading as continuous variables and core reload schemes as discrete variables. The algorithm reduced feed fuel cost by about 2% compared to standard SA [84, 85]. Babazadeh et al. (2009) applied PSO (Section 1.3.3) for LPO in VVER-1000 reactors. They maximized the core k_{eff} while maintaining the local power peaking factor below a defined threshold [86]. Rodrigues et al. (2022) developed GA for searching for the EC in PWR. This method achieved an optimal fuel loading pattern that operated at full power for 240 days with an average discharge burnup of 28.93 MWd/kgHM [4]. Hudayi et al.(2023) used SA to optimize the EC loading pattern for the APR1400 reactor. The study effectively minimized the maximum PPF while meeting the core design constraints [87].

Multiple Reload simply involves considering N successive loading schemes, one after another. Therefore, it requires tp specify a total of n types of FA in each of the N cycles, meaning the arrangement of $n \cdot N$ integers.

$$[MR] = \begin{bmatrix} P_1^{(1)} & P_1^{(2)} & \dots & P_1^{(N)} \\ P_2^{(1)} & P_2^{(2)} & \dots & P_2^{(N)} \\ \dots & \dots & \dots & \dots \\ P_{n-k}^{(1)} & P_{n-k}^{(2)} & \dots & P_{n-k}^{(N)} \\ P_{n-k+1}^{(1)} & P_{n-k+1}^{(2)} & \dots & P_{n-k+1}^{(N)} \\ \dots & \dots & \dots & \dots \\ P_n^{(1)} & P_n^{(2)} & \dots & P_n^{(N)}. \end{bmatrix} \tag{2.4}$$

Therefore, this problem is the Single-Reload from Section 2.1.2 repeated N times. Then the space of possible solutions is the product of the spaces of each of Single-Reload subproblems, so results in $\left[\left(\frac{n!}{k!}\right)^2 \cdot \frac{l^k}{(n-k)!}\right]^N$ possible configurations. For example, for the previously considered case ($n = 36$, $k = 12$, and $l = 6$) and for $N = 3$ considered reloads, we obtain a space of possible solutions of the order of $\sim 10^{154}$. Because of both more complicated implementation and the significantly larger space of possible solutions, the case of multiple-cycle optimization is not often considered in practice.

In one of the first approaches to optimizing fuel cycles, Wall and Fenech [49] in 1965 analyzed the entire fuel campaign of a 1000 MWe PWR reactor, considering approximately 30 consecutive cycles. In their study, they considered fuel arranged in three zones as well as fresh fuel. They used a zonal fuel shuffling approach, placing fresh fuel in a selected zone. Dynamic Programming together with FEVER (one dimensional, four energy group depletion code) were utilized for optimization with the goal of minimizing energy production costs. In 1997, Yamamoto and Kanda [60] compared EC optimization with 10 subsequent cycles for PWR. Their study aimed to maximize cycle length and discharge burnup, focusing on total energy production over the cycles. Each of the optimizations was repeted 3 times and, on average, EC optimization, provided better results than successive multicycle optimization while being also computationally more efficient. Ziver et al. (2003) developed a GA-based optimizer (GAOPT), coupled with the PANTHER nodal code. They applied GAOPT to the multicycle optimization of Advanced Gas-Cooled Reactors (AGRs) at Hinkley Point B and Hartlepool. Their approach considered up to 8 subsequent cycles, optimizing multiple objectives such as minimizing the maximum radial form factor (RFF) and maximizing profit [88]. In a study from 2004, they extended the GA approach by integrating Artificial Neural Networks (ANNs) to speed up the optimization process [88, 89].

2.1.4 Irradiation Scheme

The irradiation scheme is not a typical in-core fuel management problem although it is also an important factor taken into account during planning and decision making for research and production reactors cycles. If we think of the core design or targets configuration optimization in terms of maximizing the isotope production, instead of, e.g. energy production, then, we can easily categorize this problem under in-core fuel management, but with a different goals of optimization.

Due to the diversity of isotope production techniques, it is difficult to establish a typical definition for this problem. Most of the studies that aimed to optimize the production of selected isotope, focused more on the design of the reactor core or the target, not the loading scheme [90–93]. Some studies have focused on optimizing the neutron spectrum and flux by use of shields and filters or adjusting the core configuration [94–96]. Other studies have focused on optimizing the beam and parameters of the accelerator for optimal reaction rate [91, 93].

Meftah et al. (2006) optimized the NUR research reactor core configuration to improve neutron flux for irradiation and ensure operational safety. The new configuration increased use-

ful neutron flux by a factor of up to 10 and reduced irradiation costs while maintaining safety margins [94]. In 2012, Hogle conducted a study on optimizing transuranium isotope production in the High Flux Isotope Reactor. By perturbing the neutron flux spectrum using filter materials, the net consumption of curium and yields of key isotopes were significantly altered. Neural networks and GA were employed to develop an optimization framework for selecting filter materials, enhancing the efficiency of ^{252}Cf production [96]. In 2014, Liem proposed the design optimization of a new aqueous homogeneous reactor (AHR) for the production of Mo-99/Tc-99m medical radioisotopes. The study focused on minimizing the critical ^{235}U mass and maximizing reactor power to enhance production capacity [90]. Hedayat proposed a practical optimization method for refueling programs in research reactors using a modified version of SA. By incorporating safety margins and using a stepwise penalty function, the method successfully extended the refueling cycle length and enhanced neutron flux while maintaining operational limits and safety conditions. The results demonstrated significant improvements in refueling efficiency and neutron flux distribution, providing a practical and effective solution for fuel management in research reactors [95]. In 2017, Kooyman et al. analyzed and optimized minor actinides transmutation blankets in fast reactors, focusing on minimizing neutron and gamma sources while maximizing transmutation performance. The study found that using a moderated spectrum in the blankets, particularly with hydrides, is beneficial [97]. In the same year, Khotbeh-Sara conducted a study on optimizing the production of Mo-99 using a high-power electron accelerator LINAC. The study found that a hemispherical target geometry is more efficient, increasing photoneutron yield by 14.33% compared to cylindrical geometry [91]. Tsechanski (2019) conducted a study to optimize the bremsstrahlung converter for photonuclear production of Mo-99 using Monte Carlo simulations. The study introduced an effective bremsstrahlung efficiency criterion, resulting in a 2-2.8% increase in Mo-99 production [98].

Daily and McDuffee (2020) conducted a study on the optimization of ^{238}Pu production in $^{237}NpO_2$ targets irradiated at the HFIR and ATR. By varying target materials, containment designs, and irradiation histories, the study aimed to meet NASA's requirement for a steady production of $^{238}PuO_2$ [99]. Salyer (2022) proposed a meta-heuristic optimization framework using Gnowee to improve the efficiency of medical isotope production through special target design. The study utilized the FNS at the University of Tennessee to simulate neutron transport and optimize target materials and geometry, resulting in an efficient production method

for isotopes such as ^{64}Cu [100]. In 2023, Conant conducted a neutronic design optimization of an accelerator-driven subcritical assembly for Mo-99 production using MCNP simulations. The study optimized reflector type, pitch, and LEU/NU mass to maximize k_{eff} and Mo-99 production [92]. Chen conducted a shielding assessment and optimization for the target station used in medical isotope production based on a superconducting proton linac. The study used FLUKA simulations to ensure radiation protection requirements were met and achieved dose rate reductions through optimized shielding [93].

2.1.5 Other applications

Evolutionary algorithms have been repeatedly used for many other optimization applications related to ICFM, core design, nuclear power plants, and nuclear engineering.

Zhao, Edwards, and Lee (1997) developed a hybrid feedforward and feedback control system for nuclear steam generators using GA. This system enhances performance and robustness across various operational conditions [101]. Schlieck et al. (2001) focused on optimizing gadolinia as BA for in-core fuel management in Pressurized Water Reactors (PWRs). The study presented optimized gadolinia concepts that are technically feasible and economically beneficial, resulting in higher fuel assembly discharge burnup, better power distribution, and effective control of excess reactivity [102]. Kim (2010) applied evolutionary algorithms to radiation shielding design, highlighting the improvements in shielding effectiveness and material utilization [103]. Chen (2019) and Ying (2016) further applied evolutionary algorithms to radiation shielding design, achieving notable improvements in design efficiency and effectiveness [104, 105]. Pevey et al. (2020) used GA to optimize the design of a coupled fast and thermal Fast Neutron Source. By adjusting core components to maximize neutron flux while maintaining criticality, the study significantly reduced the need for expert input and design time [106]. Solans (2021) investigated the optimal arrangement of spent fuel in storage casks using evolutionary algorithms, resulting in improved storage efficiency and safety [107]. Liu et al. (2022) expanded the application of evolutionary algorithms to the design of space nuclear reactors and other components of nuclear power plants, such as steam generators [108, 109], pressurizers [110], and passive thermal-hydraulic safety systems [111]. Rafiei (2022) also contributed to this field by optimizing space nuclear reactor design [112]. Xu et al. (2022) presented an NSGA-II algorithm-based Linear Quadratic Gaussian (LQG) controller design for nuclear reactor power control. This approach demonstrated enhanced performance in terms of response speed, accuracy, stability, and robustness in PWRs [113]. Andersen and Kropaczek (2023) in-

troduced MOOGLE, a genetic algorithm-based tool for 3D design optimization of nuclear fuel assemblies. The research highlighted MOOGLE's ability to produce optimized solutions by balancing trade-offs between different fuel elements, demonstrating its flexibility and effectiveness in fuel assembly design for both PWRs and BWRs [114].

2.2 Reactor Type

Similar to the proportions of nuclear reactors operating worldwide, most studies on the optimization of ICFM problems focus on PWR. A smaller portion of the research is dedicated to BWR, research reactors, and Generation IV technology. This Section briefly presents the characteristics of the problems considered along with the current state of research, categorized by the selected reactor type.

2.2.1 PWR

Optimizing the loading pattern (LP) of a Pressurized Water Reactor (PWR) involves managing numerous constraints and variables, making it a complex challenge. The main goal is to minimize fuel cycle costs while adhering to constraints like power peaking factors, reactivity bounds, maximum enrichment, power density, burnup limits, and shutdown margins. One key challenge is ensuring a negative moderator temperature coefficient (MTC) at higher powers, which often requires burnable absorbers (BA) to flatten power distributions and reduce soluble boron concentration. PWR typically use both integral and discrete BA, such as gadolinia and erbium oxide, to manage reactivity [82].

Many studies, particularly the earliest ones, have focused on simplified problems with a generic PWR featuring three different types of FA or different sectors with varying degrees of enrichment or burnup [63, 67, 118]. In the literature, there is a number of papers addressing specific plant designs or existing, operational plants. Table 2.1 summarizes the studies mentioning the analyzed technologies. Hongchun (2001) analyzed the reloading optimization of the AC-600 model PW, aimed to minimize PPF and maximize soluble boron concentration [115]. In 2006, Gozalvez et al. conducted a sensitivity study on the TMI-1 reactor, using a hybrid GA with Neural Networks. Their study focused on optimizing the fuel assembly parameters to maximize cycle length and minimize gadolinium usage while maintaining power limits [116]. Babazadeh et al. (2009) optimized the fuel core loading pattern of a VVER-1000, a Russian-type PWR, using PSO. The algorithm successfully maximized the core k_{eff} and minimized the

Name of the reactor	Year	Type	Optimization goals	Optimization Algorithm	Ref.
Westinghouse three-loop core	1996	SR	Minimize power peaking, fresh fuel and BP inventories, maximize cycle length	Non-dominated GA	[63]
Cylindrical 3-enrichment-zone generic PWR	2003	SR	Minimize average peak-factor, maintain constraints on thermal flux, criticality, and sub-moderation	Island GA	[67]
AC-600 PWR	2001	Design	Minimize power peaking, maximize boron	GA	[115]
TMI-1 PWR	2006	Design	Maximize cycle length, minimize Gadolinium	GA-NN	[116]
VVER-1000 PWR	2009	EC	Maximize Keff, minimize power peaking	PSO	[86]
CHASNUPP Unit-1	2014	FC	Flatten power density	SA, GA, hybrid GA(SA)	[73]
100 kWe space reactor	2022	Design	Minimize cost, power peaking	GA, PSO	[112]
Almaraz II	2022	EC	Optimize cycle length, keff, power peaking	GA	[4]
APR1400	2022	FC	Minimize fuel cost	EA, SA	[117]
APR1400	2023	EC	Minimize Pin Peaking Factor	SA	[87]

Table 2.1: Summary of selected PWR optimization studies.

local PPF [86]. A hybrid optimization approach using SA and GA was proposed by Zameer et al. (2014) to optimize the core loading pattern of the Chashma Nuclear Power Plant (CHASNUPP) Unit-1, a 300 MWe PWR [73]. In 2022, Rafiei and Ansarifar utilized both GA and PSO to design and optimize the fuel composition of a 100 kWe space nuclear reactor. Optimization of reactors for space applications [103, 112] is particularly important due to their limited allowable mass and volume, where every gram and cubic centimeter matters and can significantly impact the feasibility of the project and the final costs. Their new design showed improved reactivity and safety compared to existing configurations [112]. Rodrigues et al. (2022) applied

GA to find the EC for the Almaraz II Nuclear Power Plant. This optimization identified an optimal loading pattern that met physical conditions and 240 days cycle [4]. Umarov et al. (2022) focused on optimizing the initial core parameters of the APR-1400 PWR using an evolutionary algorithm and SA. They achieved a cost-efficient, high burnup initial core model meeting safety and economic requirements [117]. In 2023, Hudayi et al. used SA to optimize the EC for the APR1400. Their study aimed at flattening the power distribution [87].

2.2.2 BWR

Boiling Water Reactors (BWR) face unique challenges in nuclear fuel management compared to PWR. As water in BWR boils, two phases — liquid and gas, appear in the core simultaneously, the axial power distribution plays a crucial role and requires full 3D modeling coupled with thermal solvers.

The fuel assemblies of BWR are about 40% smaller compared to PWR and are typically composed of 8×8 or 9×9 grids of fuel rods [7,119]. Unlike PWR assemblies, BWR assemblies contain various levels of enrichment within the same assembly. Specifically, less enriched fuel is placed at the corners of the assembly to compensate for additional water gaps [7]. BWR use around 750 FA compared to around 200 in PWR [7], increasing the search space and further complicating the problem. BWR do not have traditional control rods. Instead, they use cross-shaped control structures inserted from the bottom between the assemblies. Due to boiling water in the reactor, it is impossible to use soluble boron because it dissolves well in water vapor and would deposit on the turbine [119]. Therefore, BWR require precise control over reactivity and spatial power distributions using control elements and BA, like gadolinia. The shutdown margin (SDM) for BWR must ensure subcriticality under cold conditions, even with one control rod stuck, whereas PWR use soluble boron to manage SDM at hot zero power (HZP). BWR need to carefully manage the placement and burnup of fuel assemblies and BA, integrating the control rod program (CRP) with the loading pattern. This fuel management is more complex than in PWRs, where soluble boron simplifies reactivity control [82].

The presence of two fluid phases in the core, multi-enrichment of fuel assemblies, lack of soluble boron, and the coupling of control element movement with loading configuration make practical patterns and fuel management more complex in the case of BWR. Due to the higher complexity of the problem and the smaller number of BWR plants in the fleet, the topic of BWR ICFM is less frequently addressed in the literature. It often focuses more on the optimization of the fuel assembly design itself. Table 2.2 summarized selected applications.

Reactor name/type	Year	Type	Optimization goals	Optimization Algorithm	Reference
ABWR 1356 MWe	2002, 2003	SR	Minimize keff deviation, power peak, FLCPR, LPD	Two-stage GA	[69, 120]
Laguna Verde Nuclear Power Plant (LVNPP)	2009	SR	Optimize loading pattern for 10th cycle	GA	[72]
LVNPP with MOX fuel	2011	EC	Maximize energy production, meet thermal and reactivity margins	GA	[121]
General BWR	2021	Design	Maximize Cycle Length, minimize power peaking factor	RL	[65]
PWR and BWR fuel assembly	2023	Design	Minimize peak pin power, maximize end-of-cycle reactivity	GA	[114]

Table 2.2: Summary of selected BWR optimization studies.

Kobayashi and Aiyoshi, in their works from 2001 and 2002, presented a two-stage (LP + CRP) GA for optimizing fuel management in BWR. This approach incorporated heuristic rules to enhance performance, focusing on minimizing the deviation of the effective multiplication factor (k_{eff}), reducing power peaks, and maintaining Fraction of Limiting Critical Power Ratio (FLCPR) and Limiting Power Density (LPD) below the limits [69, 120]. In 2009, Martín-del-Campo et al. proposed a flexible GA using a binary representation for BWR fuel loading pattern optimization. This method included heuristic functions to enforce positioning rules and objective functions evaluating neutronic and thermal-hydraulic parameters [72]. François et al., in 2011, optimized an EC for a BWR loaded with MOX fuel and minor actinides. Their goal was to maximize energy production while meeting thermal and reactivity constraints. The GA they used, achieved a 5% increase in energy output, improving shutdown margins and overall energy efficiency [121]. Radaideh et al. (2021) explored the use of reinforcement learning (RL) for nuclear assembly design optimization. They compared RL to stochastic optimization

algorithms such as GA and SA. The objective was to maximize cycle length while adhering to safety and economic standards. RL algorithms showed better performance in high-dimensional problem spaces, highlighting their potential n nuclear fuel assembly design [65].

2.2.3 Research Reactors

Research reactors are mainly pool-type and low-power reactors, designed not for energy production but to provide a high-density neutron flux for materials and isotope research, radionuclide production, as well as for training and educational purposes. Due to their low power, these reactors do not usually face issues with power distribution or need to maximize burnup, and other challenges encountered by commercial reactors in nuclear power plants. Due to the diversity of technologies and applications of research reactors, and their experimental nature, it is difficult to identify typical optimization problems and their characteristics for this group. Below, selected studies available in the literature are briefly described.

Binh et al. (2014) introduced a binary mixed integer coded GA for LPO in the TRIGA MARK II research reactor. The algorithm aimed to maximize k_{eff} and minimize the PPF, successfully finding optimal LP that met operational and safety constraints [74]. Phan et al. (2019) conducted a comparative analysis of GA and Differential Evolution (DE) for optimizing the ICFM strategy of the DNRR (*Dalat Nuclear Research Reactor*). The study found that DE showed higher stability and efficiency in exploring the search space compared to GA, which were prone to trapping in local optima [122]. Shaukat et al. (2021) proposed a multi-objective LPO for the Pakistan Research Reactor-1 (PARR-1) using a modified GA coupled with Monte Carlo methods. The optimization aimed to maximize the k_{eff} and thermal neutron flux while keeping low PPF [79].

In the context of research reactor optimization, it is more common to encounter design optimization (of the reactor, assembly, or experimental setup) for specific applications, e.g. material studies [96, 123] or isotopes production [90, 96, 100], by maximizing the neutron flux [79] or increasing the proportion of fast neutrons in the neutron spectrum [123].

Reactor name/type	Year	Type	Optimization goals	Optimization Algorithm	Ref.
NUR	2006	Design	Maximize neutron flux and ensure safety margins	No algorithm	[94]
High Flux Isotope Reactor (HFIR)	2012	Reload	Optimize production of 252Cf	GA, NN	[96]
TRIGA MARK II	2014	Reload	Maximize keff and minimize PPF	Binary mixed integer coded GA	[74]
AHR	2014	Design	Minimize critical U-235 mass and maximize reactor power	-	[90]
DNRR (Dalat nuclear research reactor)	2019	Reload	Maximize keff and minimize radial power peaking factor (PPF)	DE, GA	[122]
Tehran Research Reactor (TRR)	2020	Design	Enhance fast neutron irradiation	-	[123]
High Flux Isotope Reactor (HFIR) and Advanced Test Reactor (ATR)	2020	Design	Optimize production of 238PuO2	MCNP, ORIGEN, ADVANTG	[99]
Pakistan Research Reactor-1 (PARR-1)	2021	Reload	Maximize keff and thermal neutron flux	MGA, MC	[79]
Fast Neutron Source (FNS)	2022	Design	Optimize isotope production target design	Gnowee	[100]

Table 2.3: Summary of optimization studies on various research reactors.

2.2.4 Gen-IV Reactors

Fourth-generation reactors (Gen-IV), such as the sodium fast reactor (SFR), breeding reactors, and thorium reactors, have been analyzed several times as subjects for optimization. Due to their more design-oriented stage rather than operational, optimization has usually been considered at the level of the technology's design. Due to the diversity of Gen-IV reactor tech-

nologies and the limited number of examples in the literature, it is difficult to identify general trends or problems that have been the focus of optimization within this group. In 2015, Kumar and Tsvetkov proposed a modular approach that combines GA with regression analysis to optimize the design of a gas-cooled fast breeder reactor. The study focused on achieving high breeding of U-233 and Pu-239, desired k_{eff}, and high thermal efficiency while maintaining a constant total fuel mass [124]. Kooyman et al. (2017) analyzed and optimized minor actinides transmutation blankets in fast reactors. Their focus was on reducing neutron and gamma sources while maximizing transmutation performance. The study found that using a moderated spectrum in the blankets, particularly with hydrides, is beneficial [97]. In 2021, Li et al. conducted a study comparing ten different GA selection strategies for LPO in thorium-based high-temperature gas-cooled reactors (HTGRs). The study considered different sorting strategies for single-objective and multi-objective optimization [77]. The same year, Żurkowski et al. applied GA to optimize the fuel isotopic inventory and boron shield distribution in a sodium-cooled fast reactor (SFR). The study demonstrated improvements in managing sodium void reactivity and minimizing long-lived actinides, although maintaining criticality posed some challenges [125]. In 2023, Lima-Reinaldo and François employed GA to optimize the fuel loading pattern of the ALLEGRO experimental fast reactor. Their research integrated GA code with the ERANOS 2.3 deterministic neutronic simulation code, utilizing a penalty function method to handle multi-objective combinatorial optimization. Through advanced crossover operators, such as the Partially Mapped Crossover and the Order Crossover (crossover based on ordered population), alongside random mutations, the study achieved improvements in operational cycle length while adhering to constraints on PPF, excess reactivity, and linear heat generation rates [81].

3 Optimization of the Single Cycle Problem

3.1 Westinghouse PWR Core

3.1.1 BEAVRS Benchmark

The considered core is based on the BEAVRS benchmark (*Benchmark for Evaluation And Validation of Reactor Simulations*) published by the MIT Computational Reactor Physics Group [1], [126]. It defines a reactor model based on a real 4-loop Westinghouse PWR nuclear power plant with a thermal power of 3411 MWth (Table 3.2). The first fuel cycle of the BEAVRS model is used as a reference case in this study, and its original loading scheme is shown in Figure 3.1. The core consists of 193 fuel assemblies arranged in a 17×17 grid. There are nine types of fuel assemblies with three enrichments (1.6, 2.4, 3.1 wt%) and varying populations of burnable absorbers (BA) in the form of borosilicate rods: 0, 6, 12, 15, 16, or 20 per assembly (Figure 3.1). The types of assemblies used in this work are limited to those used in the first cycle of the BEAVRS model (Table 3.1).

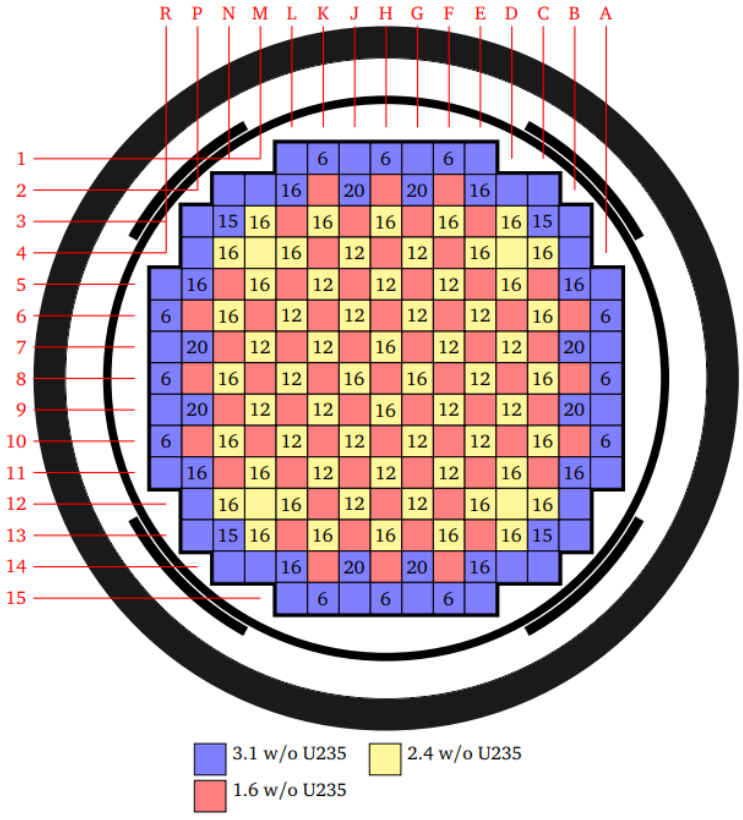


Figure 3.1: Scheme of the first loading of the core from the BEAVRS benchmark [1]. Colors correspond to different enrichment levels, and numbers indicate the number of rods with burnable absorbers.

Designation	Enrichment [%]	Number of BA rods	Total number of assemblies
FA1	1.6	0	65
FA2	2.4	0	4
FA3	2.4	12	28
FA4	2.4	16	32
FA5	3.1	0	32
FA6	3.1	6	12
FA7	3.1	15	4
FA8	3.1	16	8
FA9	3.1	20	8

Table 3.1: Fuel assemblies (FA) in the BEAVRS model [1], numbered from FA1 to FA9, their enrichment, number of BA rods, and the total number of FA of a given type used in the model.

Figure 3.4 presents a general scheme of a fuel assembly used in the BEAVRS core. Each fuel assembly consists of a 17×17 grid of pins (rods in each fuel assembly) with a diameter of about 1.16 cm each, surrounded by a zirconium cladding with a thickness of 0.4 mm. Depending on the assembly design, the rods may contain fuel with enrichments from 1.6 to 3.1 %, burnable absorber material, or an empty channel for control rods or measurement equipment (Table 3.2). For different assemblies, available channels may contain from 0 to 20 rods with burnable absorbers (positions *G* in Figure 3.2), or remain empty (filled with water). The central rod never contains an absorber and is intended for measurement equipment (position I in Figure 3.2) [1]. The characteristics of the used assemblies in the model are presented in Table 3.1. Selected core construction details are shown in Table 3.2.

3.1.2 Numerical Model

To solve the static criticality eigenvalue problem (see Section 1.2.2) along with isotopic burnup, a two-stage SCALE-PARCS methodology was employed. Simulations were performed using SCALE 6.1.2 and PARCS 3.2 codes [128], [129]. The TRITON/NEWT code [128] (a part of the SCALE package) was used to solve neutron transport and nuclide evolution equations for each element defined on a two-dimensional grid. The GenPMAXs generator was used to generate multigroup PMAXS cross-section libraries. The SCALE code used the ENDF/B-VII library with 238 neutron groups, and the results were used to generate data in two energy groups in the PMAXS libraries. Then, the PARCS code, along with the prepared libraries, was used as a

Core:	
Number of fuel assemblies:	193
Assembly configuration:	15×15
Core radius:	193.7 cm
Inner pressure vessel radius:	219.7 cm
Outer pressure vessel radius:	241.3 cm
Mass of fuel in the first loading:	81.1 Mt
Active length:	365.8 cm
Design pressure:	15.51 MPa
Thermal power:	3411 MWth
Flow rate:	$6.15 \cdot 10^6$ kg/h
Fuel Assemblies:	
Number of fuel rods:	264
Rod configuration:	17×17
Assembly grid pitch:	21.5 cm
Rod grid pitch:	1.26 cm
Enrichment:	1.6; 2.4; 3.1 % wt
Number of BA rods:	0, 6, 12, 15, 16, 20
Control Rods:	
Number of assemblies:	57
Control rod material:	B_4C, *Ag-In-Cd*
Total number of BA rods:	1266
BA rod material:	B_2O_3

Table 3.2: Operational and design data of the PWR core described in the BEAVRS benchmark [1].

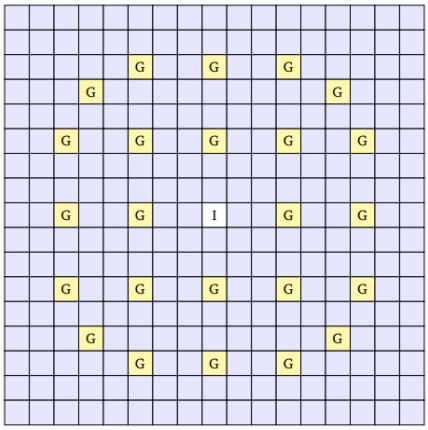


Figure 3.2: Scheme of a single fuel assembly from the BEAVRS benchmark [1]. Notations: *G* - guide tube, space for additional elements (rods with absorbers, control rods), *I* - instrument tube space for possible measurement equipment. Blue elements represent fuel rods.

three-dimensional core simulator solving neutron transport equations with a diffusion approximation (Equation 1.9) on a $17 \times 17 \times 22$ grid, using the PATHS code for thermal-hydraulic process modeling [129], [130]. Details of the two-stage methodology and the PARCS reactor model used in this work, the modeling approach, and its verification for the first BEAVRS cycle, are described in [129]. Detailed core construction data is available in [1, 126].

The PARCS core model grid has the same arrangement as the core configuration (Figure 3.3), where one radial grid node (nodes on the core plane) corresponds to one assembly. The axial division (split along the core height) consists of 20 equally spaced levels of the active core part (fuel part) and additional two parts of the reflector, upper and lower.

Fuel cycle simulations using PARCS were conducted for the full core power state equal to 3411 MWth. For simplicity, power maneuvering was not considered in the model. Additionally, all control rods were fully withdrawn from the core - the *All Rods Out* (ARO) state. This simplified the problem and eliminated any differences arising from the presence of control rods. All cases were computed for a fixed final simulation time of 600 days, which was sufficiently large to ensure that no configuration would have a longer cycle length. The core was simulated in a xenon/samarium equilibrium state, meaning transient states for Xe/Sm were not considered. This is the reason why the xenon peak is not present in the evolution of k_{eff} for the entire core in subsequent chapters. This approach primarily reduced computation time. Additionally, the transient state associated with xenon concentration changes during full power operation only affects below 0.5 GWd/tHM. Therefore, no differences were observed in calculations between simulations with transient states for Xe and Sm and those without them for later parts of the cycle, and these parts plays a crutial role from the optimization standpoint.

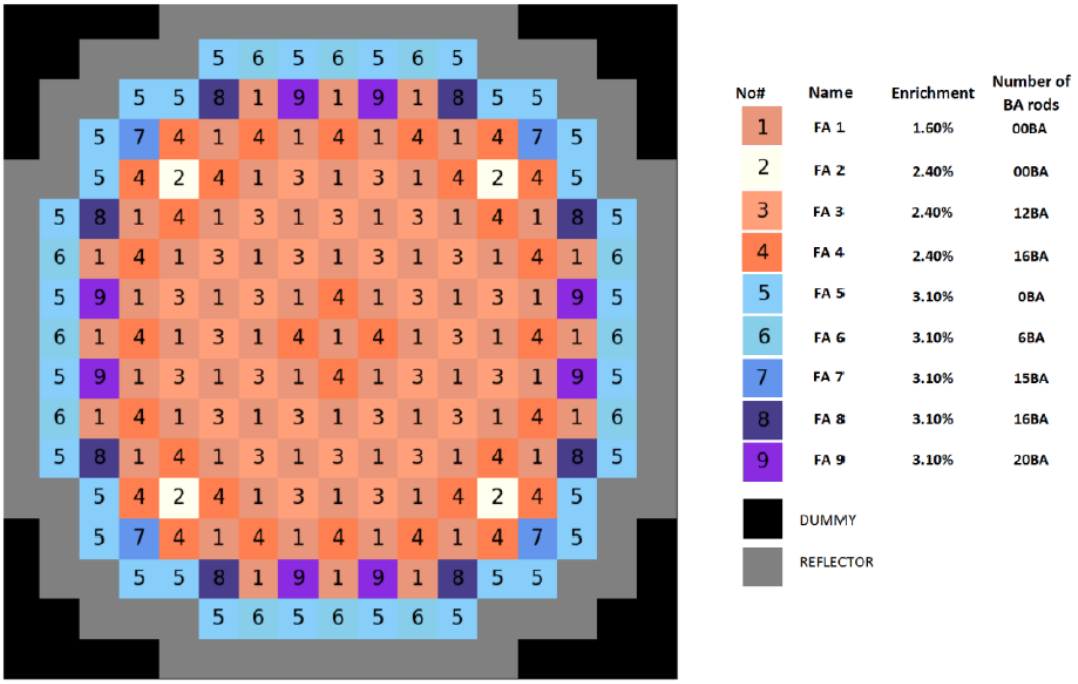


Figure 3.3: Layout of the BEAVRS core model [127].

The impact of boric acid and its concentration was not investigated in this work, and the concentration was set to *0 ppm* in all simulations. In this case, all simulations were performed in criticality mode, i.e., by determining the critical value of k_{eff}. Simulations without boron were conducted to eliminate differences caused by its presence and to avoid excessive extrapolations by PARCS in PMAXS libraries, which can easily disrupt code operation. During simulations of thousands of random core configurations with the critical concentration search mode, it is easy to reach states with non-physical boron concentrations below zero or above solubility limits, generating non-physical solutions and/or disrupting the convergence of calculations. To indicate possible differences, the reference first BEAVRS fuel cycle was calculated using the criticality search algorithm [129].

In Figure 3.5, reactivity obtained by determining k_{eff} is compared with the boron acid concentration curve obtained using the critical concentration search algorithm [127, 129]. Both curves are normalized to the initial reactivity value determined by the boron acid method. The plot shows differences resulting from the use of the k_{eff} search mode. There is a relatively small deviation, especially in the initial 2/3 of the cycle, later the difference disappears. For boron acid calculations, the end of the cycle is 334.1 days, and for the k_{eff} case, it is 333.7 days. The difference is therefore 0.4 days (around 0.1 %). The length of the fuel cycle is practically the same for both calculations, and the presence of boron will have a negligible impact on the

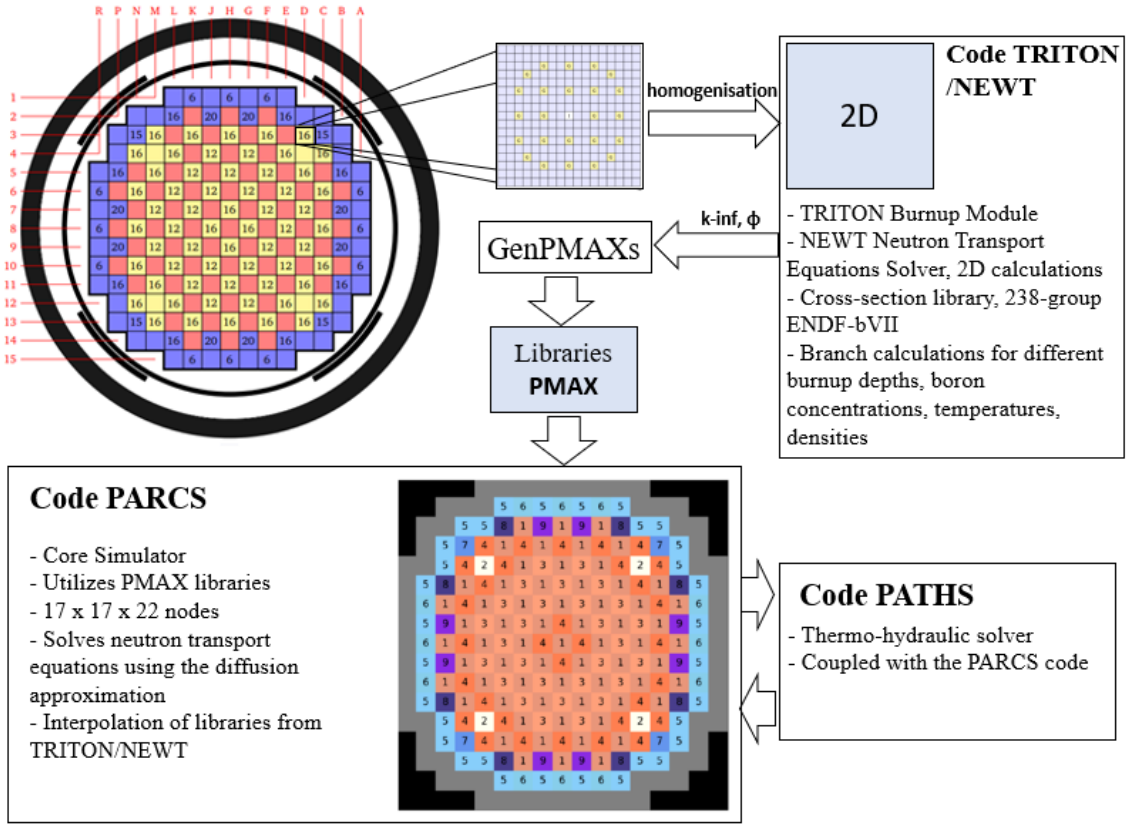


Figure 3.4: Preparation of the PARCS model [127].

algorithm's operation. In criticality mode, the end of the cycle occurs when $k_{eff} \rightarrow 1.0$, and for the boron search mode when the boron acid concentration $\rightarrow 0.0$ ppm.

3.2 Genetic Algorithm Implementation

This section describes the development and application of a genetic algorithm (GA) and numerical tool dedicated to the fuel management process in a nuclear reactor core. The developed solutions are tailored for nuclear power plants with PWR-type reactors. The algorithm has been tested on the first core cycle described in the BEAVRS MIT benchmark [1]. The analysed problem involved finding the initial core loading pattern that maximizes the fuel cycle length under additional predefined operational constraints: reactivity/effective multiplication neutron factor margin, power distribution flattening, the number of rods with burnable absorbers, and average enrichment.

3.2.1 Genetic Operators

As mentioned in Section 1.3.1, Genetic Algorithms operate on data matrices, and a given core configuration can be directly transformed into a two-dimensional array (chromosome) con-

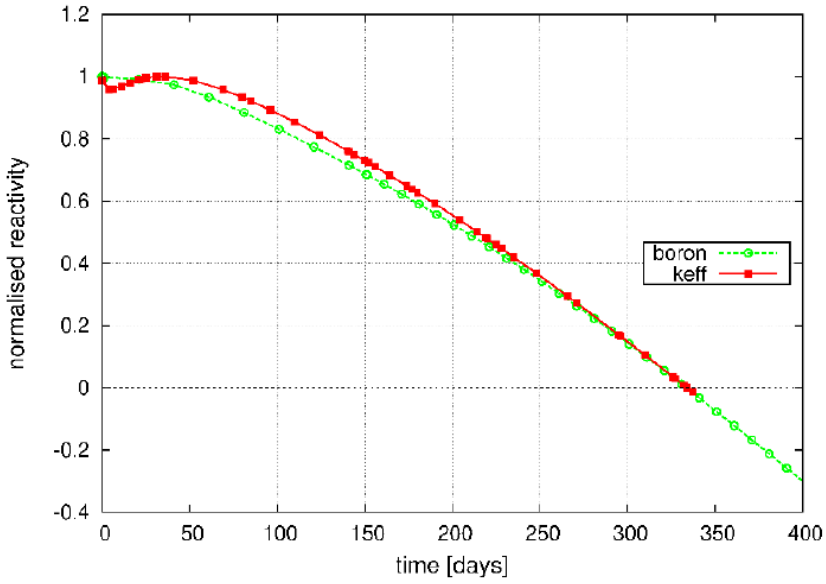


Figure 3.5: Comparison of calculations in criticality mode and with the search for boron acid concentration.

taining numbers corresponding to a specific type of fuel assembly. In the following section, the key components of GA will be briefly described: the chromosome structure, the fitness function, and the three main genetic operators, namely selection, crossover, and mutation.

Chromosome

In the considered case, the chromosome is represented as a two-dimensional matrix reflecting the current core arrangement. Figure 3.6 shows an example of a chromosome used in simulations. The algorithm was implemented for a 1/4 part of the core, which was then symmetrically mirrored to construct a full core with 1/4 symmetry. Since the complexity of finding an optimal core configuration decreases exponentially with the number of fuel assemblies [131], assuming symmetry significantly reduces the searched space of possible solutions, and the time needed for the algorithm to reach the optimum is shortened. From a design perspective, core loading patterns typically have 1/4 or even 1/8 symmetry. Consequently, the applied symmetry assumption seems reasonable, significantly simplifying the problem and improving computational efficiency. It's worth mentioning that the original BEAVRS core also had an assumed symmetry. However, in the literature, it is suggested that symmetry is not always necessary [37] and may lead to the rejection of potentially good solutions. To demonstrate the impact of the symmetry assumption on the problem's complexity and solvability, this study also considered the case of complete asymmetry.

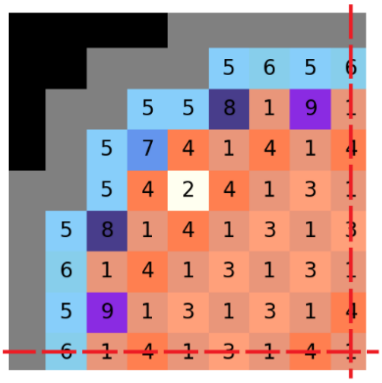


Figure 3.6: Chromosome representing a 1/4 core configuration of the BEAVRS model, numbers 1-9 corresponds to the FA described in Table 3.1, grey region is the reflector and black region is void.

In Figure 3.6, numbers 1 through 9 refer to fuel assemblies with three different enrichments (blue, orange, and brown), while the shade of each color symbolizes the number of burnable absorber rods inside the assembly. The grey area represents the reflector, and the black area is the void. This set of numbers (genes) arranged in a matrix (chromosome) of size 8×8 is called a genotype, and the set of features of such a chromosome (cycle length, k_{eff}, etc.) is called a phenotype.

Fitness Function and Selection Method

The Fitness Function (FF) is a crucial parameter in GA operations because the probability of selecting a chromosome for further processing depends on its fitness function value (in general, the higher the FF, the greater the probability). In this part of the work, six simulations were conducted with different assumptions and forms of the fitness function, but the main goal was to maximize the cycle length while optimizing and/or preserving other operational and safety parameters at the same time. Selection can be implemented in various ways; however, it usually relies on determining the FF of a given chromosome and comparing it with others. The basic method is the roulette method, where we sample N new chromosomes with a probability proportional to their FF values. This simple method is sufficient in many cases, e.g., [132], [133], [134]. Specifically, when FF values are too similar to distinguish in the roulette wheel method, other variants of selecting a new generation can be applied, such as tournament [135], [136], [137], where we compare two or more chromosomes, or ranking [138], [139], where the possibility of choosing a chromosome depends on its position in the list sorted by FF. In this part of the study, the roulette wheel method and its modified version were used (described in Section 3.2.4).

Crossover

Crossover involves exchanging data subarrays between chromosomes (parents) to create new ones (offspring). As a result, these chromosomes inherit a combination of genes from their parents, and this combination may potentially lead to a better solution. The easiest way to implement crossover is to randomly select one e.g., [132], [133], [136] or two [135], [140], [138] intersection points and exchange subarrays between parents separated by this point or points. In this study, a modification was applied in the form of a two-dimensional chromosome, and crossover was carried out in two stages. In the first step, the intersection point (x, y) is randomly selected from a uniform discrete distribution $x \subset \langle 2; D \rangle$ and $y \subset \langle 2; D \rangle$ (where D is the chromosome size). This way, the chromosome is divided into four parts, and one of the parts (randomly selected again using a uniform distribution) is exchanged between parents (see Figure 3.7).

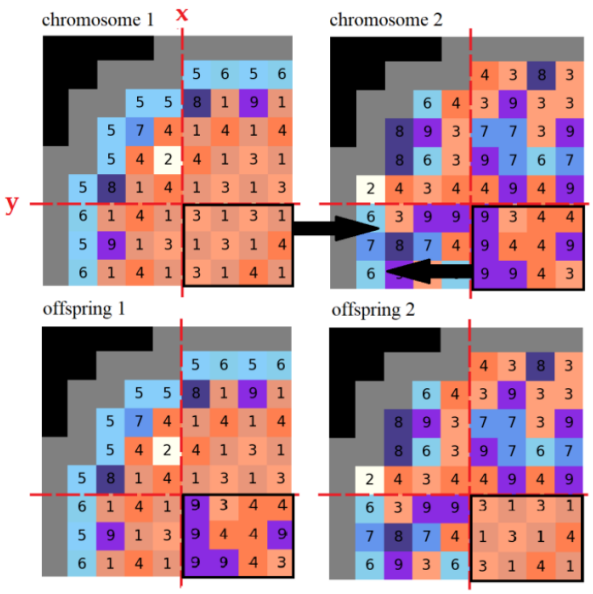


Figure 3.7: Crossover scheme.

Mutation

Mutation is a simple but extremely important GA operator. It plays a crucial role in later generations, where the population's diversity is reduced [141]. Mutation involves randomly changing a selected element in the chromosome. This simple operation allows further exploration of solution space, which is inaccessible by using the crossover operation only. Mutation can be implemented in various ways. In the literature, solutions can be found where the mutation level is constant [136], [137] or changes in time [133], [138]. In this study, the simplest option was chosen, so the mutation probability was constant over the optimization steps and the same for

each population member. Depending on the case under consideration, the mutation probability, denoted as p_m, meant: for Cases 1-5, the probability of replacing a single gene with another (randomly chosen), and for Case 6, the probability of replacing two randomly chosen genes in a given chromosome (see next Section 3.2.3).

3.2.2 Genetic Variance

In order to monitor the course of the simulation, not only on the basis of the achieved fitness function, an indicator was defined that described the diversity of individuals in the current population. This indicator was based on the definition taken from [142] and consistently called "genetic variance" (diversity), described by the formula:

$$S = \frac{1}{n_{\text{tot}}} \sum_{k=1}^{N} \sum_{l=k+1}^{N} \sum_{i,j} (1 - \delta_{s^k_{i,j}, s^l_{i,j}}), \tag{3.1}$$

where δ is the Kronecker delta, $s_{i,j}$ represents the gene (assembly) i, j in the cores k and l compared with each other, N is the size of the population, and $n_{\text{tot}} = \frac{1}{2}N(N-1) \cdot d^2$ is the number of all comparisons made in total, d is the size of a single chromosome. Then, the variance S is a normalized sum of differences between all individuals in a given population. S equals to 1 for a population in which each individual in the population is different from the others as to a single gene (no gene repeats at any position) and 0 for the population completely dominated by one individual.

3.2.3 Considered Scenarios

In this section, six versions of the algorithm are described. The versions are based on the same model and genetic operators, but different form of FF, aiming at optimizing from different perspectives. Case 1 is a test scenario assuming the simplest form of the FF, focusing solely on cycle length maximization. Cases 2-4 deal with multiparameter optimization, aiming to extend the cycle while simultaneously optimizing and/or maintaining selected parameters within a specified range. Case 5 tested the possibility of achieving the most even power distribution, defined in this study as the second most important optimized feature (this scenario was the only one not related to cycle length). Case 6 was a combination of Case 1 and 5, aiming to extend the cycle and achieve an even power distribution.

General fitness function

For Cases 1-4 and 6, the general FF had the form:

$$FF = \left(\frac{d}{\alpha}\right)^{\beta}. \tag{3.2}$$

The parameter β represents sensitivity, while α is a penalty parameter whose form depends on the selected scenario. The sensitivity parameter is either constant or adjusted by the algorithm in scenarios using variance control. The value d is the cycle length determined by the PARCS core simulator. In cases where $k_{eff} = 1.0$ is between computational points n and $n+1$, the parameter d is linearly interpolated:

$$d = d_n + (1 - k_n) \cdot \frac{d_{n+1} - d_n}{k_{n+1} - k_n}. \tag{3.3}$$

This solution is justified for the studied cases due to the linear evolution of the eigenvalue k_{eff} over time (Figure 3.5). It has been additionally tested for a 20-day time step compared to calculations for a one-day time step at the end of the cycle (EOC). The difference between the 20-day time step (the largest applied step at the end of the cycle) and the same case calculated based on a one-day time step was 0.05 days (0.01%). It was shown that linear interpolation is reasonable, and differences of this magnitude can be considered negligible in the context of calculation uncertainties. In reality, other and potentially larger sources of uncertainty are more significant here, especially considering nuclear data, diffusion approximation applications, multigroup approximations, and simplified solutions (see Section 1.2.2).

Despite the acknowledged inaccuracy of calculations, the applied precision for the cycle length calculation (two decimal places), which is far beyond the accuracy of the calculations, is essential for the distinctiveness of individuals. It is important for the correct functioning of the algorithm that the applied methods do not introduce additional variability, which could occur, for example, when rounding the cycle length to single days.

Case 1: Cycle Length Maximization and Algorithm Testing

The first set of simulations and the algorithm design were intended to maximize the fuel cycle length with a minimal set of constraints. The only constraint was that the algorithm could use only nine types of fuel assemblies used in the first BEAVRS fuel cycle. In this case, the Fitness Function takes the form of Equation 3.2 without modifications. In this case, $\alpha = \alpha_0 = 10^2$, (to

scale d value which is in hundreds of days) and β can either have a constant value or be changed by the algorithm through the variance control method, depending on the situation. Then:

$$FF_1 = \left(\frac{d}{\alpha_0}\right)^{\beta}. \tag{3.4}$$

The first intuitively deduced solution for the optimal loading might be the maximization of the initial fissile material mass and the removal of all absorbers, thus using fuel assemblies enriched only by 3.1% and without burnable absorbers (see FA5 in Table 3.1). Nevertheless, due to the presence of burnable absorbers with different burnup rates, the evolution of the core eigenvalue may proceed in a non-intuitive or complicated way. As a result, the optimal solution may not be trivial even for such a simple FF form. Other, more optimal solutions cannot be excluded. To explore this issue, additional simulations were conducted for the entire core with only one type of fuel assembly (see Section 3.2.5). As the simplest of the considered problems, Case 1 was used as a reference point for the algorithm. Simulations were conducted to test the impact of chromosome population size on generation, the total number of generations, the impact of mutation probability, statistically verify the repeatability of algorithm operation, and to test the proposed variance control method (Section 3.2.4).

Case 2: Cycle Length Maximization and k_{eff} Constraint

The second type of simulation was conducted with a constraint on k_{eff} to a specified maximum value. In this study, the total weight of control rods was taken as the limiting factor. For the reference BEAVRS core, the rod weight was calculated for different states, and it was estimated that the lowest value was about 7700 pcm. To maintain some margin, it was assumed in the simulations that the limit for the weight of control rods is 7500 pcm. Additionally, it was assumed that for operation at full power (Hot Full Power - HFP), subcriticality is guaranteed by $k_{eff} = 0.99$ for the state with all rods in the core (All Rods In - ARI). Consequently, the eigenvalue is limited by $k_{eff}^{max} = 1.07$ and corresponds to the state where all rods are withdrawn from the core (All Rods Out - ARO). In this case, the FF was modified by a penalty factor related to exceeding the allowable value of k_{eff}:

$$\alpha := \alpha_k = \alpha_0 \cdot \left(1 + \frac{k_1 - \max(k)}{k_2 - k_1}\right)^{\gamma}, \tag{3.5}$$

where $\alpha_0 = 10^2$ is the scaling factor, $k_1 = 1.07$ is the upper limit of k_{eff}, $k_2 = 1.22$ is the

maximum k_{eff} that the analyzed model can achieve, determined by filling the core with the most enriched fuel assemblies (FA05). The value $max(k)$ is the maximum k_{eff} in the full cycle of a given chromosome (core model), and $\gamma = 2$ is the penalty influence coefficient, selected to moderate the influence of the penalty. Then:

$$FF_2 = d^{\beta} \cdot \left[\alpha_0 \cdot \left(1 + \frac{k_1 - \max(k)}{k_2 - k_1} \right)^{\gamma} \right]^{-\beta}. \tag{3.6}$$

Case 3: Cycle Length Maximization with Maintaining Average Enrichment

The third set of calculations involved an algorithm aiming to maintain a constant fissile material mass (average enrichment) relative to the fissile material mass in the BEAVRS core. These simulations aimed to find a configuration that maximizes the use of nuclear fuel and improves cycle economics. For simplicity, for the analyzed problems, the mass value is relative, and the primary parameter used in the fitness function is the average enrichment in all assemblies. It was assumed that the average enrichment is equal to the weighted average of the number of assemblies of a given type:

$$\hat{E} = \frac{\sum_i N_i \cdot E_i}{\sum_i N_i}, \tag{3.7}$$

where N_i is the number of assemblies of type i and E_i is its enrichment level. This simplification leads to the conclusion that the mass of heavy metal (uranium $^{235}U + ^{238}U$) in each assembly is the same. In reality, there are small differences in uranium mass between assemblies, and there is a slight mass difference between isotopes. For simplicity, these differences have been omitted, as all fuel assembly construction parameters are the same, fuel assemblies have the same number of fuel rods, and burnable absorber rods are placed only in the guide tube locations, not disturbing the total mass of the fuel. In this case, the parameter α was modified to account for the penalty coefficient for excessive use of fissile material:

$$\alpha := \alpha_E = \alpha_0 \cdot \left(1 + \frac{\hat{E} - E_0}{E_0} \right)^{\gamma}. \tag{3.8}$$

Again, $\alpha_0 = 10^2$ is the scaling factor, $\gamma = 2$ is the penalty influence coefficient, $\hat{E}$ is the average enrichment of a given chromosome, and $E_0 = 2.36\%$ is the average enrichment of the BEAVRS model. Then:

$$FF_3 = d^{\beta} \cdot \left[\alpha_0 \cdot \left(1 + \frac{\hat{E} - E_0}{E_0} \right)^{\gamma} \right]^{-\beta}. \tag{3.9}$$

Case 4: Cycle Length Maximization with Fixed Population and Asymmetry

This case involves solving the core loading scheme problem without changing the number of assemblies of a given type and imposing no core symmetry. As a result, only the locations of fuel assemblies are optimized not their type selection. The optimization goal is the cycle length extension. Other important core parameters, such as power distribution unevenness and reactivity margin, were not analyzed; thus, the solution should be treated with the appropriate perspective for the actual core design presented in BEAVRS. This case aimed to analyze the algorithm's capability to search for solutions without a predefined symmetry and using fuel assemblies available in the BEAVRS model (while maintaining the number of fuel assemblies of a given type). In this case, the algorithm solved the fuel assembly permutation problem in the core, with the goal of maximizing the cycle length. The fitness function then took the form of Equation 3.2, for which $\alpha := \alpha_0 = 10^2$. However, since it can be expected that after applying the crossover operator, the number of fuel assemblies of a given type will no longer be constant, an additional algorithm was implemented. The *Partially Mapped Crossover* [143] method was applied, and it is based on the idea of exchanging fuel assemblies outside the crossover region to maintain a constant number of fuel assemblies of a given type in the entire chromosome. Since mutation potentially could also disturb the number of fuel assemblies of a given type in the core, it was implemented as a random exchange of positions of fuel assemblies, not a random exchange of a fuel assembly for another (as in the case of other scenarios).

Case 5: Power Distribution Evenness Minimization

In this part of the study, it was decided to only minimize the unevenness of the power distribution, defined in this work as the second-most important parameter necessary for the optimal functioning of the core. The unevenness of the radial power distribution of a given configuration was determined by the maximum value of P_{xy} - Power Peaking Factor (PPF) in the form of Equation 1.24, determining the ratio of power in a given node (fuel assembly) to the average power in the core. In this case, lenght of the cycle was not considered and the fitness function took a simplified form:

$$FF_5 = \left(\frac{1}{PPF}\right)^{\beta} \tag{3.10}$$

where $PPF = P_{xy}$ is the maximum radial power peaking factor of a given configuration, determined by PARCS calculations [5] (see definition in Section 1.2.4).

Case 6: Power Distribution Optimization and Cycle Length Maximization

Case 6 is a combination of Cases 1 and 5. Its goal was to extend the cycle while simultaneously reducing the unevenness of the power distribution. The fitness function then had the form:

$$FF_6 = \left(\frac{d}{PPF}\right)^{\beta}, \tag{3.11}$$

where d is the cycle length, and PPF is again the maximum power peaking factor.

3.2.4 Population Variance Control Method

Over the simulation, the algorithm gets rid of weaker chromosomes and promotes the better ones. Therefore, it is obvious that the differences between chromosomes will blur with time, so the often-used roulette method may prove to be insufficient. To prevent this, a new method was proposed in this study, which was based on ongoing control of the form of the FF in order to ensure appropriate differentiation between the chromosomes. In this method, FF is raised to the β_{VC} power:

$$FF_{VC} = FF^{\beta_{VC}}, \tag{3.12}$$

and FF_{VC} is then used for the standard roulette method. β_{VC} is controlled depending on the level of the relative standard deviation (RSD) of chromosomes' FFs in the current population (population is indicated by t index). If RSD in the given step (σ_t) is below a certain threshold ($\sigma_{\min}$), the algorithm increases β_{VC} parameter, thereby highlighting the difference in FFs. This method enables the algorithm to "notice" even small changes, important in terms of finding the optimal solution. So if $\sigma_t < \sigma_{\min}$, the increase in the exponent was determined by the following principle:

$$\beta_{VC}^{t+1} = \beta_{VC}^{t} \cdot \left(1 + \frac{\sigma_{\min} - \sigma_t}{\sigma_{\min}}\right), \tag{3.13}$$

where β_t, β_{t+1} are the power of FF in step t and $t+1$, respectively and RSD is calculated as follows:

$$\sigma = \sigma_{RSD} = \sqrt{\sum_{i=1}^{N} \frac{(FF_i - \overline{FF})^2}{(N-1)\cdot \overline{FF}^2}}. \tag{3.14}$$

3.2.5 Optimization Parameter Selection

This section describes tests conducted to select the algorithm meta-parameters, i.e., population size, minimum number of generations, mutation, and crossover probabilities.

Population Size (N)

In analyzing the impact of population size on the algorithm's performance, it was assumed that a constant total number of simulations (10^4) would be carried out for different population sizes. For example, for a population of 10, 1000 steps (generations) were calculated, for 20 chromosomes 500 generations were conducted, and so on. This way, the total number of simulations was constant and the result depended only on the population size and not on the number of simulations conducted. Tests were performed for 10, 20, 50, 100, 200, and 300 chromosomes. Figure 3.8 presents the obtained results.

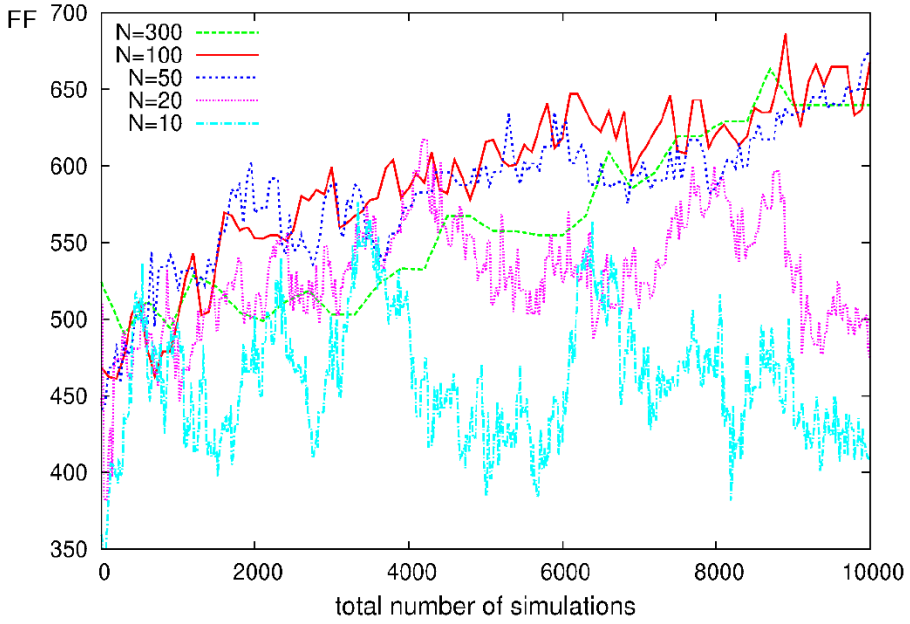


Figure 3.8: Impact of population size on algorithm performance.

For populations below 50 chromosomes, the FF value oscillates and the algorithm is unable to find a better solution, due to the limited number of different genes needed to build new combinations. Only the use of 50 chromosomes and more allows for the proper operation of the algorithm (sufficient gene diversity). On the other hand, further increasing the population size does not improve the result, but only extends the time of computing each step. For populations of at least 50 chromosomes, the more total simulations are conducted, the better the solution obtained, regardless of their distribution between the number of generations and population size.

Minimum Relative Standard Deviation (σ_{min}) in the Variance Control Method

This series of tests aimed to find the optimal value of σ_{min} in simulations that use the β exponent control (Equation 3.2 and the variance control method in Section 3.2.4). Too low σ_{min}

will not affect the operation of the algorithm (then $\beta_t \approx \beta_0$ for each step). On the other hand, too high values of this parameter may cause significant changes in the fitness function values (the β factor is increased proportionally to the difference between the current relative standard deviation of the population and σ_{min}). Too much influence of this method potentially leads to excessive divergence of FF values, rapid dominance of the best chromosomes, and drastic reduction of population diversity (variance decline). Moreover, a significant increase in the β exponent can lead to huge FF values that simply reach the limit of $float64$ ($\sim 1.80 \cdot 10^{308}$ in *Python* 3.8 [144]) and prematurely terminate the program. Therefore, it was important to find an optimal σ_{min} value that would allow for a gradual increase in the β parameter and enable the algorithm to differentiate chromosomes in subsequent generations without limiting its operation. For this purpose, simulations were conducted with σ_{min}, ranging from 0.0 to 2.0, and the results are presented in Figure 3.9. The FF variance control method with the standard roulette

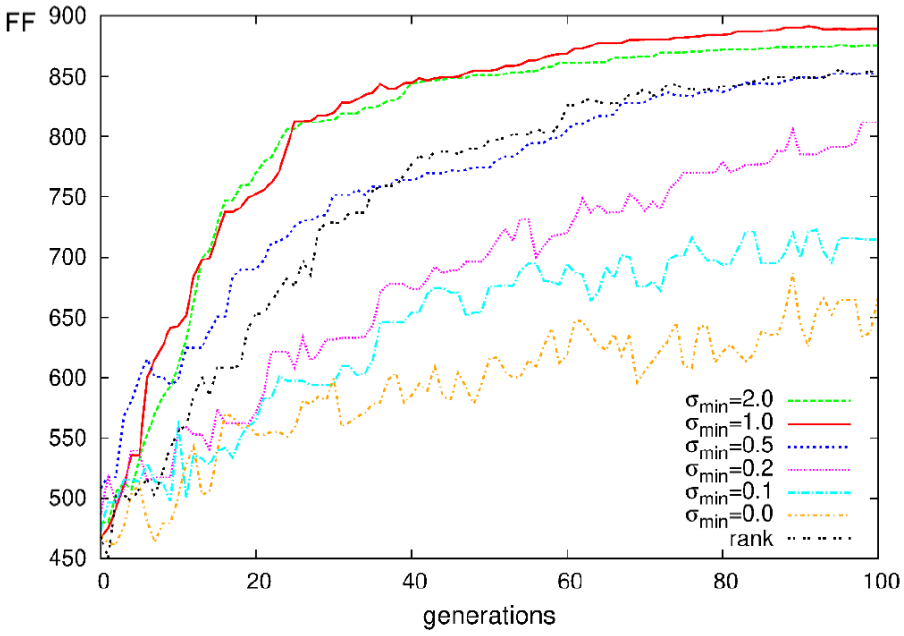


Figure 3.9: Impact of σ_{min} on algorithm performance.

method and the often used ranking method, which form was taken from [37] (Figure 3.9, dashed line "rank"). In this method, the probability of selecting a particular chromosome for crossover depends on its r ranking based on FF. In this method, the expected number of chromosomes c is equal to [37]:

$$E(c) = 2 - m + \frac{2(m-1)\cdot(r-1)}{N-1}, \tag{3.15}$$

where N is the population size and m is a certain maximum expected number of chromosomes, which is a parameter of this method. In the study on which this method is based, $m = 1.8$ [37].

It can be seen that the ranking method yields better results than the standard roulette method, but worse than the proposed variance control method with $\sigma_{min} \subset \langle 1;2 \rangle$. However, the ranking presents a very similar effect compared to using $\sigma_{min} = 0.50$. It is worth noting that the linear distribution used in the ranking method (Equation 3.15) gives $\sigma_{min} \approx 0.58$. In this case, a similar variance gives a similar behavior of the algorithm despite different probability distributions. Thus, it can be stated that the variance of the distribution related to the selection method plays a key role and determines the operation of the GA. Analyzing the graph in Figure 3.9, it is clear that the proposed algorithm works best (converges the fastest and to the highest value for $\sigma_{min} = 1.0$

Mutation Probability (p_m)

After determining the population size and variance control parameter (i.e., $N = 100$, $\sigma_{min} = 1.0$), the probability of mutation ensuring the best performance of the algorithm (with the assumed values of other parameters) was examined. For this purpose, 100 generations of simulations were conducted for mutation levels ranging from 0.1% to 5.0% (Figure 3.10 and 3.11).

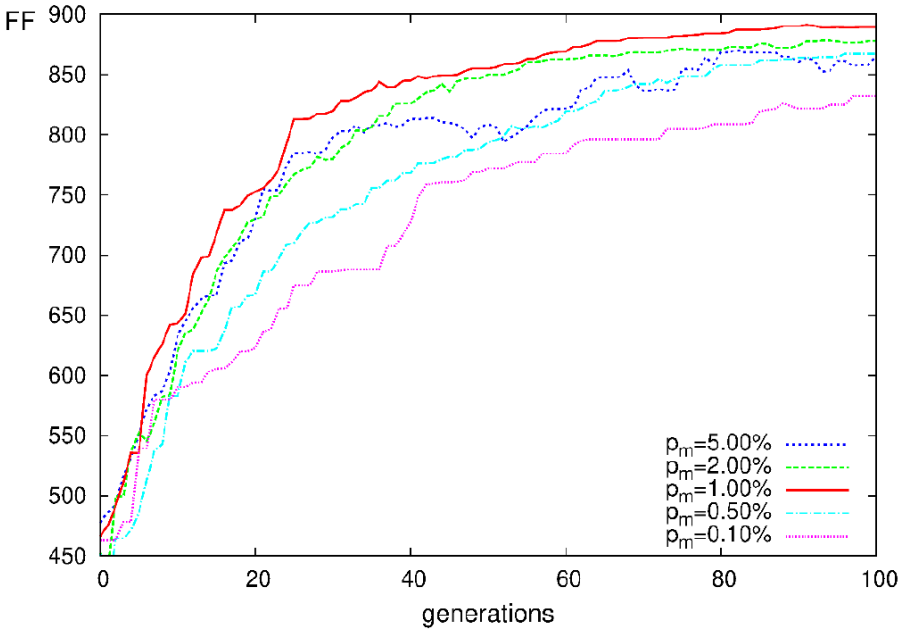


Figure 3.10: Impact of mutation probability (p_m) on the algorithm's performance.

Figure 3.10 compares the FF for different mutation probabilities. The fastest and greatest increase in FF for the studied problem is seen for $p_m < 1.0\%$. Below this value, the mutation has too little impact on the simulation (provides too little diversity in the population) and GA performs worse. Above 1.0%, the level of mutation begins to disrupt GA operation and destroy

potentially good chromosomes, also achieving a worse result. Finally, $p_m = 1.0\%$ was chosen for further simulations.

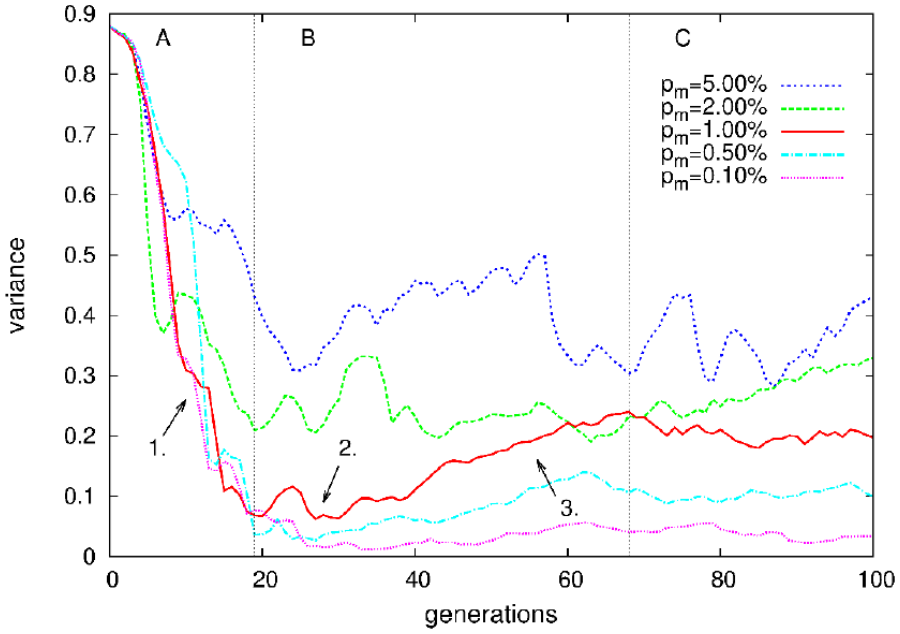


Figure 3.11: Evolution of genetic variance for different mutation probabilities.

Figure 3.11 shows the genetic variance (Section 3.2.2) for different mutation probabilities. It is visible that for simulations where the mutation is not too high and does not introduce too much oscillation in diversity ($p_m < 2\%$), three stages of the entire process can be distinguished:

A - the algorithm tries to create the best chromosome from the initial population of random patterns. It quickly gets rid of the worst combinations, resulting in a drastic decline in diversity. In this part, crossover plays an important role and exceeds the impact of mutation. The curves are therefore very similar regardless of the value of p_m. Usually between the 10th and 20th generation (point 1 in Figure 3.11) a strong chromosome appears that quickly dominates the population, causing the genetic variance to drastically drop and reach a local minimum (domination point 2).

B – in this part, both mutation and crossover are important. The population is largely dominated by the best chromosome and its versions created through mutation. These versions crossover with each other and create better solutions. Usually, after about 20-30 generations, the best chromosome has the largest share in the population, and then, as a result of further mutations and crossover, the genetic variance increases (point 3) to a certain saturation level.

C - in this part, the population is mainly based on one, partially optimized chromosome. Crossover is less important in this part, as the chromosomes are very similar to each other. Only mutation plays a key role and maintains genetic variance at a certain constant (apart from some fluctuations) level, depending on the mutation probability.

Required Number of Generations (n)

Usually the genetic algorithm may search for a solution until:

1. a generation is dominated by one individual,
2. the average/maximum fitness function value does not change for a selected number of iterations,
3. the genetic variance (Equation 3.1) or another indicator describing the diversity of the population falls below a specified value,
4. the algorithm performs a predetermined number of generations

To systematize computational time and easily compare results, it was decided to use a constant, predetermined number of generations counted by the algorithm (option 4). However, it was necessary to determine the order of magnitude so as not to prematurely end the algorithm's work, which could disrupt the assessment and comparison of scenarios. On the other hand, due to the significant computational time (several hours for each Case), it was necessary to limit the number of simulations.

To determine the optimal number of generations in terms of the achieved solution and required computational time, 1000 generations of the algorithm were conducted for Case 1 (FF in the form of 3.2), for 100 chromosomes in the population, $p_m = 1.0\%$ and $\sigma_{min} = 1.0$.

As it can be seen in Figure 3.12, the fitness function grows very quickly up to about 100 generations and saturates, while the variance stabilizes (with small oscillations related to the mutation). After this time, growth begins to weaken and after about 400 generations, an individual close to the optimum is found, and further changes in FF are very small (about 0.2% after additional 600 generations). To show how chromosomes evolve, especially in later generations, Figure 3.13 presents the best individuals for selected generations. After 434 generations, the algorithm reaches the best solution, in which fuel assemblies type 5 (FA5) dominate, as predicted, but inside the core, there are still 28 assemblies of type 6 (FA6), which differ from FA5 only by six additional BA rods (see Table 3.1). Despite these differences, the cycle length is

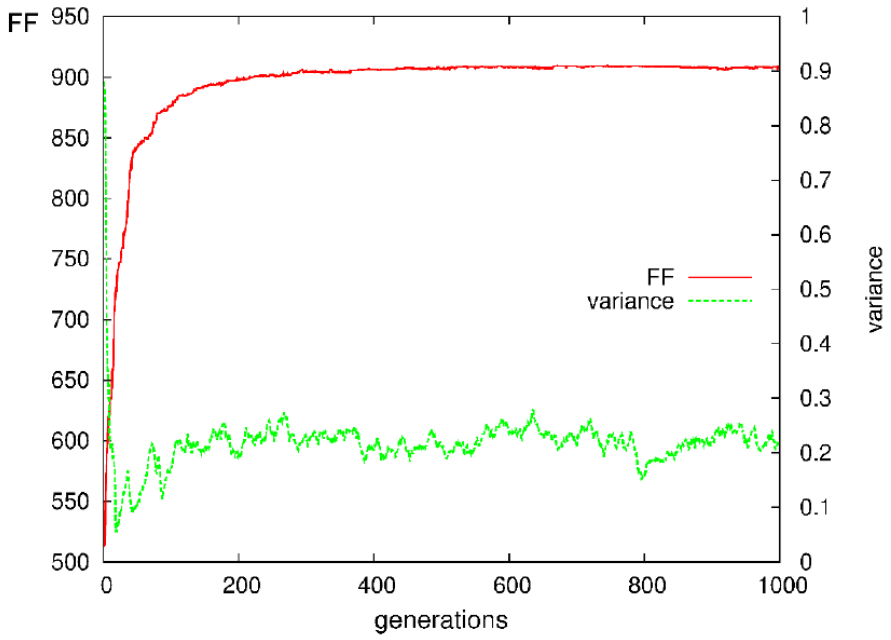


Figure 3.12: Evolution of the fitness function and genetic variance for 10^3 generations.

shorter compared to the optimal (containing only FA5) by about 0.1% (548.74 vs 549.36 days). Further analysis showed that replacing FA6 with FA5 in the center of the core (region where FA6 occurs), changes the FF by only about 0.02%. It can be noticed that when the problem is limited to only FA5 and a few FA6 in the center of the core, the algorithm significantly slows down due to the small distinguishability of these assemblies in later stages of the fuel cycle. Despite further iterations, changes in the cycle length are negligible.

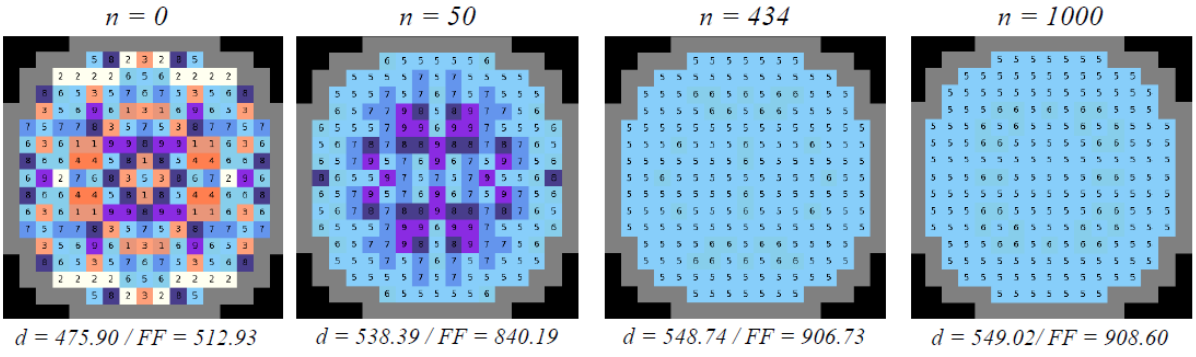


Figure 3.13: Evolution of the loading scheme for selected generations, from $n = 0$ (initial random population) to $n = 1000$. Below each configuration, its corresponding length of the cycle (d) and the FF value are printed.

Based on this test, it can be suspected that in order to solve the optimized problem with selected parameters, about 400-500 generations should be performed (for cases assuming 1/4 symmetry). However, due to the computational time and the demonstrative nature of this work,

it was decided to perform 100 generations for cases 2, 3, 5, and 6 (1/4 symmetry) and 750 for case 4 (without symmetry).

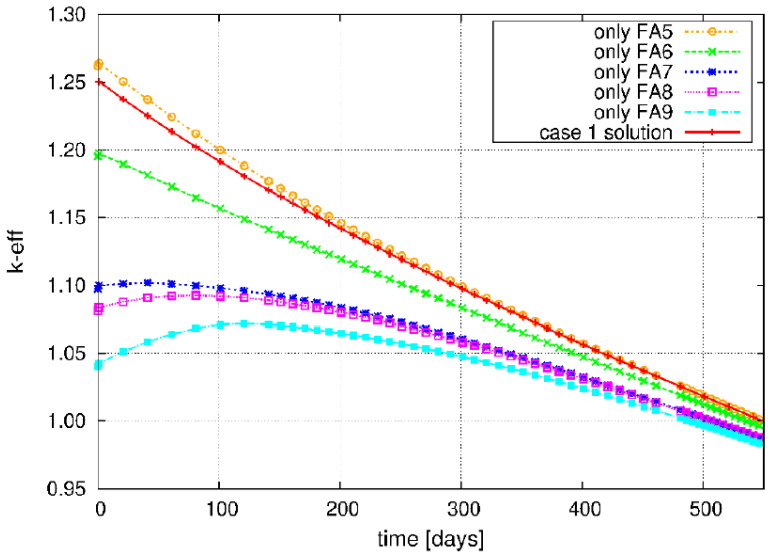


Figure 3.14: Evolution of k_{eff} for cores composed of single types of assemblies and the result for Case 1 (optimization of the length of the cycle).

Results of PARCS calculations for cores with only one fuel assembly typet: FA5-FA9 (Table 3.1) are presented in Figure 3.14, compared to the solution of Case 1 for 1000 generations (see Section 3.2.5). Convergence studies conducted for thousands of chromosomes showed that the solution with the longest possible cycle for Case 1 is a core with maximum amount of fuel, so composed only of 3.1% enriched assemblies and without BA (FA05). Thus, the preliminary and intuitive assumption discussed in Section 3.2.3 was confirmed for the studied core. Furthermore, several interesting neutron effects can be observed. Firstly, for a single-module core, the entire reactor core behaves almost like a single assembly with some negative reactivity caused by leakage (see Figure 3.14). Secondly, for the studied cores, the average burnup at EOC is 22 GWd/tHM, which is a state where most of the BA are burnt, but their influence is still present. For assemblies used in the first BEAVRS cycle, the limit burnup is about 26 GWd/tHM [127]. In Figure 3.14, it can be observed that the core (or assembly) with only FA05 dominates over the others until the consumption of all absorbers. Then, the core enters a regime where all assemblies with 3.1% are practically indistinguishable. As a result, for the studied problem, there is a possibility of multiple equivalent or indistinguishable optimal solutions, and it can be challenging to evaluate the algorithm at high burnup region. There is still one practical problem that occurs for Case 1 and for the studied algorithm. The

solution is presented in the form of the red curve in Figure 3.14. The difference in cycle length between the optimized core (chromosome) and the core with only FA05 is about 0.34 days, which constitutes a 0.06% difference, which is negligible due to statistics and nuclear code uncertainties. It should be noted that the GA asymptotically approaches the solution, and the applied number of generations (1000) allowed the algorithm to achieve a solution characterized by the expected cycle length with such accuracy.

3.3 Results

3.3.1 Statistical Analysis of Simulation Process

Since GA is stochastic, each of its realizations may lead to slightly different result. To check how random the operation of the algorithm is and how repeatable is the result, it was decided to run each simulation multiple times, using the same scenario and the same parameters. For this purpose, optimization for Case 1 was performed 8 times ($N = 100$, $n = 100$, $p_m = 1.0\%$, $\sigma_{min} = 1.0$). Figures 3.15 and 3.16 present the statistical values of the fitness function and genetic variance. The Figure shows the average value of the fitness function and genetic variance, for asymmetric standard deviations for values above the average (σ_u) and below the average (σ_d). Additionally, the charts also show the minimum and maximum values obtained for all 8 simulations. Analyzing the results, it can be noticed that although the course of optimization may vary significantly for selected realizations, the standard deviation of values significantly decreases with generations both for FF and variance and for FF decreases after 100 generations to values below 0.5%. It can also be observed that all simulations converge towards the same value. Greater variability characterizes the genetic variance, however, this variance does not directly affect the operation of the algorithm, it only serves to monitor its behavior. Similar tests should be conducted for all cases. Each simulation should be repeated multiple times to avoid obtaining a random result that significantly deviates from the norm. However, due to computational time, it was decided to assume that simulations of other cases are characterized by similar behavior. Therefore, for each subsequent case, only one final simulation was conducted. Also, the asymmetry of standard deviations of FF in Figure 3.15 may suggest that statistically speaking, an exceptionally bad realization is more likely to occur than an exceptionally good one. This conclusion should be verified with additional tests, however, it may suggest that if a single realization deviates significantly from the norm, statistically it will be one of the worst (pessimistic realization) rather than a randomly good one.

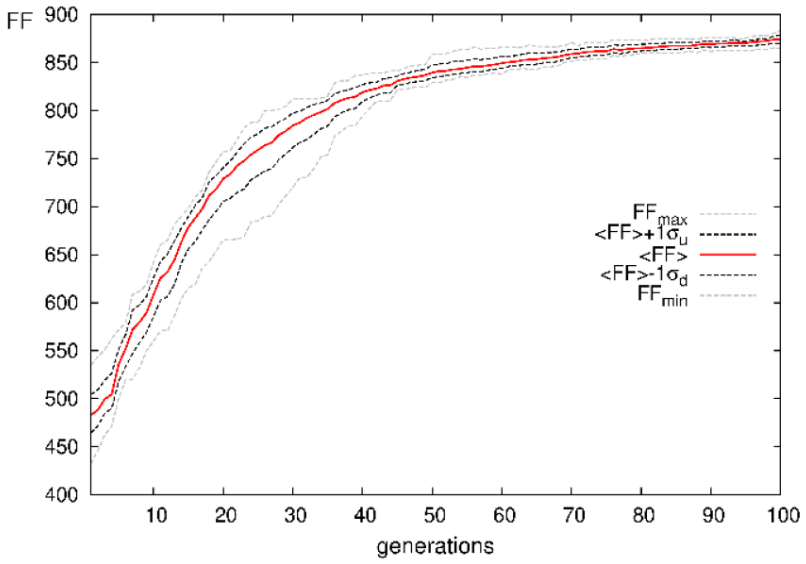


Figure 3.15: Reproducibility of the optimization - Fitness Function (Equation 3.4). Simulations for $N = 100, n = 100, p_m = 1.0\%$.

3.3.2 Cycle Extension with 1/4 Symmetry

This section presents the optimization results for scenarios where the main goal was to extend the cycle, assuming 1/4 symmetry (Cases 1-3).

Case 1 did not impose any restrictions on the choice of fuel assemblies and was a test case. In case 2, certain maximum values were imposed on k_{eff}. In case 3, the algorithm aimed to simultaneously maintain the mass of fissile elements (average enrichment in the core). Figure 3.17 shows the evolution of genetic variance and normalized FF in subsequent generations for all three cases. The chart shows that in all cases, the genetic variance converges to a saturation value of 0.15-0.25 within about 15 generations, while the fitness function grows slowly throughout the simulation area. Interestingly, regardless of the form of FF and the imposed constraints, the convergence of FF and genetic variance has a very similar shape. Therefore, it can be expected that simulations performed earlier only for case 1, should reflect (to some extent) the characteristics of other cases. Then, the selected optimal operating parameters are likely also suitable for cases 2 and 3. Examples of the best chromosomes formed after a certain number of generations are presented in figure 3.18. Table 1 contains a brief overview of the optimized configurations for cases 1-3.

In Case 1, as expected, the algorithm eliminates less enriched assemblies and replaces them with those that have more fissile material. The algorithm successfully replaces all less enriched assemblies, but 100 generations are not enough to reach the final solution with only FA5. However, the result obtained after 100 generations is 544.09 days and is only 0.90% lower

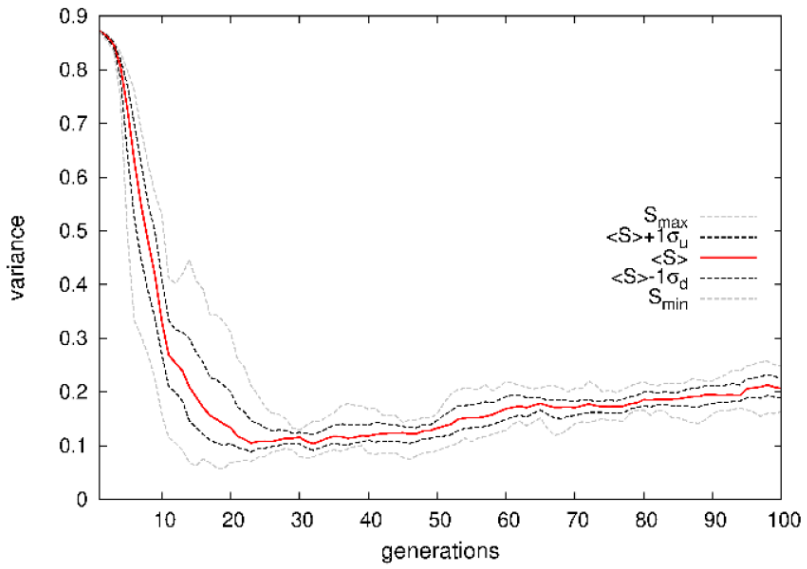


Figure 3.16: Reproducibility of the optimization - Genetic Variance (Equation 3.1). Simulations for $N = 100, n = 100, p_m = 1.0\%$. $\sigma_{min} = 1.0$

than the reference solution for 1000 generations (549.02 days, see Section 3.2.5) and 1.0% lower than the optimal solution (549.36 days). Consequently, it can be stated that the algorithm works satisfactorily even for a significantly reduced number of generations.

In Case 2, for subsequent patterns, we can observe (Figure 3.18) that the algorithm adds assemblies with 3.1%, especially FA09, which also has the largest population of BA. This is consistent with known physics as the presence of BA reduces k_{eff} at the begining of the cycle, allowing at the same time the use of more mass of fissile material and prolonging the operation of the core. Observing Figure 3.18, it can be postulated that the optimal solution is a core built with only FA09 type, for which the length of the cycle is 488.49 days, where the found solution is 483.34 days (1% difference). However, the k_{eff} of a core made of FA9 only, exceeds the maximum permissible value ($k_{eff} = 1.070$) by a small reactivity margin of 175 pcm (Δk_{eff} = 0.002). Therefore, the core made of FA9 only is not the optimal solution as does not fulfill the constraints for this case and the algorithm looks for more complicated loading patterns.

3.3.3 Cycle Extension with Lack of Symmetry

Case 4 is special because it involves optimization of the entire core geometry and does not impose any symmetry. Due to the increased search space, the considered case requires a larger number of generations to reach a feasible solution. Simulations were therefore conducted for 750 generations (7.5 times more compared to other scenarios). Figure 3.19 shows the evolution of genetic variance and the fitness function over successive generations. It can be noticed

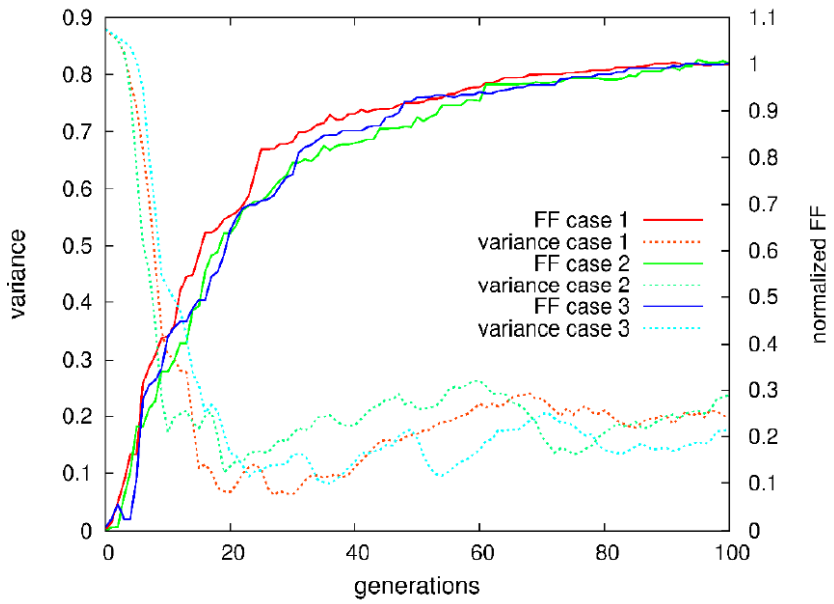


Figure 3.17: Optimization progress for Cases 1-3.

Case	fitness function	Cycle length [days]	Average enrichment [%]	Total number of BA rods	$max(k_{eff})$
1	$(d/\alpha)^{\beta}$	544.09	3.09	1024	1.192
2	$d^{\beta} \cdot \left[\alpha_0 \cdot \left(1+\frac{k_1-max(k)}{k_2-k_1}\right)^{\gamma}\right]^{-\beta}$	483.34	3.05	2040	1.070
3	$d^{\beta} \cdot \left(\alpha_0 \cdot \left[1+\left(\frac{E-E_0}{E_0}\right)^2\right]^{\gamma}\right)^{-\beta}$	426.71	2.46	288	1.204
BEAVRS	–	333.65	2.31	1268	1.080

Table 3.3: Optimization results for Cases 1-3, $N = 100$ chromosomes in the population, $n = 100$ generations, $p_m = 0.01$, compared with parameter values for the original BEAVRS model.

that the genetic variance converges to a saturation value slightly higher than 0.0, over about 50 generations. The fitness function, on the other hand, converges slowly over the entire simulation area, but its most significant increase is observed up to 250 generations. The fitness function grows slowly, often falling into local extremum windows, which can last dozens of generations. It is clear that the lack of assumed symmetry significantly complicates the analyzed problem. The problem of arranging 193 fuel assemblies of 9 types in a full core compared to the problem of arranging these assemblies in 1/4 of the core (56 elements – 1/4 of the core along with the central column and row), increases the solution space by about $9^{193}/9^{56} \approx 5 \cdot 10^{130}$ times. It can be expected that such a complex problem would probably require thousands of generations, which, due to computational time, could not be realized. The best chromosomes for selected

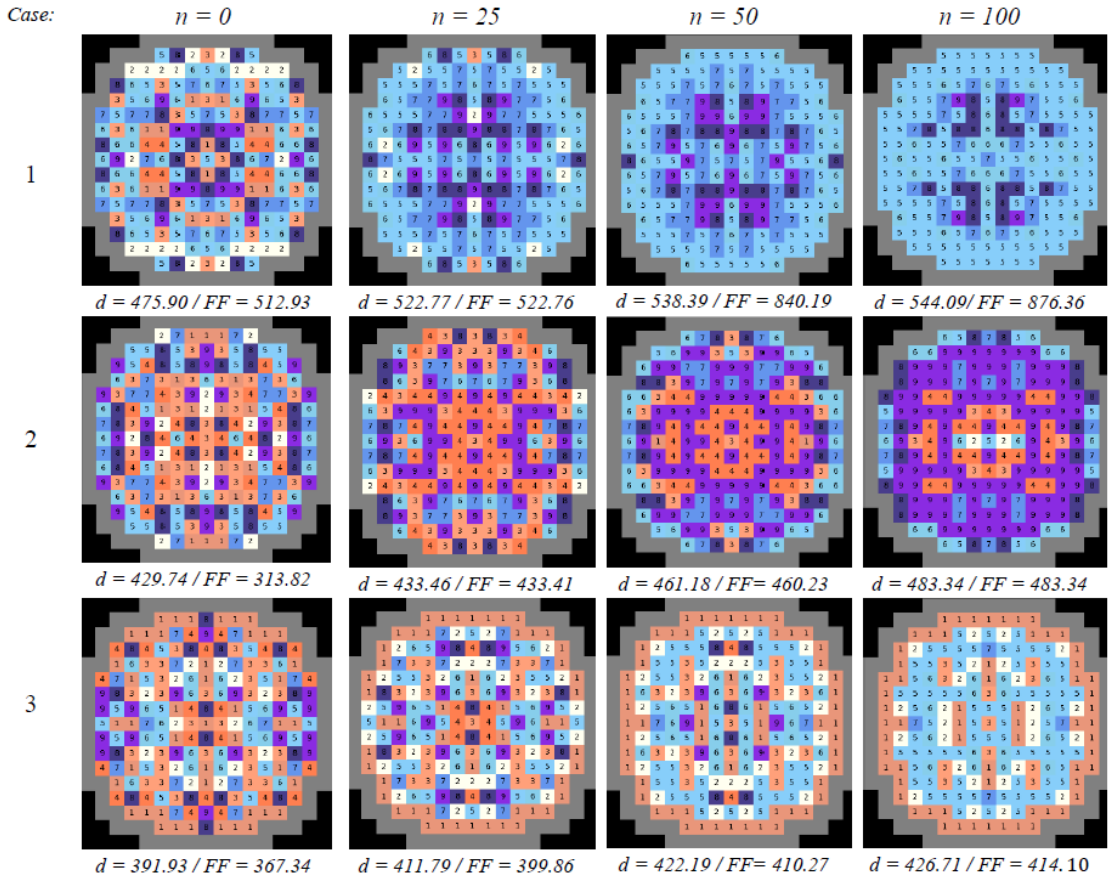


Figure 3.18: Optimization progress for Cases 1-3, for generations from $n = 0$ (initial random population) to $n = 100$ (final generation). Below each configuration, its corresponding length of the cycle (d) and the FF value are printed.

generations are presented in Figure 3.20. In the figure, it can be noticed that the symmetry is broken. However, it can be observed that after 250 generations, the assemblies begin to arrange themselves in a ring. Assemblies with higher enrichment (mainly assemblies FA5, FA6) migrate to the inner part of the core, while those with lower enrichment (mainly FA1), are pushed to the boundary of the core. It can be noticed that this leads to the beginning of the formation of non-random shapes. Over generations, subsequent configurations differ less and less from each other, and the algorithm slows down. After 750 generations, no significant changes in configuration are visible, and the FF increases more slowly. It is evident that due to the size of the possible solution space, the considered optimization (without imposed symmetry) would require a significantly larger number of generations. It may be surprising that despite the incomplete convergence of the algorithm and the lack of final symmetry, the cycle length of the configuration proposed after 750 generations is 374.87 cycle days, i.e., 12.3% (41 days) more than in the case of the reference BEAVRS model. However, it should be remembered that the

considered case was only intended to check the influence of symmetry on the convergence and operation of the algorithm. It assumed therefore the simplest form of the fitness function (3.2), focusing only on cycle length. Of course, the increase in the fitness function shows the correct operation of the algorithm. However, it did not converge to a specific solution, showing a definite influence of the assumed symmetry on the solvability of the problem.

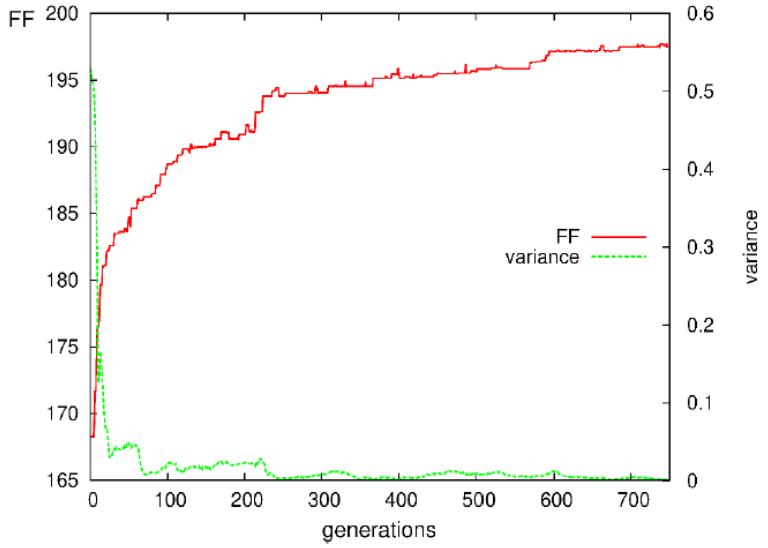


Figure 3.19: Optimization process for Case 4.

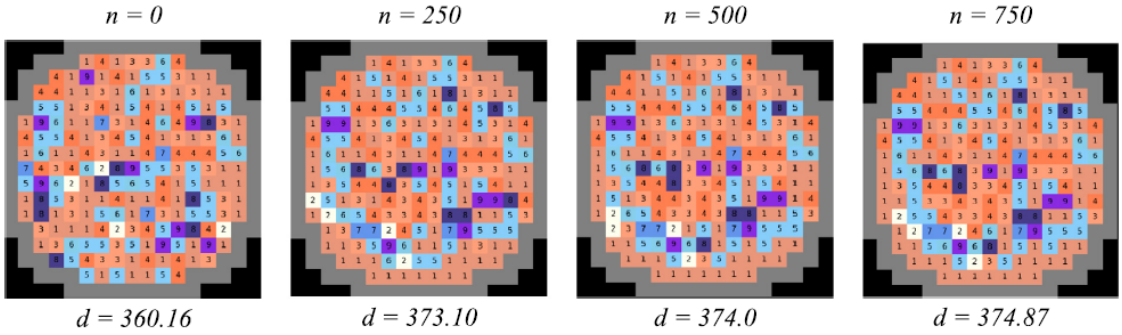


Figure 3.20: Optimization process for Case 4.

3.3.4 Power Distribution Optimization

In this part of the study, optimizations for Cases 5 and 6 were conducted, aimed at reducing the unevenness of power distribution, by minimizing the PPF coefficient (Equation 1.24). For this purpose, FF 5 and 6 (equation 3.10 and 3.11) were used. Case 5 involved only minimizing the PPF while Case 6 aimed also at extending the cycle length. The optimization process is shown in Figure 3.21.

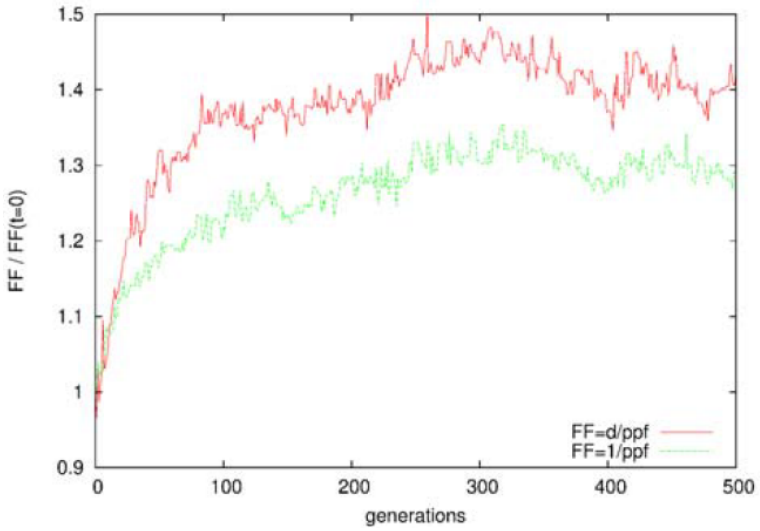


Figure 3.21: Optimization process for Case 5 and Case 6.

In Cases 5 and 6, it was decided to perform more generation (500 vs 100 in Cases 2-4) of the genetic algorithm. For case 5 (green line in Fig. 3.21) the fitness function increased by about 35% compared to the best individual randomly generated in the initial population. For case 6 (red line), it was about 50%. Figure 3.22b,c shows a comparison of the obtained configurations with the BEAVRS core configuration (Figure 3.22a). The figure shows that assemblies with higher enrichment are placed at the boundary of the core for each configuration to minimize flux drop. Then in the center, assemblies with different enrichment are alternately placed, usually containing a larger number of BA rods to flatten the flux distribution in the central part of the core. Characteristics of the obtained configurations and the BEAVRS core are presented in Table 3.5. It includes the indicator of the amount of fissile material used (average enrichment calculated as Equation 3.8), the number of BA rods, initial k_{eff}, maximum PPF value in the cycle, and length of the cycle. Since PPF was the optimized parameter for both cases, the power distribution at the beginning (BOC) and end of the cycle (EOC) for BEAVRS and cores obtained by the algorithm are plotted (Figure 3.23a-f). As can be seen, the power distribution for BEAVRS is characterized by an initial peak in the center of the core, which spreads towards the outer boundary throughout the cycle. The maximum value of P_{xy} is 1.33 at BOC and 1.21 at EOC. In the configurations for optimized cases, higher values of P_{xy} are distributed throughout the core, resulting in a relatively flat power distribution. In Case 5, the maximum value of P_{xy} is 1.26 both at the beginning and at the end of the cycle. In Case 6, the maximum initial value of P_{xy} is 1.34, which drops to 1.17 at EOC. Table 3.4 presents a comparison of these values.

	BEAVRS	Case 5	Case 6
$max(P_{xy})$, BOC	1.33	1.26	1.34
$max(P_{xy})$, EOC	1.21	1.26	1.17

Table 3.4: Maximum values of P_{xy} for the beginning (BOC) and the end (EOC) of the cycle for Case 1, Case 2, and the BEAVRS model.

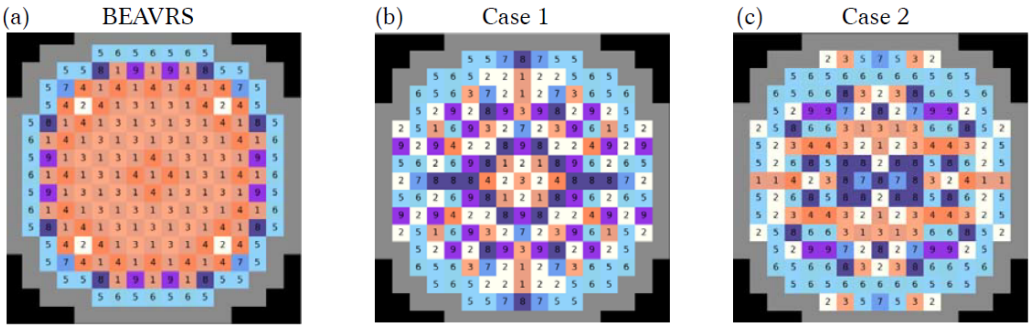


Figure 3.22: Optimization results for Cases 5 and 6.

As the result of the simulations, two configurations were obtained, characterized by a longer fuel cycle compared to the original configuration. In Case 5, the maximum PPF value was reduced from 1.88 to 1.82, and the cycle was extended by 118 days. In Case 6, a slightly higher PPF value (1.89) was obtained, and the cycle was extended by 179 days. Of course, other core parameters also changed. Cores in Cases 5 and 6 use more fissile material (average enrichment 2.76% and 2.78% vs 2.36% in the BEAVRS case), thus also have a higher initial k_{eff} (both 1.13 vs 1.08). The larger initial mass of fissile material is the main reason for the longer cycles compared to the BEAVRS core. Additional limitations in the form of allowable k_{eff} or fissile mass will be the subject of future research. However, the above examples demonstrate the effective operation of the algorithm. Within the simulations, it was possible to maximize a two-parameter fitness function, i.e., flattening the power distribution and extending the cycle. To achieve a comprehensively optimized configuration that could be applied in a nuclear power plant, future simulations may include more complex FF that would consider other limitations, such as maximum average enrichment, acceptable k_{eff} range, control rod value, reactivity coefficients, and/or other parameters, optimizing the core using multiple goals at the same time.

Case	fitness function	Average Enrichment [%]	Number of BA Rods	Initial k_{eff}	PPF_{max}	Cycle Length [days]
5	$\left(\frac{1}{PPF}\right)^{\beta}$	2.76	1392	1.13	1.82	451.7
6	$\left(\frac{d}{PPF}\right)^{\beta}$	2.78	1356	1.13	1.89	512.5
BEAVRS	-	2.36	1268	1.08	1.88	333.6

Table 3.5: Optimization results for cases 5, 6, $N = 100$ chromosomes in population, $n = 100$ generations, $p_m = 0.01$, compared with parameter values for the original BEAVRS model.

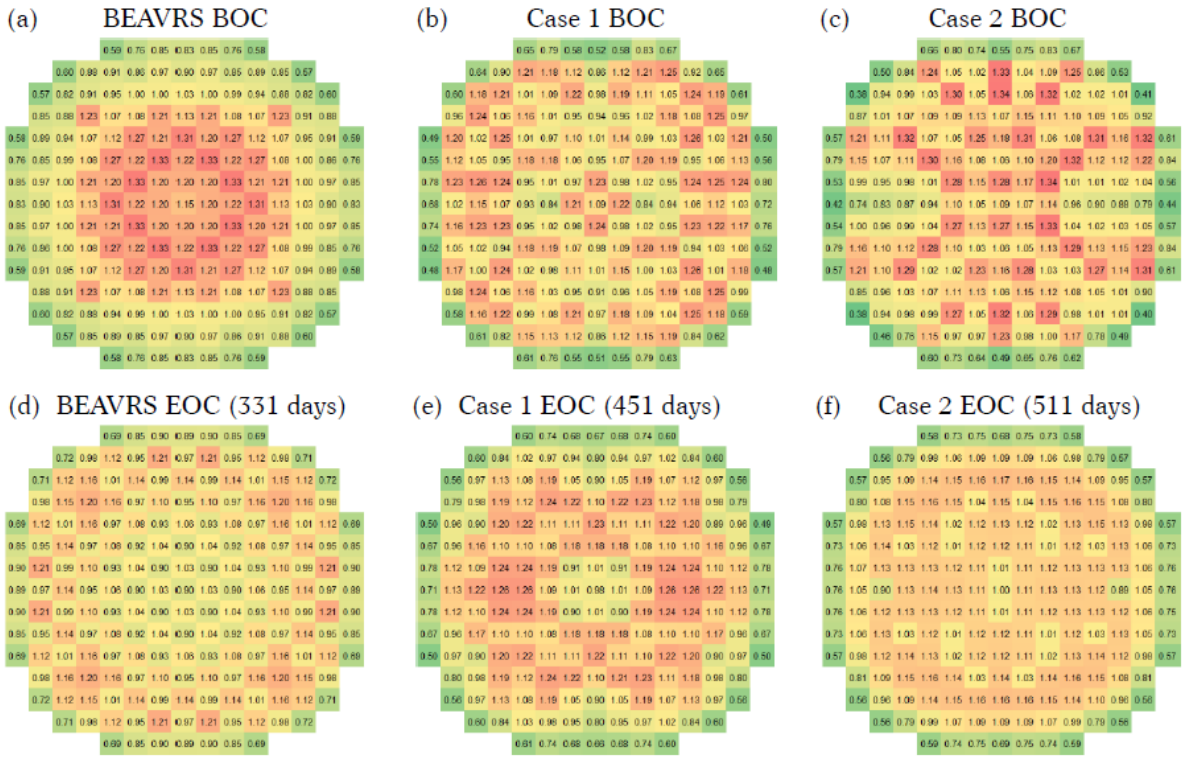


Figure 3.23: Optimization results for Cases 5 and 6.

4 Optimization of the Multi-Cycle Problem

4.1 Description of the problem

In this part, the Equilibrium Cycle (EC) was subject to optimization, which is a general scheme for shuffling the core after each cycle. It is a universal scheme used for analyses during the design stage when it is necessary to assume the long-term properties of the core. Unlike a single refueling, in the EC only the Fresh Fuel Assemblies (FFA) matter, their distribution, and how they are transferred from cycle to cycle. The rest of the assemblies are in the core only in the initial few cycles, then they are removed and only FFA and FFA placed in the core in previous cycles remain. Although in practice refueling is usually planned from cycle to cycle, the optimization of EC is important because of the possibility of finding general trends in the shuffling of assemblies, on which single refueling is usually based. In this study, the algorithms used were expanded and, in addition to Genetic Algorithms (GA), Parallel Simulated Annealing (PSA) was also applied, often used for the problem of core loading optimization. Additionally, a hybrid method combining GA and PSA (PSA+GA) was proposed, utilizing selection and crossover from GA as well as the cooling scheme from PSA. This part of the study was conducted during a six-month visit to the Data Science and Applied Mathematics Pole, at the Framatome company headquarters.

4.1.1 Linear reactivity model and core refueling

It has been noted that for a fuel assembly or lattice of a typical PWR, particularly when the fuel enrichment exceeds 2.0%, the k value (derived from the neutron transport Equation 1.2), and reactivity ρ, change almost linearly with burnup [145]. Specifically, Sefcik [146] demonstrated that for the Maine Yankee supercell reference lattice, with uranium enrichment ranging from 1.50% to 4.34%, linear least squares curve-fits of ρ as a function of burnup (for the range 150-50,000 MWd/t) resulted in an average coefficient $R^2 = 0.995$. This indicates that only 0.5% of the variance is unexplained by the correlation.

Figure 4.1 presents the evolution of ρ as a function of burnup for the Maine Yankee supercell [147] for fuel enrichments from 2% to 5%. Although it is not illustrated in the Figure 4.1, there is a sudden drop in reactivity at the BOC, within the first 150 MWd/t [148] caused by the build-up of xenon and samarium fission product poisons reaching saturation levels. Outside this

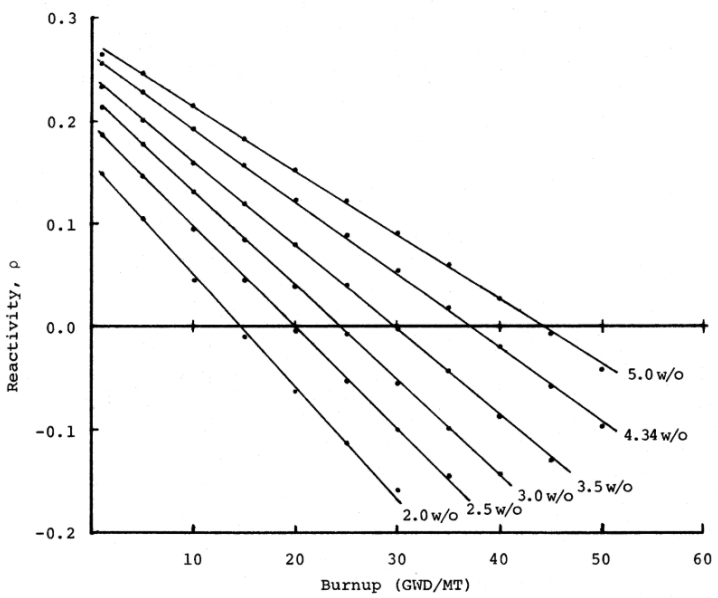


Figure 4.1: Reactivity evolution in the Main Yankee Supercell for different fuel enrichments, taken from [146].

initial transient, the reactivity evolution can be approximated within the linear model [148]:

$$\rho(BU) = \rho_0 - a \cdot BU, \tag{4.1}$$

where ρ_0 is the initial excess reactivity, a is the slope of the curve (t/MWd) and BU is the burnup (MWd/t).

Impact of the number of reloads on core reactivity and final burnup

Using the linear reactivity model, we consider the discharge burnup ($B_{\text{dis,n}}$) for an n-batch core operating under steady-state refueling conditions, assuming constant thermal power of the core with uniform power distribution across fixed-size fuel batches. Then the freshest batch undergoes a burnup process to $(B_{\text{dis,n}})/n$, followed by the subsequent older batch reaching $(2B_{\text{dis,n}})/n$, and so on. The system reactivity is the average reactivity calculated over the set of fuel batches at the end-of-cycle (EOC) [148]:

$$\rho_i(EOC) = \rho_{0,n} - a \cdot \frac{i \cdot BU_{dis,n}}{n}. \tag{4.2}$$

and system reactivity:

$$\rho_{sys} = \frac{1}{n}\sum_{i=1}^{n}\rho_i = \frac{1}{n}\sum_{i=1}^{n}\left(\rho_{0,n} - a\cdot\frac{i\cdot BU_{dis,n}}{n}\right). \tag{4.3}$$

Taking into account that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$:

$$\rho_{sys} = \frac{1}{n}\left(n\cdot\rho_{0,n} - a\frac{n(n+1)}{2}\cdot\frac{BU_{dis,n}}{n}\right) = \rho_{0,n} - a\frac{n+1}{2n}BU_{dis,n}. \tag{4.4}$$

At the End of Cycle (EOC), when the system reactivity (ρ_{sys}) equals zero due to reactivity limitations, Equation 4.4 can be written as:

$$BU_{\mathrm{dis},n} = \frac{\rho_{0,n}}{a},$$

and for a single-batch $(n = 1)$:

$$BU_{\mathrm{dis},1} = \frac{\rho_{0,1}}{a}.$$

When a single-batch core achieves the same discharge burnup as an n-batch core ($B_{\mathrm{dis},1} = B_{\mathrm{dis},n}$), the reload reactivity (so the reload enrichment) of the n-batch core will be smaller than the one batch core:

$$\rho_{0,n} = \frac{n+1}{2n}\rho_{0,1}. \tag{4.5}$$

On the other hand, if the reload reactivity stays the same ($\rho_{0,n} = \rho_{0,1}$), an n-batch core can achieve a higher burnup:

$$BU_{\mathrm{dis},n} = \frac{2n}{n+1}BU_{\mathrm{dis},1}. \tag{4.6}$$

From Equation 4.5 and Equation 4.6 one can see that the higher the number of batches is, the bigger the final burnup (or smaller initial reactivity). For large numbers of batches (continuous discharge), the burnup tends to be twice as high as for $n = 1$ batch, while the initial necessary excess reactivity is half the value [148].

Optimal number of fuel reloads

Using the linear reactivity model and formula in Equation 4.6, we can conduct a simplified economic analysis of the number of reloads during a cycle. If we denote $BU_{dis,1}$ in units of *EFPD* (*Effective Full Power Days*), then the burnup (expressed in days) can be treated as the total en-

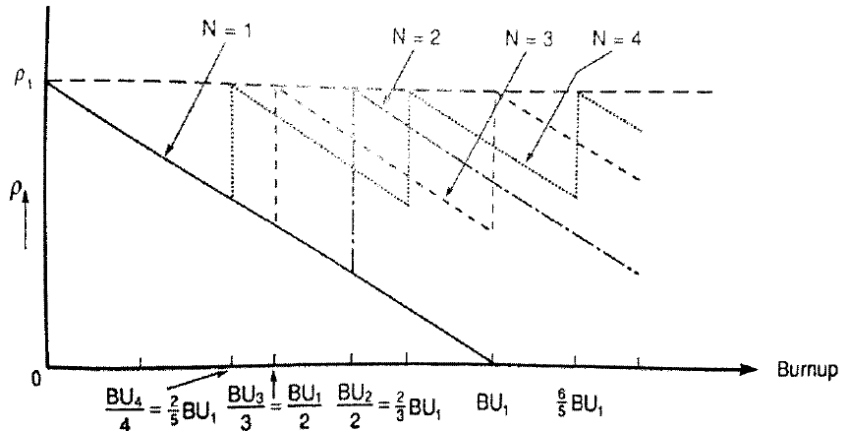


Figure 4.2: The impact of the number of fuel reloads (cycles) on the evolution of the core reactivity, taken from [149].

ergy production (the number of days the reactor operates at full power), for one batch. We can also denote BU_{break}, determined in *EFPD*, as the number of full power days lost due to the time required for each reload. Then, consequently, the reactor effectively operates (produces) energy for $BU_{dis,1} - BU_{break}$ full power days in case of one batch and $BU_{dis,n} - n \cdot BU_{break}$, for n batches and n reloads between them. If we consider the linear reactivity and the final burnup depending on the number of matches described in Equation 4.6, we can define the effective number of days of energy production as:

$$BU_{eff}(n) = \frac{2n}{n+1} BU_{dis,1}[EFPD] - n \cdot BU_{break}[EFPD]. \tag{4.7}$$

This is a function of the number of reloads n within a cycle, for which one can seek an optimal value, given a specific reference burnup for a single batch and the number of days required for reloading. Modern PWR (Pressurized Water Reactor) types are typically loaded in three batches, achieving a cycle length in the range of 500 EFPD, which is equivalent to 500 days of operation at full power. Using the liner reactivity model it corresponds to a max burnup for one batch $BU_{dis,1} \approx 333 EFPD$ (Equation 4.8), to which we can conduct a comparative analysis.

$$BU_{dis,3} = \frac{2 \cdot 3}{3+1} BU_{dis,1} \implies BU_{dis,1} = \frac{2}{3} \cdot 500 EFPD \approx 333 EFPD. \tag{4.8}$$

Figure 4.3 presents the values of the effective number of days of energy production calculated with Equation 4.7, for the defined reference $BU_{dis,1} = 333 EFPD$, for different numbers of batches and the 14, 28, and 42 days required for fuel reloading. From the figure, we can observe that with reload times ranging from 14 to 42 days, the optimal number of batches is 3

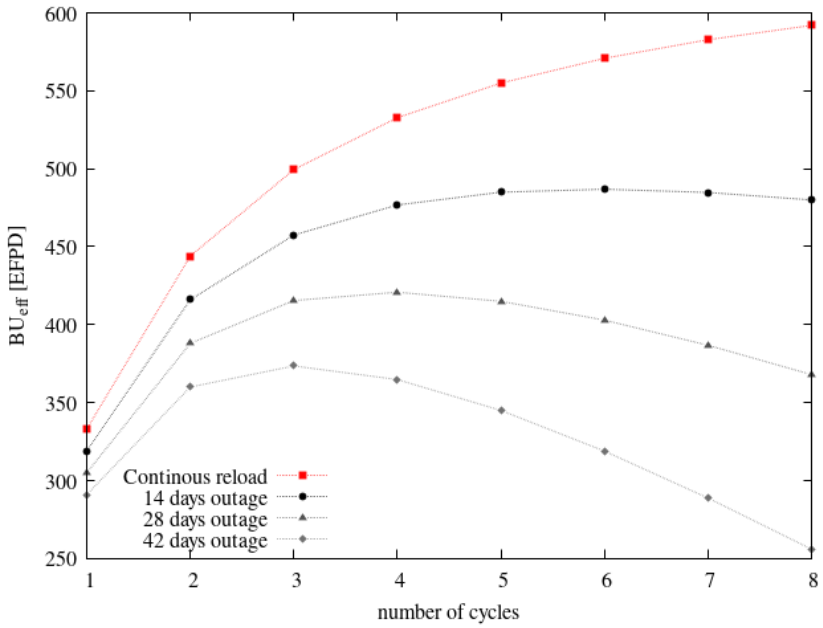
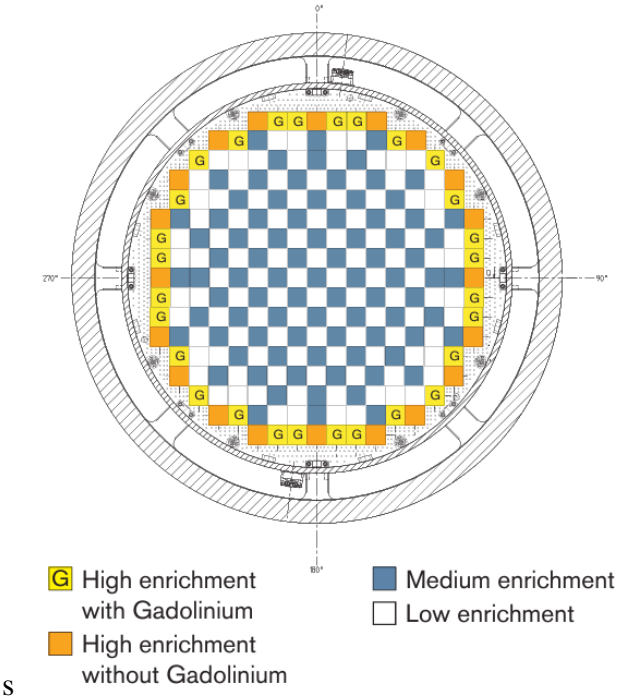


Figure 4.3: Effective days of energy production (Equation 4.7) for a reference $BU_{dis,1} = 333 EFPD$ for different numbers of batches and various reload times (14, 28, and 42 days).

to 5. Despite the very simplified model, these results are confirmed in practice, as most of the modern reactors operate with 3, and at most 4, batches. However, it is important to remember that in addition to economic considerations, significant roles are played by physical properties, such as material endurance against the neutron field and high temperatures, maximum burnup, and the maximum time a fuel assembly can remain in the core. Then, the choice of the number of batches, and consequently the number of reloads during cycles, may be significantly more limited by other factors.

4.1.2 Framatome EPR Core

The part of the study is based on the European Pressurized Water Reactor (EPR) [150], a design of the third-generation, pressurized water reactor, developed by Framatome, EDF (*Électricité de France*), and Siemens, based on the French N4 and the German KONVOI reactors [151]. Originally referred to as the European Pressurised Reactor within Europe, its name evolved internationally to Evolutionary Power Reactor [152] and is currently referred to as simply EPR.



s

Figure 4.4: Typical loading of EPR [153]

Parameter	**Value**
Core thermal power	4300 MWth
Net power output	$\sim$ 1600 MWel
Reactor coolant system	
Number of loops	4
Operating pressure	155 bar abs
RPV inlet/outlet temperature	295.9 / 327.2 °C
Total coolant flow per loop	28330 m³/h
Core	
Number of fuel assemblies	241
Number of control rod cluster assemblies	89
Fuel assembly array	17 x 17
Active height	420 cm
Average linear heat rate	156.1 W/cm

Table 4.1: Main specifications of EPR [150]

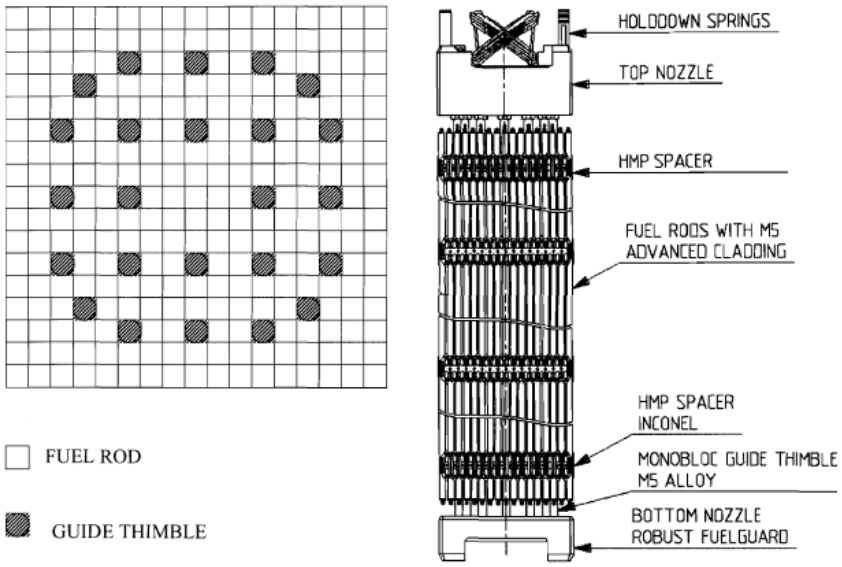


Figure 4.5: EPR fuel assembly design [154].

Typical EPR loading is presented in the Figure 4.4. The EPR core consists of 241 FA, arranged in 17×17 grid. For the first loading, there are four FA groups, two groups with high enrichment (one with BA rods), one with medium enrichment, and one with low enrichment [153]. The fuel rods are an effect of years of engineering optimization and proven design parameters, with enrichment levels of up to 5 wt% ^{235}U. There are 265 rods arranged in the fuel assembly also in a grid of 17×17 [154] (Figure 4.5). Integrated into the fuel matrix is Gd_2O_3, serving as an integral burnable absorber with concentrations varying between 2 wt% to 8 wt% [154].

The considered core is inspired by EPR and reflects its main features (see Table 4.1), but it is not an existing design or real configuration. The EPR-like core used in this study uses 6 different FA types, varying in the enrichment level (from 1.50% to 4.95%) and number of BA rods (from 0 to 24). The model used in this study was provided by the *Department of the Validation of Neutronic Codes and Methods* of Framatome. As the model is used in the Department it has been validated and verified properly, and it was not done in the frame of this work.

4.2 Implementation of the optimization algorithm

4.2.1 Equilibrium cycle encoding

One of the most important parts of the optimization algorithm is to develop a proper transition between the optimized system and its encoded version, manageable by the algorithm. In case of the 1/8 symmetry assumption, the problem needs to be divided into two parts to avoid FA exchange between different symmetry zones (as in the Figure 2.2d). Then, the first intuitive idea would be to assign to each FA at cycle n and to the set of FFA (Fresh Fuel Assemblies), their new positions in the cycle $n+1$. This simple idea is sufficient for the single reload problem [5], [6], but in the case of the EC, due to the random nature of meta-heuristics, such a method would most likely lead to an invalid EC, with loops or too short and too long fuel path (as in Figures 2.2a-c). To tackle this problem, a matrix representation of the EC is proposed in this study.

Matrix representation of the Equilibrium Cycle

In this study an original and simple method of presenting the equilibrium cycle was porposed. As was mentioned in Section 2.1.3, if we consider batch size of k and we assume that each FA can stay in the core maximum m cycles, we can represent EC as a $k \times m$ matrix of fuel assemblies paths:

$$[EC]^{m\times k} = \begin{bmatrix} pos_{FA_1,1} & \dots, & pos_{FA_k,1} \\ \vdots & \ddots & \vdots \\ pos_{FA_1,m} & \dots, & pos_{FA_k,m} \end{bmatrix}^{m\times k} . \tag{4.9}$$

Thus, each path (or the positions of assembly in the subsequent cycles) can be written as:

$$PFA_i = \{pos_{FA_i,1} \rightarrow pos_{FA_i,2} \rightarrow ... \rightarrow pos_{FA_i,m}\}. \tag{4.10}$$

For the EC in the (4.9) form, the way to create a valid random EC was very simple and it just required the generation of a random perturbation of $k \cdot m$ integers. This simple yet very helpful form of EC representation has enabled easy manipulation of various configurations and extremely easy generation of EC. The search for the configurations that have the same properties

of each subsequent cycle, is itself also the subject of research [4].

Obviously, in many cases, $k \cdot m$ (size of the matrix) may be greater than the optimized part of the core. In such a case, it was decided to fill the matrix with negative numbers that were ignored by the algorithm, so paths with a negative position were simply shorter (less than defined m). Figure 4.2 presents an example of a random EC for $k = 13$ FFA and $m = 3$ maximal number of cycles that an assembly can stay in the core. For instance, for $EC_{11} = [29, -1, 36]^T$ in Figure 4.2, position *-1* is ignored, the fuel path $PFA_{11} = \{29 \rightarrow 36\}$ and the assembly will stay in the core for two cycles instead of three.

m \ *n*	FF01	FF02	FF03	FF04	FF05	FF06	FF07	FF08	FF09	FF10	FF11	FF12	FF13
N	20	16	15	21	14	22	23	17	34	33	29	30	35
1	19	10	12	18	13	5	3	6	24	25	-1	32	31
2	1	11	9	2	4	8	0	7	28	27	36	26	37

Table 4.2: Matrix form of the EC. *N* - new fuel assembly, *1* - assembly that spent in the core 1 cycle, *2* - assembly that spent in the core 2 cycles

In this way, one can understand the matrix form of Equation 4.9 as a set of fuel paths in the core, as f.i. in Figure 4.2. A graphical explanation of one of the paths is presented in Figure 4.6. In this example $PFA_3 = \{15 \rightarrow 12 \rightarrow 9\}$ which means that FFA number 3 will be placed in position *15*, after one cycle it will be placed in position *12* then in position *9* and after three cycles it will be removed from the core.

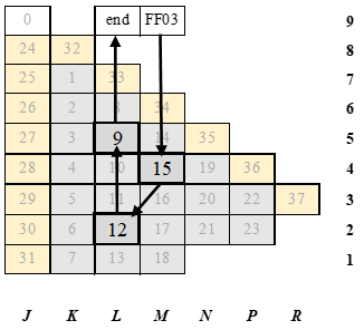


Figure 4.6: Graphical explanation of the fuel path PFA_3.

Naturally, the chromosome could be represented directly as the 2D matrix (as in [37], [5], [6]), however, due to *deap* library [155] that operates on 1D vector, finally each EC was rewritten to a vector of $k \cdot m$ elements (Equation 4.11), and then processed by the algorithm in this form, then:

$$[EC] = [pos_{FA_1,1}, pos_{FA_1,2}, ..., pos_{FA_1,m}, ..., pos_{FA_n,1}, pos_{FA_n,2}, ..., pos_{FA_n,m}]. \tag{4.11}$$

This study aimed at optimization of the 1/8th and 1/4th of the EC configuration, in terms of selected parameters and imposed constraints. Each time we consider a reload of a constant number of FFA, equal to around 1/3 of the total number of all fuel assemblies in the core, each could stay in the core for a maximum $m = 3$ times. There are 38 positions with 13 FFA in the 1/8 of the core and 61 positions with 21 FFA in the case of 1/4 of the core.

4.2.2 Optimization operators

In this section the operators of the optimization are described: the fitness function that describes the goal of the optimization and the direction of the population evolution, the selection operator, which defines the way the new population is created, the crossover operator responsible for the exchange of information between specimens, and the mutation/perturbation operator that allows creating solutions (in case of GA) or moves the solution within the search space (in case of PSA).

Fitness Function

The study's objective was to develop an optimization tool for identifying the optimal EC based on specific parameters. Existing literature mostly focuses on using GA/PSA in nuclear engineering to maximize k_{eff}, cycle length, or minimize power peaking factor (PPF) [73], [6]. However, prioritizing maximum k_{eff} and the longest cycle may not align with safety requirements. Minimizing PPF helps reduce power distribution disparities but it neglects factors like coolant flow, core inlet temperature and pressure which are crucial for accounting for boiling crisis risk. Therefore, in this study, the below-listed optimization goals were considered:

- **Maximize the core's mean burnup** to extend the cycle length,
- **Maintain the hot channel enthalpy rise factor** $F_{\Delta H}$ (Equation 1.25) below a specified threshold (defined as the $F_{\Delta H}$ value of the reference core), aiming at even power distribution and preventing departure from nucleate boiling (DNB),
- **Restrict the maximum burnup** of fuel assemblies $\max(BU_a)$ to meet safety and materials requirements,
- **Limit neutron leakage** $\bar{L}$ to a designated level ($\bar{L}$ value of the reference core), defined as the percentage difference between k_∞ and k_{eff} (Equation 1.26), to prevent FFA from being placed at the periphery of the core and limit vessel fluence.

While this set of parameters does not encompass all EC design requirements due to the complex nature of the task, it was believed that it could form a simple FF, facilitating relatively fast convergence but also providing a reasonable solution.

As previous works state [5], one of the most important parts of the optimization algorithm is the FF that determines the direction of the evolution of the population and the final results. In the frame of this part of the study, two FF forms were considered. First, the test form focused on the extension of the cycle (by maximizing the mean burnup in the core $\overline{BU}_c$) and minimization of the $max(F_{\Delta H})$ value, while keeping the $max(BU_a)$ below the defined threshold. The second form, developed after test simulations, considered all the parameters of the first FF but in addition, it took into account the influence of the neutron leakage $\overline{L}$ and aimed at keeping $max(F_{\Delta H})$ value below the threshold instead of its minimization. Then, FF was defined in the forms of Equation 4.12 (test FF) and Equation 4.13 (main FF):

$$FF_{test} = \frac{1}{c} \cdot \frac{\overline{BU}_c}{\max(F_{\Delta H})} \cdot p_1(\max(BU_a)), \tag{4.12}$$

$$FF = \hat{BU}_c \cdot p_1(\max(BU_a)) \cdot p_2(\max(F_{\Delta H})) \cdot p_3(\overline{L}), \tag{4.13}$$

where $\hat{BU}_c = \frac{1}{a}(\overline{BU}_c - b)$ is the scaled mean burnup of the core, and $a = 2.2GWd/t, b = 15.0GWd/t$, $c = 17.2GWd/t$ are scaling factors such as $FF(ref) = FF_{test}(ref) = 1.0$, whereas $p_i(x)$ are penalty functions:

$$p_i(x) = \begin{cases} 1 & \text{if } x \leq x_{max,i}, \\ e^{-\gamma(x - x_{max,i})} & \text{otherwise}, \end{cases} \tag{4.14}$$

and

$\gamma_1 = 5 \cdot 10^{-2}, x_{max,1} = BU_{a,max} = 60GWd/t,$

$\gamma_2 = 1, x_{max,2} = F_{\Delta H,ref} = 1.48,$

$\gamma_3 = 1 \cdot 10^{-1}, x_{max,3} = \overline{L}_{ref} = 2.62\%,$

where the value of $BU_{a,max} = 60GWd/t$ is a typical fuel assembly burnup operation limit, $F_{\Delta H,ref} = 1.48$ and $\overline{L}_{ref} = 2.62\%$ are $F_{\Delta H}$ and the average neutron leakage of the reference core. In this way, the main goal of the algorithm was to maximize the mean burnup, however, to keep other factors at least as good (or possibly close) to the reference case.

Because the problem is multidimensional, it is not possible to plot the final shape of FF dependent on all parameters, and to check if the FF gradient is similar to each direction.

Therefore, it was decided to investigate the influence of each component separately. The FF was studied by plotting the scaled mean core burnup $B\hat{U}_c$, multiplied by each penalty function p_i against the main goal of the optimization - core burnup, and given penalty parameter in the estimated range of its variability (Figure 4.7a-c). It is an equivalent of plotting FF with an assumption that only one penalty value at a time is above the threshold, so other penalty functions are constant and equal to one. The variability range was assessed from the initial population and limits that were imposed on the parameters. The penalty function parameters γ_i were selected to ensure an equal influence of each of them, i.e. each of the penalty functions could decrease the FF by a maximum of around 30%. On the charts, one can also see the function's breakpoints corresponding to the imposed limit values.

Because it is impossible to generate a chart of FF dependent on all considered parameters, the chart in Figure 4.7c shows FF as a function of parameters to be maximized (mean core burnup) and parameters to be minimized/limited presented in arbitrary units, that is, within the anticipated range of variability (in arbitrary unit $AU \in (0, 1)$). It is a very simplified way of plotting the shape of FF as it assumes that all the parameters increase or decrease simultaneously, but it gives an idea of how the FF may change within the variability of the considered parameters.

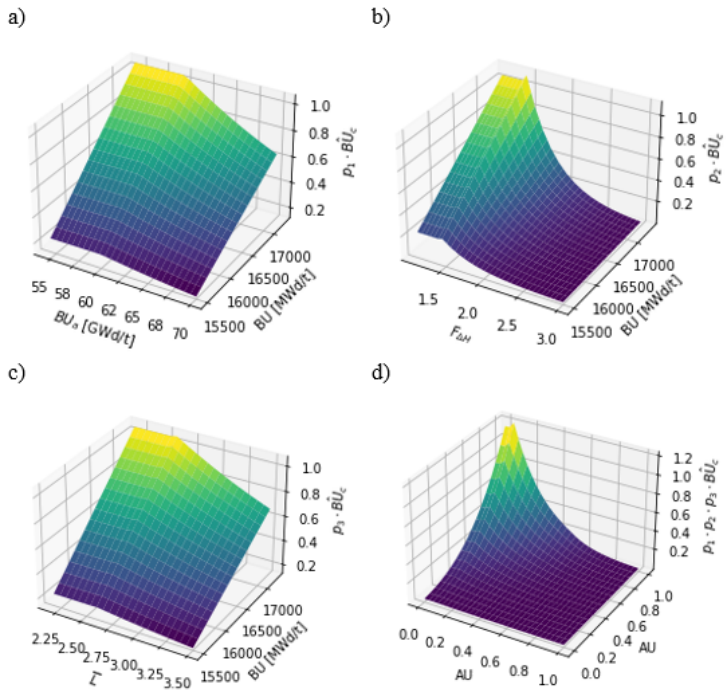


Figure 4.7: Plot of influence of the components on the FF, in range of their expected variability determined based on a random initial population: a) max BU of the fuel assembly, b) max $F_{\Delta H}$, c) average leakage, d) combination of all the components (in arbitrary units).

Selection

For the selection method, the tournament was used. In this method, n specimens are selected and compared and the best of them is passed to be processed by the algorithm, the procedure is repeated N times until the nez population is created. Usually, $n = 3$ but in this case the number of specimens in the population ($N = 200$) is bigger compared to the classical GA (usually $N = 50 - 100$) [35], so it was decided to increase the number of specimens compared at a time. Additional tests showed that GA/PSA performs the best for $n = 20$ and this value was used for all the optimizations.

Crossover

The two-point crossover was used with a partially mapped crossover (PMC, same method as in Case 4 in Section 3.2.3). In this crossover, two randomly selected specimens are exchanging subarrays between two random exchange points. PMC method maintains the constant number of given elements in the chromosome [143], by adjusting the elements outside the exchange region.

Mutation

The mutation was implemented using a *mutation per specimen* approach, meaning that each individual in the population had a probability p_m of undergoing mutation. If the mutation occurred for an individual, one random gene was exchanged with another. In the case of 1/8 symmetry, for which the optimized system consisted of two parts, gene exchange occurred within each respective part.

4.2.3 Shannon Entropy

In order to monitor and possibly control the performance of GA it was decided to define Shannon Entropy (SE) of the population at the given generation g, defined as:

$$H_g = -\sum_{i,j} P_{i,j} \cdot \ln P_{i,j}, \tag{4.15}$$

where:

$$P_{i,j} = \frac{1}{N} \cdot \sum_{k=1}^{N} \delta^{(k)}_{pos_i, gene_j}. \tag{4.16}$$

$P_{i,j}$ is a probability of finding position i in the gene j of the chromosome. N is the size of the population and $\delta_{pos_i,gene_j}$ is the Kronecker delta and $\delta_{pos_i,gene_j} = 1$ when position pos_i is found in $gene_j$ of the chromosome (otherwise it is 0). Similar quantities were used in some previous works (*variance* [156], *genetic variance* [5]), but SE seems to be a good quantity as it usually describes the level of the randomness in a system. SE is the highest when the genes in the population are uniformly distributed (probabilities $P_{i,j}$ are equal) and it reaches zero for a population dominated by one specimen. As in the previous work, this kind of monitoring quantity can be used for ongoing adjustment of the parameters, e.g. mutation rate [156] or FF form [5] (so-called adaptive GA) or it can be used as a stopping criterion, i.e. the GA can stop when SE drops below a certain value [156].

4.2.4 Description of the algorithms

In this part of the study, the optimization algorithm has been prepared in three different versions: Genetic Algorithm (GA), Parallel Simulated Annealing (PSA), and Hybrid (PSA+GA). As it was mentioned, GA are frequently studied in Laoding Optimization roblems (LPO), they were already used for power distribution flattening, cycle extension, boron concentration/excess reactivity/burnup maximization, fissile material usage optimization, fast spectrum reactor design, and other factors/features optimizations [5, 6, 37, 125, 157, 158]. PSA has been used in the LPO problem, for instance, for optimizing hot channel enthalpy rise factor and boron concentration [159] and for multi-cycle cost optimization [160]. GA-PSA hybrid method was studied in the past, usually by combining GA and PSA operators [161] (as in this study) or by applying GA and PSA alternately [73]. A more detailed summary of the optimization methods in the LPO problem can be found in Section 2.1.1 and in [36].

The diagram in Figure 4.8 presents the schematic idea of the operation of all three versions of the algorithm. In yellow, operations originated for GA are marked, in blue ones that come from PSA, and in green, operations that are common in both, GA and PSA. If one follows the shortcut arrows marked in yellow or blue dotted lines in the diagram in Figure 4.8, a scheme only for GA or PSA can be read. All the algorithms start with a random initial population of N solutions (a set of N vectors of random permutations, as in Figure 4.2). Then, at each step:

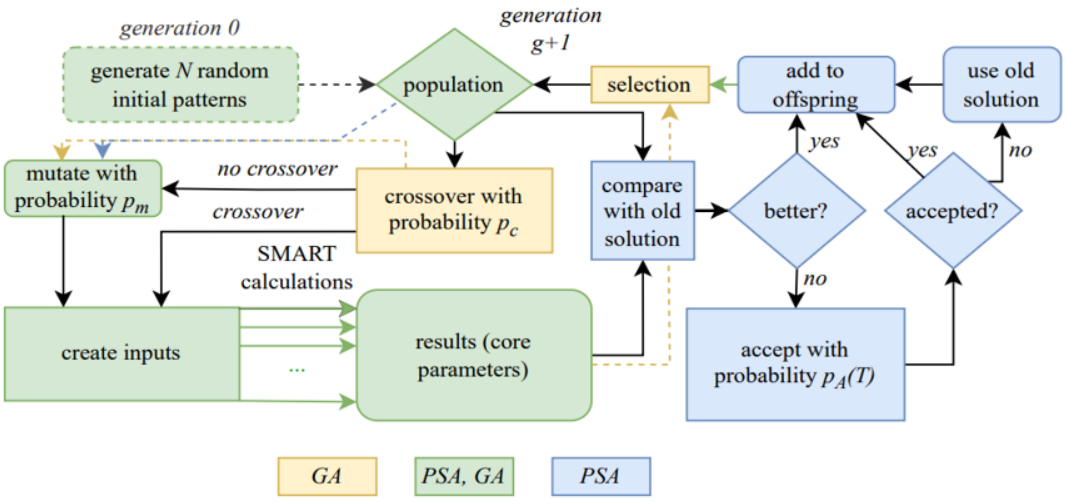


Figure 4.8: Block scheme of the GA, PSA and hybrid PSA+GA algorithms

In the case of GA:

- solutions are randomly paired and sub-sets of the vectors are exchanged between them with probability p_c using PMC (Partially Mapped Crossover) method,
- each solution is mutated with probability p_m (random exchange of genes in the same solution),
- each solution is evaluated and each FF is calculated,
- a new population is selected using the tournament method and calculated FFs.

In the case of PSA:

- each of N solutions is randomly moved in the search space (equivalent to GA's mutation with $p_m = 1$),
- FF is calculated and a better solution is accepted, a worse solution can be accepted with probability p_A,
- N solutions are selected (tournament method),
- the temperature of the system is decreased.

In the case of the hybrid PSA+GA (Table 4.3):

- solutions are moved in the search space but applied GA operators (crossover or mutation),
- each new solution can be accepted or discarded using PSA criterion,
- a new population is selected from the set of new solutions using GA approach (tournament),
- the temperature of the system is decreased.

```
For g in G:
    offspring = []
    FFs = []
    For xi in population:
        # move solution using GA operators (crossover or mutation)
        if random<pc: xi^new = crossover(xi, xr) # xr - random crossover partner
        else  random<pm: xi^new = mutation(xi) # in our case pm = 1.0
        # keep or discard the solution using SA criterion
        if FF(xi^new)>FF(xi): xi = xi^new
        else if random<pA(xi^new, xi, Tg): xi = xi^new
        Tg = decrease(Tg)
        # create offspring as in the GA
        offspring.append(xi)
        FFs.append(FF(xi))
    # use GA selection method
    # select N elements from offspring, based on their FFs values
    population = select(offspring, N, FFs)
```

Table 4.3: Python-like, pseudo-code of the hybrid PSA+GA algorithm.

4.2.5 Coupling with the neutronic code

The goal of this part of the thesis was the implementation of an optimization tool combined with the SMART neutron code, which is part of the *Science v2* package [162]. SMART is a neutronic code based on the 2-energy group diffusion equation (1.10) for the core under static or transient conditions. Paired with the FLICA thermal-hydraulics code [163], it solves neutron transport equations, providing flux or power distributions at the assembly and pin levels. SMART employs the nodal expansion method and finite differences for large mesh calculations, using discontinuity factors and a transverse leakage model to ensure accuracy in nodal current conversion [162]. The code relies on cross-section data from APOLLO2 [164], an assembly lattice code developed by CEA (*Commissariat à l'énergie atomique*). APOLLO2, based on the *Sn* method, solves the 99-group transport equation in 2D assembly to generate 2-group, collapsed cross sections using the collision probability and integral-differential methods [164].

The transition between the EC model in a vector form of Equation 4.11 and SMART code is realized in four steps (see Figure 4.9):

1. **Encoded EC**: the algorithm operates on the vector in the form of Equation 4.11, being a permutation of integers;
2. **Matrix of FA paths**: the vector form is reshaped into a matrix (Equation 4.9), representing a set of FA paths of the EC;
3. **FA configuration**: based on the matrix form, path by path, FA are placed in the 1/8 or 1/4 in the core (depending on the scenario) to create the SMART code input (Figure 4.10);
4. **Full core**: the defined region is mirrored and SMART input is extended to create full core input.

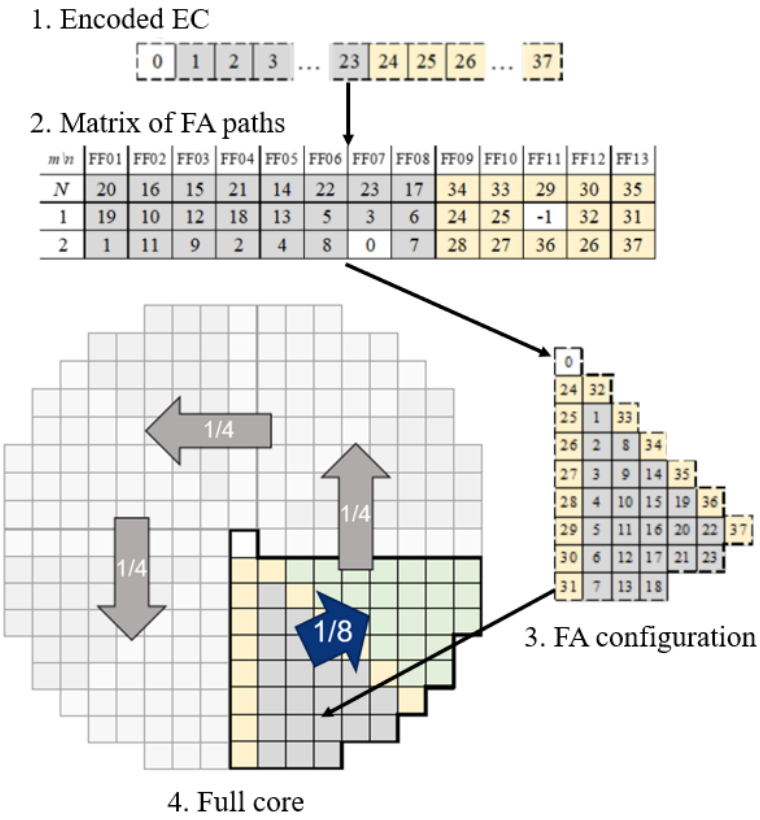


Figure 4.9: Transition of the EC in the vector form (Equation 4.9) to the full core, Implemented in four steps: 1. Encoding EC in vector form, 2. Transposing the vector into matrix form, 3. Determining the FA configuration in 1/8 or 1/4 of the core (depending on the scenario), 4. Symmetrical reflection of symmetry to create a full core.

The definition of the EC in the SMART input is presented in the Figure 4.10. Each line in the input reflects the movement of FA. It consists of the current position of FA (two integers defining the x, y position in the core). Then, there is a name of the cycle, five values of burnup (for quarters and average value), analogically five values of relative power, then there is a number and name that define FA. At the end of the line, there is a previous position of FA. There are 241 lines of this part of the input, reflecting 241 positions and movements in the core that define the EC.

The input is created from the matrix form by iterating, path by path, and assigning to each position the previous position, and, in the case of FFA, its composition (for FFA there is a star "*" instead of burnup and power). In the case of FA that was placed in the core for the subsequent cycle, FA composition did not influence the EC calculation, as after the third cycle all the FA are exchanged.

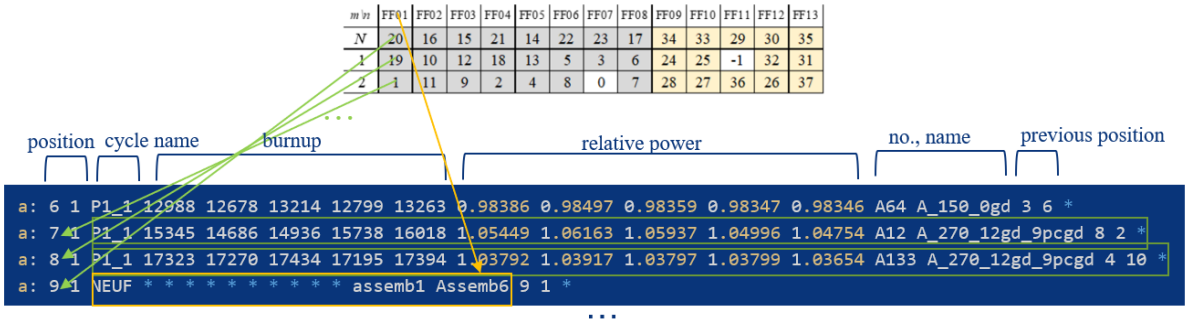


Figure 4.10: Creation of the SMART code input. Each column in the matrix form of EC corresponds to the path of a given FFA. This is translated into subsequent elements of the input of the SMART code. Respectively - new position of the assembly, its parameters (burnup and relative power, number and name of the assembly), and previous position.

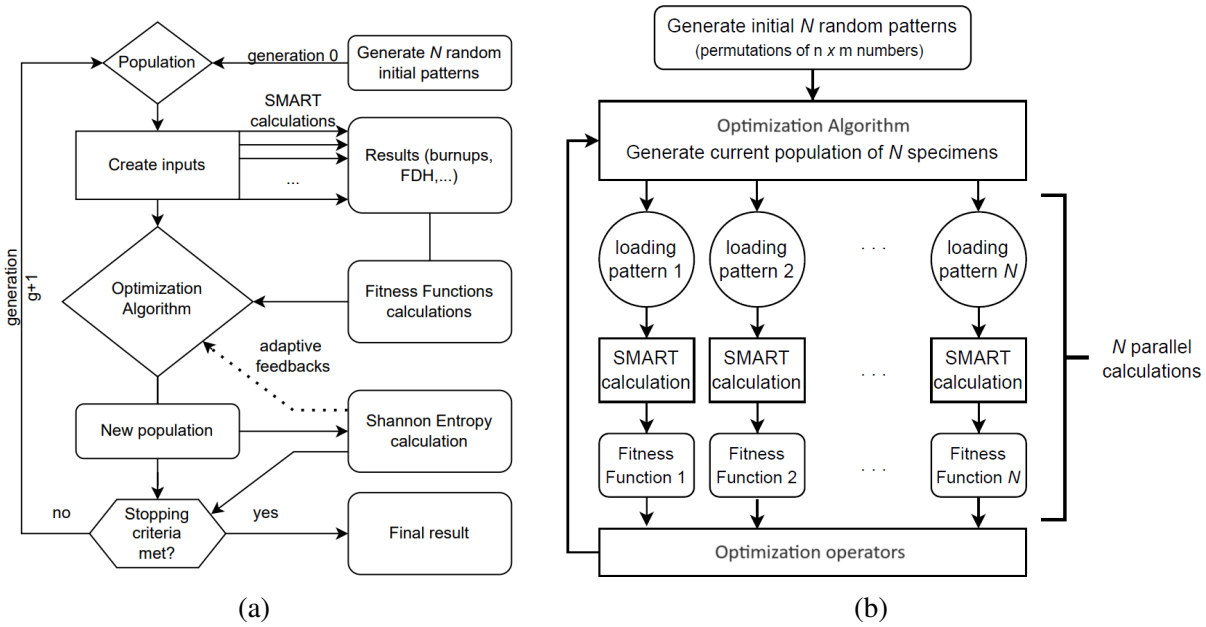


Figure 4.11: Flowchart of the algorithm (a) and SMART calculation parallelism (b).

After defining the connection between the EC definition in the SMART code, the algorithm can start its operation. First, the initial population of N specimens is randomly generated (*generation 0*). Then, based on the random patterns N, inputs for the SMART code are created. The calculations are done in parallel (Figure 4.11b). After the calculations, the results are read by the algorithm to assess each pattern (specimen) and calculate the corresponding FF (Equation 4.13). Then, the main part of the algorithm creates a new population, depending on the method, using selection, crossover, and/or mutation operators. After that, SE is calculated (Equation 4.15) and the algorithm controls whether the stopping criteria are met or not. If so, it stops its operation and selects the best specimen that was created during the optimization. If not, it creates new inputs for SMART, runs the calculations, reads results, assesses specimens,

creates a new population, and so on. At this point, SE can be also used to apply adaptive feedbacks (dotted line in Figure 4.11a) and adjust selected parameters of the optimization algorithm, like mutation rate [156] or FF from [6]. This was, however, not considered in the current study. The transition between the Framatome's diffusion calculation code SMART and the algorithm is presented in the flowchart in Figure 4.11a.

4.2.6 Optimization tool

As the used code is embedded into the Science v2 package with *Copilote* calculations management tool [162], its flexibility allows a simple connection between SMART code, optimization algorithm, and the user. Therefore, one of the goals of the study was to plan and prepare the software for the production of an industrial tool that would serve any user as a support in the process of the nuclear fuel cycle design. Figure 4.12 shows the developed *Copilote* calculation plan and its GUI that allows the user to specify the path of the EC calculation to be optimized. In the *Copilote* EC plan the user defines the core to be optimized, its geometry, properties, fuel assemblies, compositions, and other settings of the EC plan, like burnup steps and calculation type. In the LPO GUI, the user provides the optimization parameters such as the optimization method, size of the population, number of generations, FF form, initial temperature, mutation rate, and others.

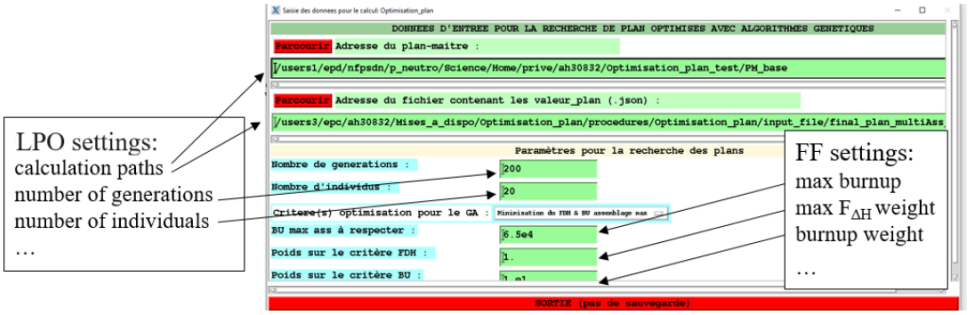


Figure 4.12: Example of the GUI for the cycle optimization, allowing for the initiation of the optimization process - specifying the path to the model to be optimized, defining the parameters of the algorithm's operation, and the objectives of optimization.

Figure 4.13 presents an example of the realization of the *Copilote* LPO plan. The optimization starts by duplicating the EC plan N times and reinitializing them with random loading patterns to use it as the population for the optimization algorithm ("*current generation calculations*" in Figure 4.13). The coupling between the optimizer and SMART is realized via the *Copilote* tool that allows parallel calculations, thus all the copied plans are run at the same time (as presented in Figure 4.11b). After each generation, the algorithm saves the best solution as

the user may be interested in other, potentially useful loading patterns ("*best configuration at given generation*" in Figure 4.13).

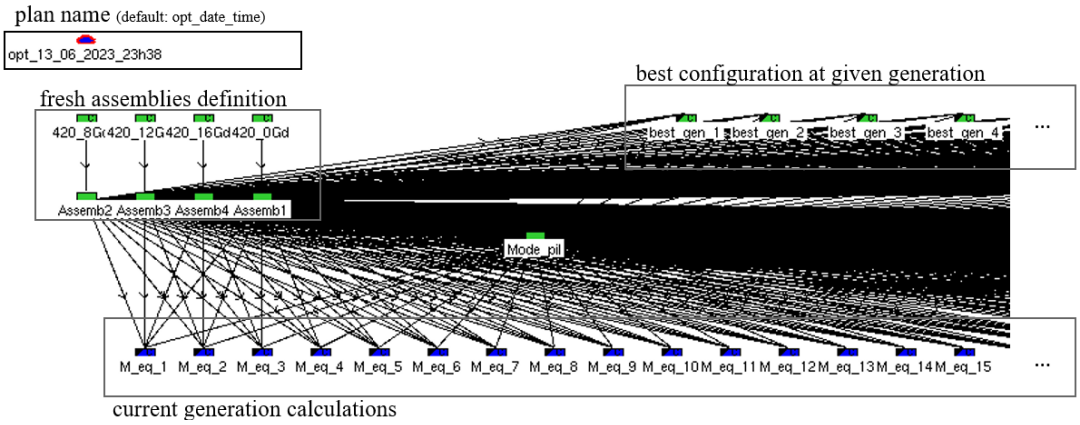


Figure 4.13: The Science v2 calculations for the LPO (Copilote GUI).

Input file

The optimization algorithm was prepared using `deap` library [155] and `Python 3.8` language in an object programming manner. The project was prepared to be easily compatible with any *Copilote* project, be flexible and easy to use. To use the optimizer one needs to import `LoadingPatternOptimizer` class. The class contains several methods that need to be called to set the parameters and run the optimization (see Table A.15 in Appendix A that presents an example of the `main.py` file, where the class is created and the algorithm is initiated).

4.2.7 Considered Scenarios

In this part of the study, it was decided to run the optimization algorithm for three cases: a test case with 1/8 symmetry and a test form of FF and two scenarios with improved FF and the same optimization goal, for two different symmetries - 1/8 and 1/4.

Test case

The test case describes the first attempt at optimizing the EC problem using the first version of the objective function FF_{test} (Equation 4.12), based on which the extended, main objective function FF (Equation 4.13) was prepared. For computational reasons, the test case was conducted only for GA and 1/8 symmetry. However, it allowed for drawing certain conclusions, improving FF, and applying them to other algorithms and 1/4 symmetry.

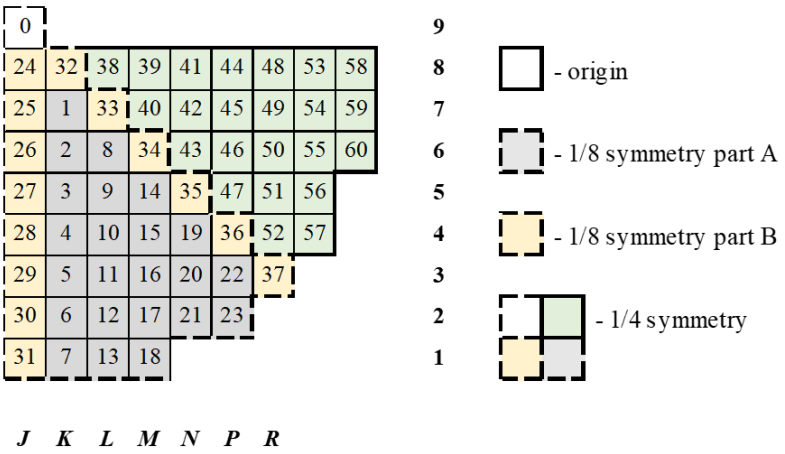


Figure 4.14: Optimized core with 1/8 (gray and yellow parts) and 1/4 symmetry (gray, yellow and green) symmetries.

Case 1: 1/8 symmetry

For the scenario with 1/8 symmetry the optimized system consists of three parts, denoted as white, yellow, and green regions in Figure 4.14. Due to the way the whole core is constructed, the gray part is mirrored eight times, the yellow part is mirrored four times and the white part (origin) is not mirrored. It imposes restrictions on the creation of the core and does not allow the assemblies to be moved between these two parts. If we move one FA from the yellow to the gray part, we will miss four FA for four copies of the gray part while creating the full core. And vice-versa, if we move FA from the gray to the yellow part we will have four redundant FA, not used in the full core. That is why, for the 1/8 symmetry, the optimization problem was divided into two independent parts - *A* (23 elements) and *B* (14 elements) in Figure 4.15. The white part (the origin), as it is not mirrored. Only one FA is always moved to the origin (1 element). It can come from both, the yellow or gray part.

Case 2: 1/4 symmetry

In the case of 1/4 symmetry FA can migrate within the whole region without any restrictions as it all will be mirrored four times (except for the origin). The optimized system consists of one part (white, yellow, gray and green parts together in Figure 4.14), 61 elements in total (parts *A*, *B* and *C* in Figure 4.15).

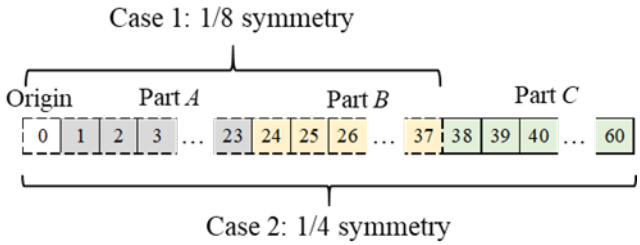
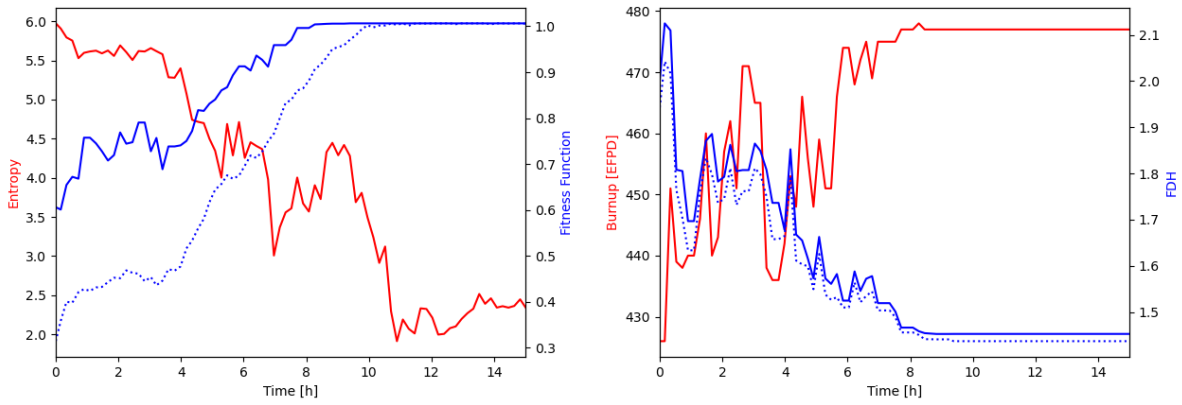


Figure 4.15: EC in the vector form for Case 1 (1/8 symmetry) and Case 2 (1/4 symmetry).



(a) (b)

Figure 4.16: Test FF: Optimization for $N = 200$ specimens, 80 generations, $p_m = 0.01$, $p_c = 0.90$, FF in the form of Equation 4.13. (a) Best FF in the current population (blue solid line), average FF (blue dotted line) and Entropy (Equation 4.15)(solid red line), (b) average core burnup [EFPD] (red solid line), $\max(F_{\Delta H})$ (solid blue line) and initial $F_{\Delta H}$ (dotted blue line) of the best specimen after given time.

4.3 Results of the Test Case

4.3.1 Test Fitness Function

Figure 4.16a and 4.16b show the convergence of the GA as a function of the optimization time. Each generation took around 12 minutes, charts present 80 generations ($\sim 15h$). It can be seen that at the beginning of the optimization, the worst specimens are slowly being removed from the population (SE decreases). After around 6 hours (30 generations), a good scheme is found. By crossover and mutation, GA is searching different combinations of this scheme, which results in the increase of diversity in the population (SE increases). Around 9^{th} hour (45 generations), the best specimen is found and it starts to dominate the population so SE decreases rapidly until a certain level, determined by the probability of mutation (because of it SE will not drop to zero completely).

The algorithm starts from average value of $\overline{FF}_0 \approx 0.31$, it increases in 51 generations (around 10h or $51 \times 200 \approx 10^4$ simulations) around 3 times to $FF_{51} \approx 1.01$ and achieves a

solution that is characterized by $\overline{BU}_c \approx 477$ EFPD, $\max(BU_a) \approx 56.5$ GWd/t and $\max(F_{\Delta H}) \approx 1.45$. Therefore, compared to the reference case, GA proposes an EC with around 2% lower $\max(F_{\Delta H})$ (1.48 vs 1.45), however by the cost of the almost 3% increase of $\max(BU_a)$ (55.0 vs 56.0 GWd/t, but still kept below required 60 GWd/t) and a bit more than 1% decrease in the mean burnup of the core $\overline{BU}_c$ (477 vs 483 EFPD). Overall, one can say, that the features of the proposed core are much improved compared to the best configuration taken from the initial 200 randomly generated patterns. The final solution is also comparable to the reference EC that was designed by the engineers. It can be even said that it is slightly better, in terms of the FF defined by Equation 4.12, as FF of the solution is almost 1% higher than the FF of the reference case (see Table 4.4). However, while analyzing the generated loading pattern (Figure 4.18) one can notice that the GA tends to place FFA at the boundary of the core, which results in the flattened power distribution but also increased leakage during the whole cycle (Figure 4.17). Therefore $1/F_{\Delta H}$ relation seems to be too strong and affects the FF too much.

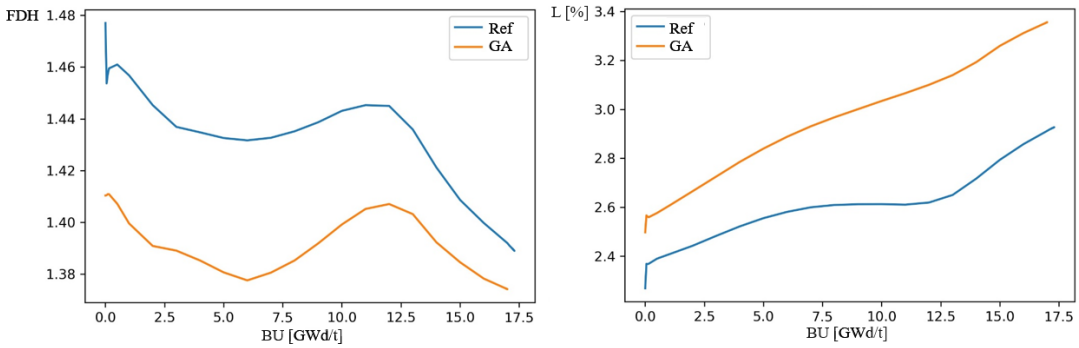


Figure 4.17: Test FF: Evolution of the $F_{\Delta H}$ (left) and $\overline{L}$ (right) for the GA solution and the reference.

It has to be remembered that the form of FF determines the solution, so the solution is as good as is the form of the FF. Thus, a higher FF does not mean a better EC, but it certainly shows a proper operation of the GA. Despite the significant reduction in $max(F_{\Delta H})$ while simultaneously extending the cycle, it is essential to be aware that many physical processes occur in the nuclear core, affecting each other. As can be expected, flattening the power distribution (Figure 4.17) during the cycle simultaneously increases the leakage of neutrons due to the relative increase in flux at the core's periphery. To optimize the $F_{\Delta H}$ value, it is necessary to consider neutron leakage, as these two values will compete with each other, and improving one will usually be associated with worsening the other.

Parameter p	$GA(g=0)$	$GA(g=51)$	Reference	$\Delta_{GA(g=51)/Ref}$ [%]
$\overline{BU}_c$ [EFPD]	426	477	483	-1.2
$\max(BU_a)$ [MWd/t]	67.4	56.5	55.0	+2.7
$\max(F_{\Delta H})$	2.01	1.45	1.48	-2.0
FF (Equation 4.13)	0.61	1.01	1.00	+0.8

Table 4.4: Test FF: Comparison of the reference core and GA solutions after generation g: $GA(g=0)$ - best randomly generated (best solution from the initial population), $GA(g=51)$ - best solution after 51 generations, and $\Delta_{GA(g=51)/Ref}$ - ratio of the FFs (in %).

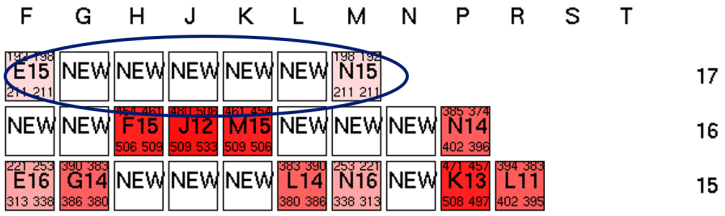


Figure 4.18: Test FF: Top part of the loading pattern for the GA solution.

4.3.2 Main Fitness Function

After conducting test optimization using the initial form of the FF (Equation 4.12), GA was run using the modified FF (Equation 4.13). Figure 4.20a and Figure 4.20b show the convergence of the GA as a function of the optimization time. The charts present 150 generations ($\sim 30h$). It can be seen that after around 20 generations, a good scheme is found. Around 80 generations ($\sim 16h$), the best specimen is found and it starts to dominate the population so SE decreases again to a certain level, determined by the probability of mutation.

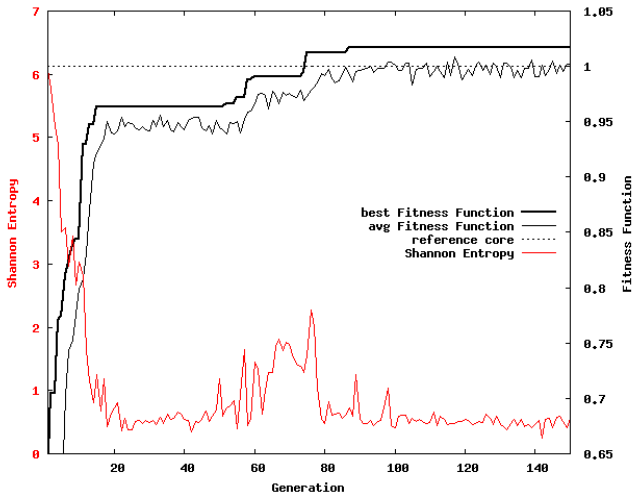
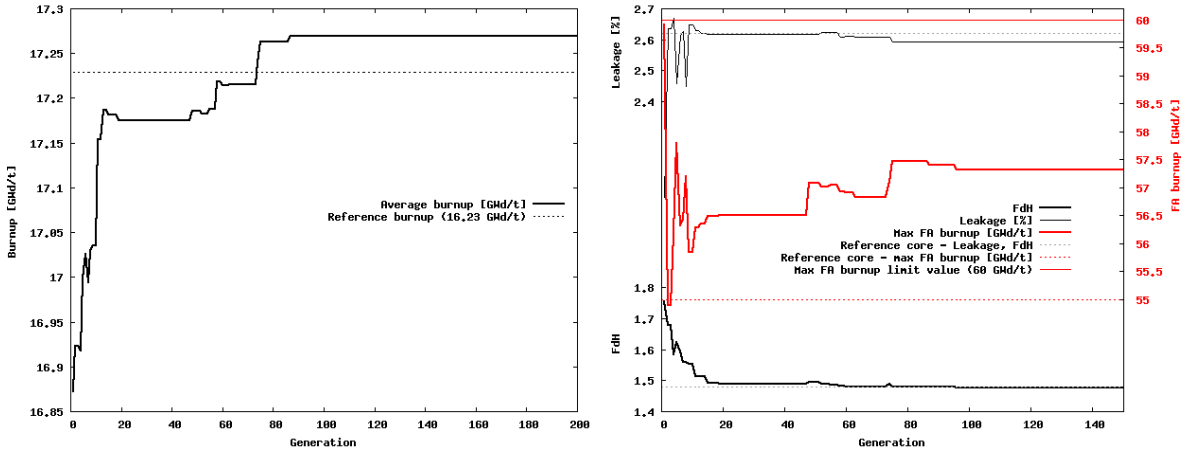


Figure 4.19: Main FF: Evolution of the best (black solid line) and average value (black thin line) of the FF at each generation and evolution of SE (red line) compared to the reference (dotted line). $N = 200$ specimens, 150 generations, $p_m = 0.10$, $p_c = 0.90$ and FF in the form of Equation 4.13.



(a) (b)

Figure 4.20: Main FF: Evolution of the optimized parameters: (a) main goal of the optimization - average burnup of the core $\overline{BU}_c$ and (b) evolution of the penalty functions factors - $\max(F_{\Delta H})$, average leakage $\overline{L}$, max FA burnup $\max(BU_a)$ and their threshold values. Optimization for $N = 200$ specimens, 150 generations, $p_m = 0.01$, $p_c = 0.90$, FF in the form of Equation 4.13.

The algorithm starts from average value of $\overline{FF}_0 \approx 0.26$, it increases in 96 generations ($\sim 19\ h$ or total number of $96 \times 200 \approx 2 \cdot 10^4$ simulations) around 4 times to $FF_{96} \approx 1.02$ and achieves a solution that is characterized by $\overline{BU}_c \approx 17.27 MWd/t$ (or 484.1 EFPD), $\max(BU_a) \approx$ 57.3 GWd/t, $\max(F_{\Delta H}) \approx 1.478$ and $\overline{L} \approx 2.59$ %. Table 4.5 presents a comparison between the best randomly generated core (from the initial population), the proposed solution and the reference case.

Compared to the reference case, GA proposes an EC characterized by an extended life-time (+1.1 EFPD), decreased average leakage (-0.03pp.), and similar value of $\max(F_{\Delta H})$ (1.477 vs 1.478), but on the other hand by the cost of around 4% increase of $\max(BU_a)$ (55.0 vs 57.3 GWd/t, but kept below required value). Similar to the previous case, one can say, that the features of the proposed core are much improved compared to the best random configuration ($max(FF_0) = 0.64$, see Table 4.5). The final solution is also comparable to the reference and, again, it can be considered to be slightly better, in terms of the defined FF (2% higher FF). This time, $max(F_{\Delta H})$ does not dominate the optimization, GA increases the length of the cycle and keeps the other parameters close to the defined values, including leakage (Table 4.5).

Therefore, after the tests, FF was used in the form of Equation 4.13 for the remaining part of the study, i.e. for GA, PSA, and Hybrid algorithms and 1/8 and 1/4 symmetries. The test GA solution is not analyzed in detail as full optimization is done and described in Section 4.4 and Section 4.5.

Parameter p	$GA(g=0)$	$GA(g=96)$	Reference	$\Delta_{GA(g=96)-Ref}$
$\overline{BU}_c$ [EFPD]	473.4	484.1	483.0	+1.1
$\overline{BU}_c$ [GWd/t]	16.9	17.27	17.23	+0.04
$\overline{L}$ [%]	2.09	2.59	2.62	-0.03
$\max(BU_a)$ [GWd/t]	59.9	57.3	55.0	+2.3
$\max(F_{\Delta H})$	1.755	1.478	1.477	+0.001
FF (Equation 4.13)	0.64	1.02	1.00	+0.02

Table 4.5: Main FF: Comparison of the reference core and GA solutions: $GA(g=0)$ - best randomly generated (best solution from the initial population), $GA(g=96)$ - best solution after 96 generations, and $\Delta_{GA(g=96)-Ref}$ - differences between the values of the optimized and reference cores.

4.4 Results of the Case 1: 1/8 symmetry

The following section presents an optimization conducted for the before-described core, based on the EPR reactor (Section 4.1.2). Simulations were carried out for three algorithms: GA, PSA, and PSA+GA hybrid, and 1/8 symmetry, for $N = 200$ specimens assessed at the same time (parallel SMART calculations through the *Copilote* GUI [162]), $p_m = 0.10$ (mutation per specimen), $p_c^{GA} = 0.9$, $p_c^{PSA+GA} = 0.2$ (PMC method) and $n = 20$ (tournament size). All the algorithms were run for $G = 100$ generations. The section details the course of the simulations and the analysis of the obtained solutions.

4.4.1 Optimization

Figure 4.21 presents the convergence of the algorithms, i.e. the maximal value of the FF, average FF, and the standard deviation over the population at each generation step. All the algorithms start from a random FF value of around 0.3-0.7 and in around 50-60 generations (around 14 hours of calculations) they achieve a solution characterized by FF value comparable to the reference case ($\sim$1.0, see Figure 4.21).

Table 4.6 presents summarized properties of the optimized solutions proposed by PSA, GA, and PSA+GA compared with the reference case and best random configuration, selected from $3 \cdot 200 = 600$ randomly generated starting points. One can see that PSA+GA solution is characterized by the highest FF value (1.031), the second best is the PSA solution (1.012), and then the GA one (1.005). The solutions offer extended cycles by around 3.1, 3.4, and 2.0 EFPD respectively. In the case of GA, it is achieved by the cost of increased leakage (around 0.2 pp.). All the solutions are characterized by increased maximal FA burnup (additional 0.3-2.9 GWd/t),

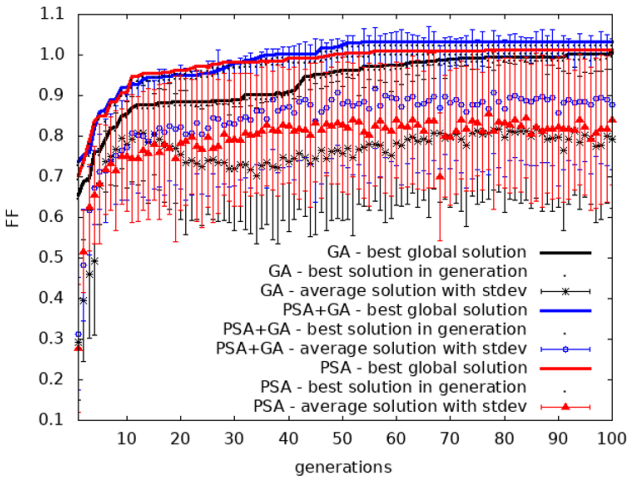


Figure 4.21: Case 1: Convergence of the optimization algorithms.

however, for all of them, it is kept below the required 60 GWd/t.

Table 4.6: Case 1: Summary of the optimization results, for PSA, GA, and PSA+GA algorithms, compared with reference case and random solution, 1/8 symmetry.

Parameter	**reference**	**random***	**PSA**	**GA**	**PSA+GA**
FF (Equation 4.13)	1.000	0.737	1.012	1.005	1.031
$\overline{BU}_c$ **[GWd/t]**	17.23	16.88	17.35	17.30	17.33
$\overline{BU}_c$ **[EFPD]**	483.0	473.2	486.4	485.0	486.1
$\mathbf{max}(F_{\Delta H})$	1.477	1.765	1.517	1.480	1.491
$\bar{L}$**[%]**	2.62	2.65	2.37	2.84	2.50
$\mathbf{max}(BU_a)$**[GWd/t]**	55.0	56.4	57.9	55.3	57.5

*best solution out of the randomly generated initial $3 \cdot 200 = 600$ configurations

Figure 4.22 presents the evolution of the population over time during optimization. The points on the chart represent a single individual of the population for the entire optimization, respectively $F_{\Delta H}$ (Figure 4.22a-c), leakage (d-f) and $max(BU_a)$ (g-i) as a function of $\overline{BU}_c$. Figure 4.22j-l present $F_{\Delta H}$ as a function of leakage (two strongly inter-dependant parameters), together with assumed threshold values (dotted horizontal line), for PSA, GA and PSA+GA algorithms respectively. The colors in the figures represent the number of individuals in the population, with blue color indicating single individuals, while red indicates that the majority of the population is concentrated at that point. In the figure, one can see that from an evenly distributed initial population, the population moves towards optimal values, i.e. higher burnup, lower leakage, lower $F_{\Delta H}$, and maximal FA burnup below the defined value.

Looking at the charts in (Figure 4.22), one can see differences in the distributions be-

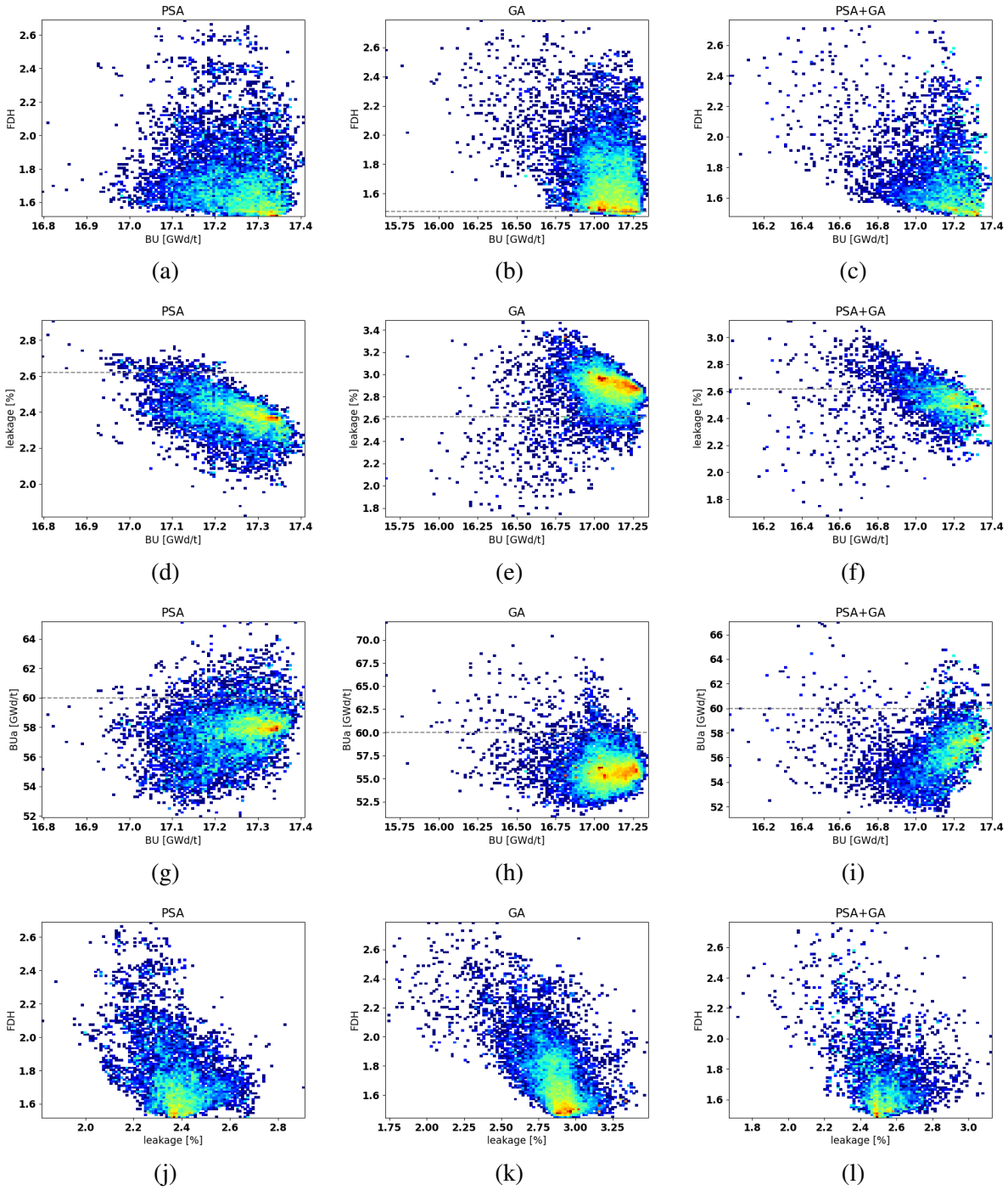


Figure 4.22: Case 1: Population evolution in the space of penalty parameters and core mean burnup for PSA, GA, and PSA+GA, 1/8 symmetry, N=200 specimens for G=100 generations. Dotted horizontal lines present defined limits imposed on the parameters.

tween algorithms. For PSA, most of the population is concentrated around one point, with some distribution around it. Thus, it can be inferred that the algorithm has practically converged to a solution, which is evident in Figure 4.21 by the constant value of FF for the best global solution, although the relatively high standard deviation suggests that the population has not yet converged to a single solution (despite the temperature in the PSA dropping to zero), probably

due to the fairly large number n in the tournament method, the selection is not as strict, which allows for maintaining diversity in the population.

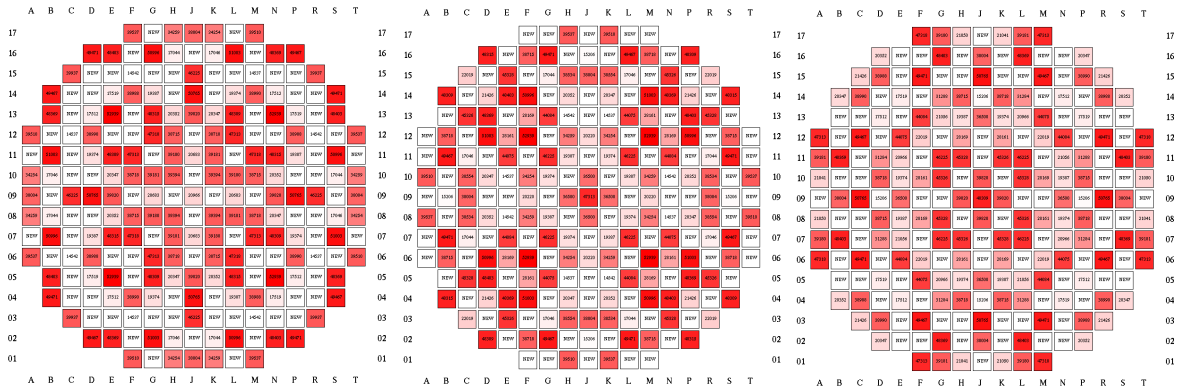
For the GA solution, the area of light yellow, green, and red color is significantly larger, indicating a lack of convergence of the population to a single solution and a broader search for solutions, also related to less restrictive selection and additionally to mutation, and in this case too few generations (FF is still increasing, and the standard deviation is large, as can be seen in Figure 4.21).

For the PSA+GA, the solution seems to converge most to a single point, with the population concentrated around it and with little dispersion, which is confirmed by FF plot in Figure 4.21. It should be remembered that apart from soft selection, the hybrid method also involves mutation (which does not decrease with generations) and maintains diversity at a certain level.

From the charts in Figure 4.22, it can be inferred that $F_{\Delta H}$ is the most problematic parameter among those considered. Among random configurations (blue points sparsely distributed across the chart), it is easy to find a solution with low leakage and a low value of $max(BU_a)$. However, none of the random configurations meet the requirement for the maximum value of $F_{\Delta H}$. Only after a certain number of generations the algorithms approach the imposed value, essentially only GA manages to reach a solution with the required uniformity of power distribution. It is apparent that this is achieved at the expense of leakage (and lower FF value), as the GA solution approaches the set boundary "from above", i.e., from higher leakage values, whereas for PSA and PSA+GA, the algorithms reach the boundary value "from below", i.e., from lower values. Different strategies and pursuits of different solutions are visible here.

4.4.2 Discussion on the Results

Figure 4.24 presents the evolution of $F_{\Delta H}$, F_Q, Leakage, and BC during the cycle (see definitions in Section 1.2.4), for the reference case, and Figure 4.23 presents the loading patterns proposed by the algorithms.



(a) PSA (b) GA (c) PSA+GA

Figure 4.23: Case 1: EC proposed by the algorithms, 1/8 symmetry.

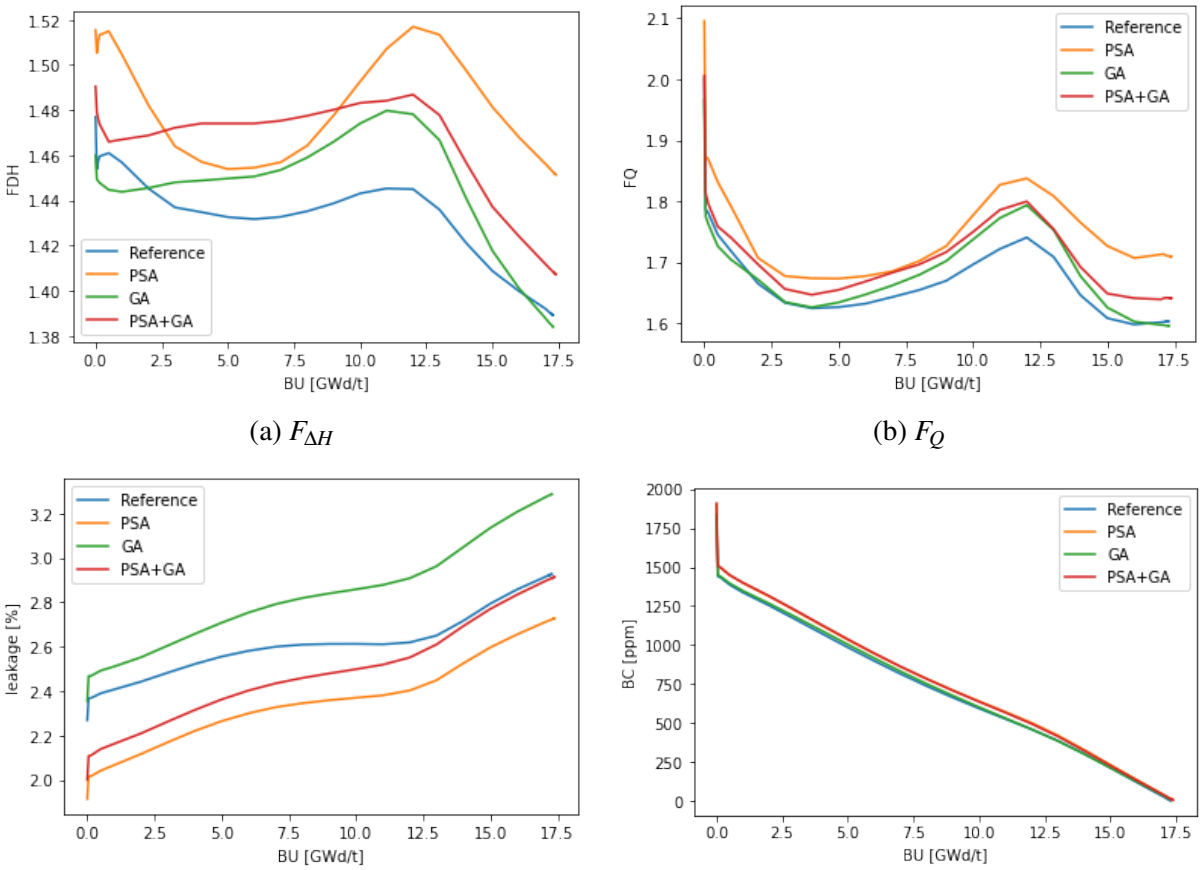
Since values such as $max(F_{\Delta H})$, average leakage, or cycle length are values describing the objective function and allow for the algorithm's operation, they do not, however, reflect the course of the cycle and are not sufficient to accurately assess the obtained solution. For this purpose, a more detailed cycle analysis was conducted for the obtained solutions. Figure 4.24 presents the evolution of $F_{\Delta H}$, F_Q, Leakage, and BC during the whole cycle, for the reference case and the solutions proposed by the algorithms, for the 1/8 symmetry case. Below we briefly analyse their characteristics.

PSA solution

From Figure 4.24 one can see that PSA proposed a low-leakage solution, with leakage lowered by around 0.4-0.2 pp throughout the cycle, compared to the reference. At the same time, $F_{\Delta H}$ and F_Q are above the reference core values, but still kept within reasonable boundaries, below 1.5 and 1.9, respectively, for most of the cycle. The solution requires more BC at BLX, 1496 vs 1432 ppm. In this configuration (Figure 4.23a), an outer ring of highly burnt fuel assemblies is visible, maintaining low neutron leakage. Then, there is a ring of mainly new assemblies. Inside the core, there is a cross structure composed of both fresh and burnt fuel assemblies. However, the burnt assemblies dominate, reducing the unevenness of the power distribution.

GA solution

GA proposes a solution with $F_{\Delta H}$ lowered at BOC, then it is higher than the reference but smaller than required $max(F_{\Delta H})$ (Figure 4.24). The leakage is, however, increased throughout the whole cycle, by around 0.05 pp. at BOC, up to almost 0.4 pp. at EOC, which increases the average leakage and decreases FF. In the configuration (Figure 4.23b) there are several FFA arranged right at the edge, which explains the high leakage. It also appears to consist of rings of burnt and fresh fuel assemblies, and the core itself is intersected by diagonal structures. However, the



(a) $F_{\Delta H}$

(b) F_Q

(c) Leakage

(d) Boron concentration

Figure 4.24: Case 1: Evolution of the FF parameters, for 1/8 symmetry, for PSA, GA, PSA+GA and ref. case.

specific arrangement strategy is less distinct.

PSA+GA solution

The hybrid algorithm proposes a solution between GA and PSA. The $F_{\Delta H}$ is higher than in the case of GA, it's higher than the reference case, but also almost maintained below the designated limit (Figure 4.24). On the other hand, the solution is characterized by lower neutron leakage, compared to the GA solution, but is higher than the PSA one. Compared to the reference case, the leakage is smaller, especially at BOC, and then it converges to similar values. The solution also requires increased BC at BLX, at a level similar to the PSA one (1498 vs 1496 ppm). In the configuration (Figure 4.23c), at the edge of the core mainly fuel assemblies with medium and high burnup levels are arranged, ensuring low leakage. Inside, new and partially burnt fuel assemblies are alternately placed, resulting in a relatively flat power distribution.

4.5 Results of the Case 2: 1/4 symmetry

The following section presents an optimization conducted for the EPR-like core with 1/4 symmetry, for $N = 200$ specimens, $p_m = 0.10$, $p_c^{GA} = 0.9$, $p_c^{PSA+GA} = 0.2$ and $n = 20$, for $G = 100$ generations. The section details the course of the simulations and the analysis of the obtained solutions.

4.5.1 Optimization

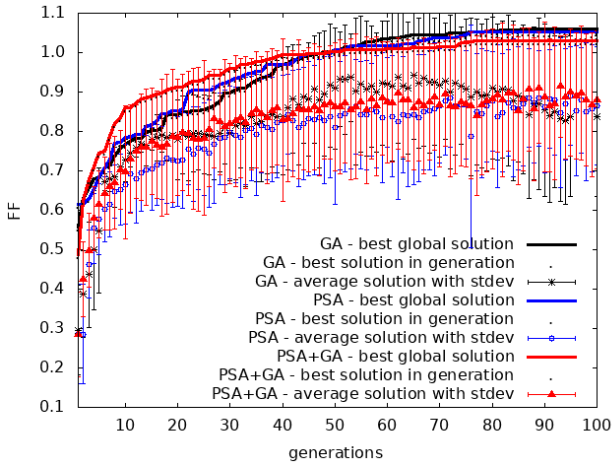


Figure 4.25: Case 2: Convergence of the optimization algorithms.

Figure 4.25 presents the convergence of the algorithms, i.e. the maximal value of the FF, average FF, and the standard deviation over the population at each generation step. All the algorithms start from a random FF value of around 0.4-0.5 and in around 70-80 generations (around 16 hours of calculations) they achieve a solution characterized by FF value comparable to the reference case ($\sim$1.0, see Figure 4.25). Table 4.7 presents the summary of the properties of the optimized solutions proposed by PSA, GA, PSA+GA compared with the reference case and best random configuration, selected from $3 \cdot 200 = 600$ randomly generated starting points. One can see that GA solution is characterized by the highest FF value (1.061), the second best is the PSA solution (1.053), and then the PSA+GA one (1.035). The solutions offer extended cycles by around 3.7, 3.4, and 2.8 EFPD respectively. In the case of PSA and PSA+GA, it is achieved by the cost of increased leakage (around 0.1 pp.). All the solutions are characterized by increased maximal FA burnup (additional 1.2-1.5 GWd/t), however, for all of them, it is kept below the required 60 GWd/t.

Figure 4.26 presents the evolution of the population for each of the algorithms within the space of the penalty parameters (Equation 4.13) and mean core burnup, together with assumed

Table 4.7: Case 2: Summary of the optimization results, for PSA, GA, and PSA+GA algorithms, compared with reference case and random solution.

Parameter	**reference**	**random***	**PSA**	**GA**	**PSA+GA**
FF (Equation 4.13)	1.000	0.614	1.053	1.061	1.035
$\overline{BU}_c$ **[GWd/t]**	17.23	16.83	17.35	17.36	17.32
$\overline{BU}_c$ **[EFPD]**	483.0	471.8	486.4	486.7	485.8
$\mathbf{max}(F_{\Delta H})$	1.477	1.765	1.477	1.475	1.478
$\bar{L}$**[%]**	2.62	2.65	2.64	2.54	2.65
$\mathbf{max}(BU_a)$**[GWd/t]**	55.0	56.4	56.5	56.2	56.5

*best solution out of the randomly generated initial $3 \cdot 200 = 600$ configurations

threshold values (dotted horizontal line). As in the case of 1/8 symmetry, the population moves towards optimal values, starting from evenly distributed specimens. The differences are visible in the spread of each distribution, for each algorithm. PSA seems to be mostly spread, probably not achieving a full convergence after assumed 100 generations. For GA and PSA+GA there are more visible convergence paths (narrow red regions in Figure 4.26), with some dispersion around the solutions, related to the mutation and the softness of the selection method. On the other hand, for the realizations and defined settings (acceptance rate, mutation, etc.) PSA seems to be the least selective algorithm, so it has the lowest tendency to discard worse solutions, GA seems to be more selective (despite mutation) and it searches for solutions in the proximity of the current best solution, whereas PSA+GA seems to be in between these two strategies.

4.5.2 Discussion on the results

Figure 4.28 presents the evolution of $F_{\Delta H}$, F_Q, Leakage, and BC (definitions in Section 1.2.4) during the cycle and Figure 4.27 presents the loading patterns proposed by the algorithms. Below we briefly analyse their characteristics.

PSA solution

From the Figure 4.28 one can see that PSA proposed a solution with lowered $F_{\Delta H}$ and F_Q values and flattened power distribution. Both $F_{\Delta H}$ and F_Q are below the reference core values within the whole cycle. The solution is characterized by lower than reference leakage at the beginning of the cycle, but it increases above the reference, in the second half of the cycle. It also requires more BC at the beginning of the cycle with xenon equilibrium (BLX), 1466 vs 1432 ppm. In this configuration, one can see the rings of spent fuel and fresh fuel stacked alternately, as seen

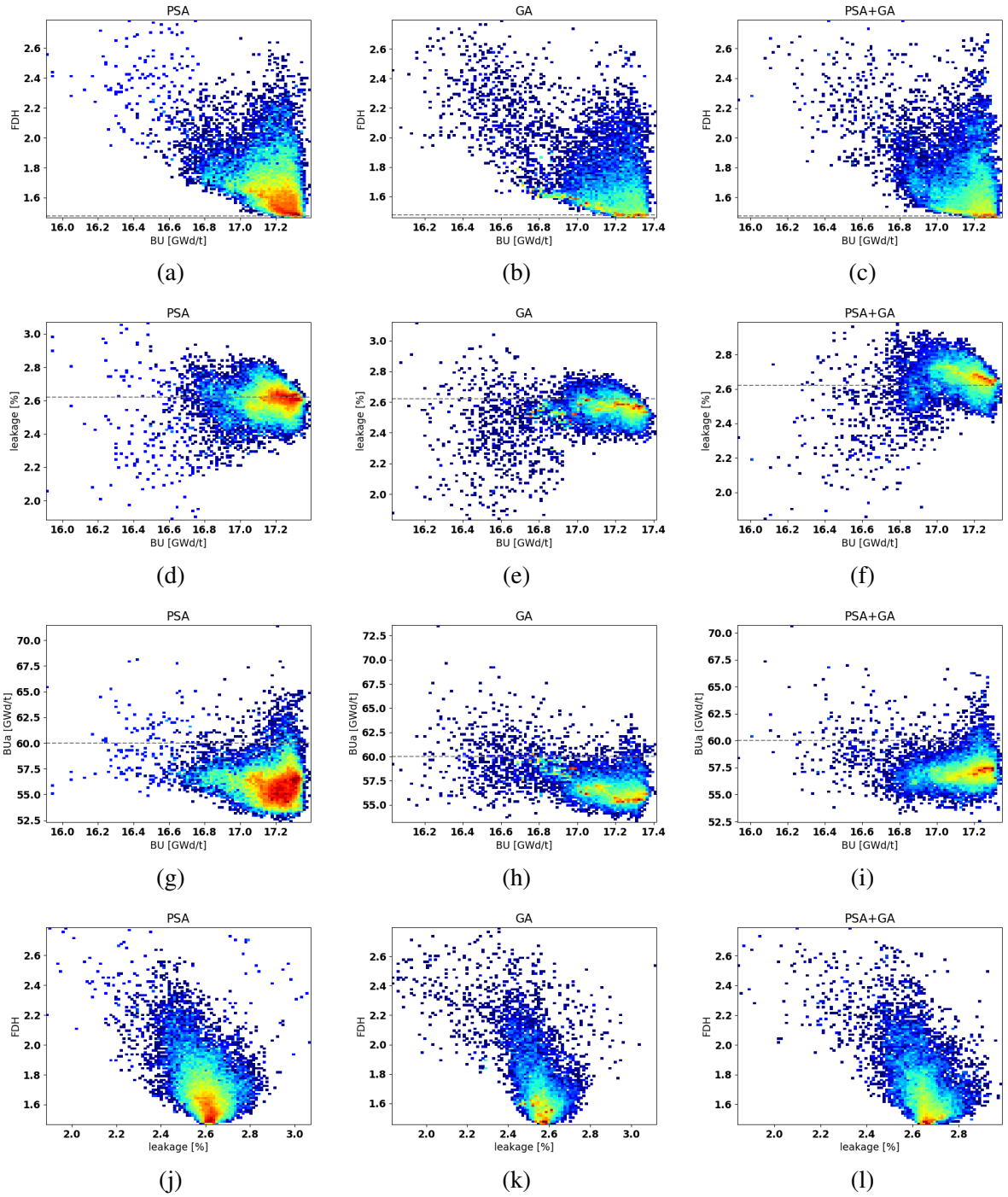
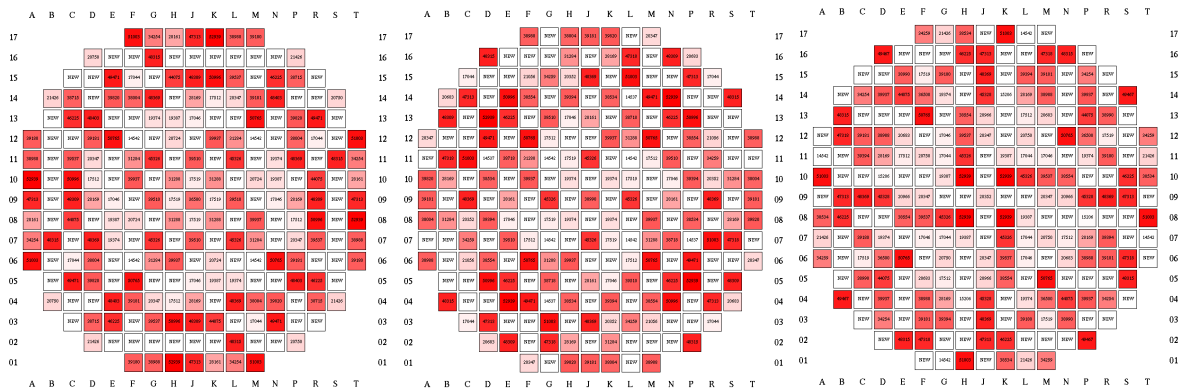


Figure 4.26: Case 2: Population evolution in the space of penalty parameters and core mean burnup for PSA, GA, and PSA+GA, N=200 specimens for G=100 generations. Dotted horizontal lines present defined limits imposed on the parameters.

from the outer edge of the core. The centre is dominated by FA with a medium burn-up level (see Figure 4.27a).

GA solution

GA proposes a low-leakage loading scheme, with leakage lowered by around 0.2pp throughout the whole cycle. Although the max($F_{\Delta H}$) is below the reference case (resulting in relatively



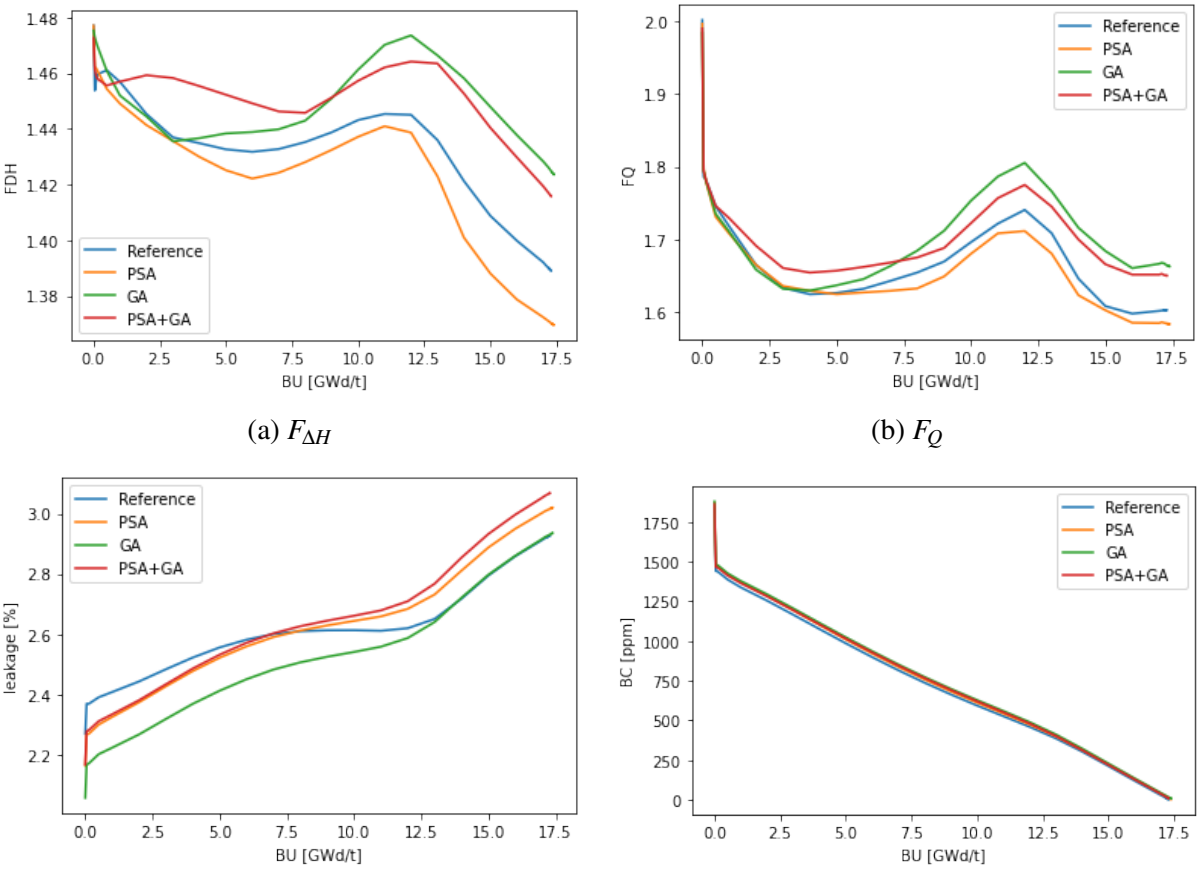
(a) PSA (b) GA (c) PSA+GA

Figure 4.27: Case 2: EC proposed by the algorithms.

high FF), the $F_{\Delta H}$ and F_Q values are higher than reference for most of the cycle. It is, however, kept below a reasonable level (below 1.48) and is more constant during the cycle, compared to other solutions (it oscillates around 1.46 for most of the time). The solution is characterized by the highest F_Q value for MOC, achieving 1.80 vs 1.74. It requires the highest value of BC at BLX (1473 ppm). The configuration (Figure 4.27b) is quite interesting, resembling alternately, rotated by 45 degrees square structures, inscribed in a circle, with successively high and low burnup levels. The center of the core is filled with medium and low burnt FA, with high burnt FA spirally converging towards the center. In general, FAs are more homogeniously distributed in terms of burnup than PSA one.

PSA+GA solution

The hybrid algorithm proposes a solution in between GA and PSA, in terms of power distribution factors. It is, however, characterized by worse leakage properties than the PSA solution. On the other hand, it requires the smallest BC at BLX, than other proposed solutions (1457 ppm), which may be crucial f.i. in the case of the moderator temperature coefficient (MTC). The FA pattern of the solution is more evenly distributed, the outer layer contains a mixture of fresh fuel and high burn-up fuel, mainly arranged in vertical and horizontal lanes. The center, apart from the lanes of burnt FA, contains mainly medium level burnt FA (Figure 4.27c).



(a) $F_{\Delta H}$

(b) F_Q

(c) Leakage

(d) Boron concentration

Figure 4.28: Case 2: Evolution of the FF parameters, for 1/4 symmetry, for PSA, GA, PSA+GA and ref. case.

4.6 Summary

4.6.1 Comparison of the considered cases

As part of the EC optimization, an EPR-like core with two symmetries, 1/8 and 1/4, was considered. On one hand, the smaller symmetry means a significant reduction in the space of possible solutions (from the order of 10^{86} to 10^{46}, which is about 10^{41} times less, see Section 2.1.1). On the other hand, greater symmetry implies imposing more stringent requirements regarding the selection and movement of fuel assemblies in the core. Comparing the advantages of these two scenarios, we may conclude that one is associated with a smaller search space and the other with greater freedom of the algorithm. Therefore, it is not so straightforward to predict which problem may lead to a better solution.

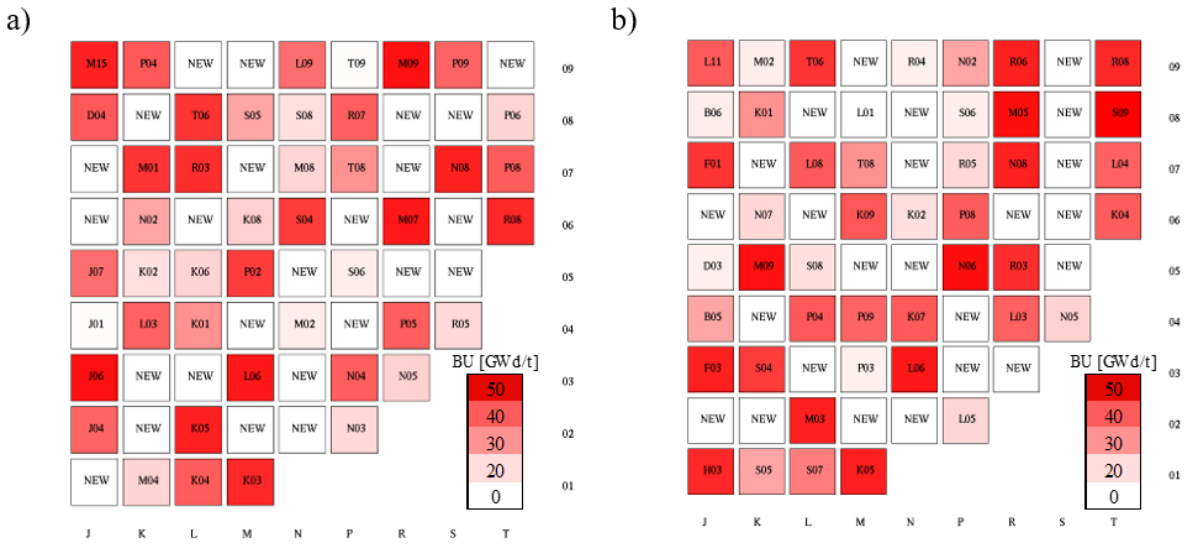


Figure 4.29: EC reloadings for Case 1: PSA+GA 1/8 and Case 2: PSA 1/4 solutions.

This section contains a comparison of selected solutions for 1/8 and 1/4 symmetries. For the 1/8 case, the solution with the highest FF value - obtained through PSA+GA, being a low-leakage type solution, was chosen. For the 1/4 case, the solution with the highest FF value is the GA solution, however, it is characterized by a higher $F_{\Delta H}$ value than the reference case and also has a low-leakage configuration. A slightly lower FF value (1.053 vs 1.061) is achieved by the PSA solution, which at the same time is characterized by a similar leakage to the reference case while having a reduced $F_{\Delta H}$ value, which was determined as the most challenging parameter to optimize (in the case of 1/8 symmetry none of the algorithms managed to improve it). To compare different solutions and focus on $F_{\Delta H}$, despite a slightly lower FF, for the 1/4 symmetry, it was decided to compare PSA+GA 1/8 solution with the PSA 1/4 one.

The simulations carried out show that each version of the algorithm, for each scenario works correctly and is able to propose, in less than 100 generations (around 20 hours), a solution comparable to the reference or slightly better within the defined objective function. In each case, the algorithm proposed a solution with slightly different characteristics. All proposed a longer cycle, which was the main objective of the study but here we focus on two selected solutions: PSA+GA 1/8 which proposed an EC with reduced leakage and PSA 1/4 which proposed a solution with flattened power distribution.

Figure 4.29 present reloading schemes of compared solutions. Each position in the figure represents the position of the assembly in the previous cycle or the position of the FFA (depicted as "NEW"). Color denotes the level of burnup, from white (FFA) to red (ca. 50 GWd/t).

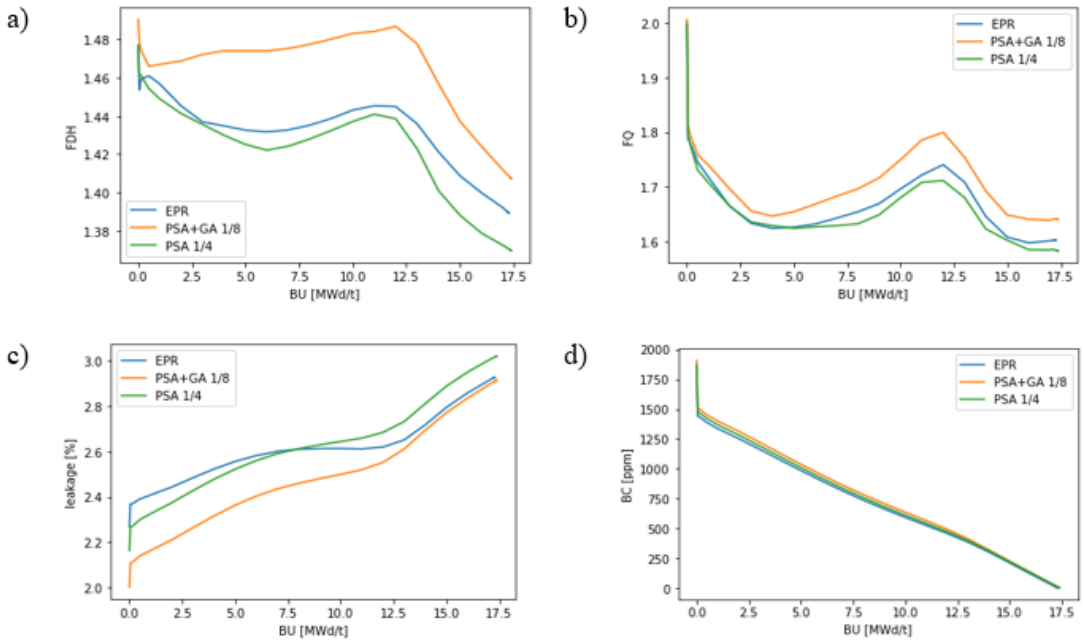


Figure 4.30: Evolution of the FF parameters, for Case 1: PSA+GA 1/8 and Case 2: PSA 1/4 solutions and the reference case.

For the PSA+GA 1/8 solution, most of the FFA are placed at around 3/4 of the radius of the core, and then most of them are either moved to the inner part and come back to the outer part or vice versa, being a combination of the out-in and in-out shuffling methodologies [7]. The solution is characterized by extended lifetime (485.8 vs 483.0 EFPD), lowered leakage (2.50% vs 2.62% on average), relative power at the nodes at the boundary is decreased, especially at the beginning of the cycle at BLX with maximum 0.82 vs 0.98 (see Figure 4.31). During the cycle, the power peaking moves slightly towards the edge of the core, achieving leakage and power

distribution more similar to the reference case (Figure 4.30c and Figure 4.31d-f). On the other hand, throughout the whole cycle, it requires higher BC (Figure 4.30d) and is characterized by higher $F_{\Delta H}$ and F_Q values than the other cases (Figure 4.30a,b). The value of the max(BU_a) is higher by around 2.5 GWd/t compared to the reference case, but still below the required safety level (60 GWd/t).

For the PSA 1/4 solution, FA are also placed in a mixed out-in/in-out manner, however, FFA and FA with different burnup levels are more evenly distributed resulting in flattened power distribution compared to the PSA+GA 1/8 solution, and similar or slightly better when compared to the reference case. The solution at BLX is characterized by slightly higher $F_{\Delta H}$ than the reference case but starting after 0.5 GWd/t it becomes lower (Figure 4.30a). For the solution, leakage is lower at the first half of the cycle, and then it is higher than the reference, achieving 2.64% on average (vs 2.62%, Figure 4.30c).

		Symmetry 1/8		Symmetry 1/4	
	Reference	**PSA+GA**	**ΔPSA+GA-Ref**	**PSA**	**ΔPSA-Ref**
FF (Equation 4.13)	1.000	1.031	+0.031	1.053	+0.053
$\overline{BU}_c$ **[EFPD]**	483.0	485.8	+2.8	486.4	+3.4
$\overline{BU}_c$ **[GWd/t]**	17.23	17.33	+0.10	17.35	+0.12
$max(F_{\Delta H})$	1.477	1.491	+0.014	1.477	0.000
$\mathbf{L^-}$**[%]**	2.62	2.50	-0.09	2.64	+0.02
$max(BU_a)$ **[GWd/t]**	55.0	57.5	+2.5	56.5	+1.5

Table 4.8: Summary of the results for Case 1: PSA+GA 1/8 and Case 2: PSA 1/4.

4.6.2 Conclusions

In this part of the study, EC optimisation was performed to extend the cycle while maintaining or improving selected factors - power distribution, maximum fuel assembly burn-up and neutron leakage. The study assumed two symmetries - 1/8 and 1/4 - and three algorithms were used: GA, PSA and PSA+GA. This section will describe the conclusions that can be drawn from the optimizations carried out.

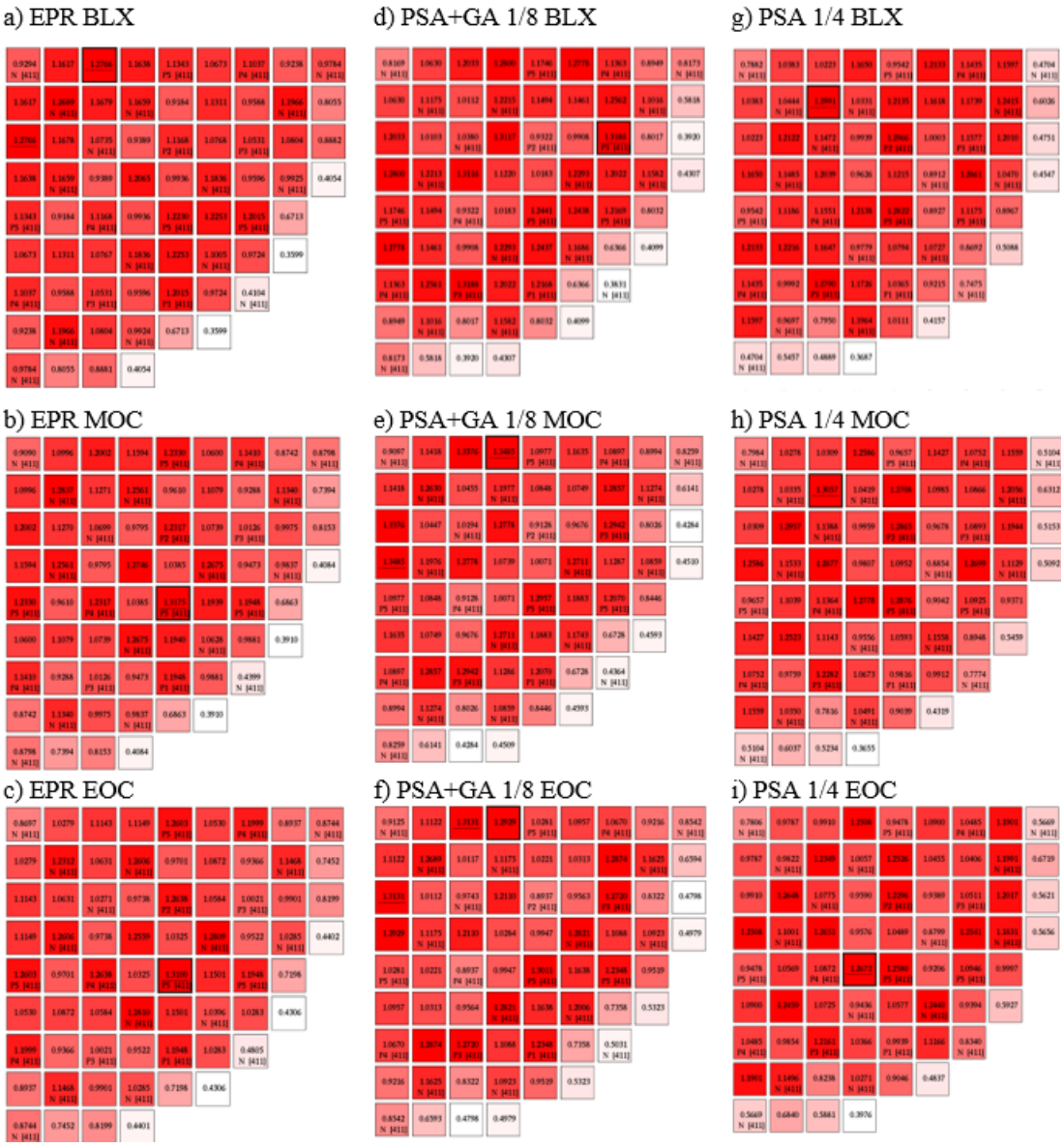


Figure 4.31: Power distribution for the reference EPR (a-c) and proposed solutions Case 1: PSA+GA 1/8 (d-f) and Case 2: PSA 1/4 (g-i) for BLX, middle of the cycle (MOC, 9000 MWd/t) and end of the cycle (EOC, sin17000 MWd/t).

Algorithm and optimization

The analyzed cases showed that each algorithm works correctly and is able to propose from a random configuration a solution comparable to the reference case and even improve the indicated parameters. Importantly, on the basis of the simulations carried out, it is not possible to unambiguously identify the best algorithm for the search for the optimal EC. Admittedly, it can be concluded that the proposed hybrid PSA+GA method gives the best result for 1/8 symmetry, while the GA and PSA give a better result for 1/4 symmetry. This is consistent with the general perception that GA is relatively more specialized in searching high-dimensional spaces compared to other algorithms. However, it is important to emphasize that it was just one realization of each of the algorithms and scenarios, so it is hard to draw any general conclusions. Based on Figure 4.26 and Figure 4.22 one should think of the described optimization as searching for a hyper-volume (red regions in Figure 4.26 and Figure 4.22) or a set of schemes, rather than searching for a single, optimized solution, as the optimized parameters are interdependent.

Symmetry

One of the main questions was to determine which symmetry is more advantageous from the point of view of the optimisation algorithm - 1/8 or 1/4. As previously mentioned, one case involves a much smaller search space but additional constraints, while the other involves a larger space but also more freedom for the algorithm. From the simulations, it might be concluded that it is more optimal to use 1/4 symmetry. For all versions of the algorithm, FF reaches a larger value, with the same number of generations. The impact of the reduced search space for the 1/8 symmetry can be seen in the faster (but to a smaller value) overall convergence of FF. Interestingly, for both scenarios, convergence to similar values occurs after a similar number of generations, and then for the 1/4 symmetry FF continues to increase. This may be counter-intuitive, as it would appear that increasing the search space may slow down the convergence of the algorithm for 1/4 compared to the 1/8 symmetry.

For the conducted scenarios, it may therefore be concluded that searching for solutions with 1/4 symmetry is more effcient. However, it should be mentioned that for 1/8 symmetry, some restrictions were imposed that could be avoided. In this scenario, the division of FFA between octagonal and diagonal parts was predetermined (taken from the reference model), and movement between these parts was not possible, which in general could be changed.

These changes (initial distribution of FFA and their movement) would provide access to solutions that were inaccessible, without significantly increasing the solution space as is the case with 1/4 symmetry. Such an in-between case could be interesting to analyze, but due to its more complicated implementation, it was not considered in this work. The analyses also indicate that increasing the space to the one used in Case 2 does not significantly affect the algorithm's performance and does not disrupt its operation. At this point, it seems that 1/4 symmetry produces better results and can be applied if other technical requirements do not impose more detailed symmetry.

Solutions

The solutions obtained and analysed are mainly characterised by either reduced leakage or reduced power imbalance, but the overall characteristics are otherwise similar to the model reference - the BC waveform, power distribution, cassette alternation, is similar for each of the selected cases. One could say that from a random pattern the algorithm was able to generate a loading pattern with similar characteristics to the one designed by the fuel design team. This demonstrates the incredible potential of a tool that can support the fuel team, automating and speeding up the process. Of course, this study did not consider all the factors that the cycle design team investigates. Among other things, no limits were placed on the BC, especially at the beginning of the cycle, which can have a significant impact on, for example, the moderator temperature coefficient (MTC), which for light-water reactors must necessarily remain negative.

Role of the algorithm

Based on the analyses carried out, it can be shown that the application of optimisation tools has great potential in nuclear engineering and can serve as a tool to support the design and use of nuclear systems. However, it should be recalled that the fuel cycle is an extremely complex problem, combining many physical, neutronic, and thermo-mechanical phenomena and issues. However, existing design is usually based on a similar or pre-existing solution, but an algorithm can provide the fuel pool with an efficient starting point. They can significantly reduce the workload required to design a fuel management strategy, as well as enabling them to go beyond existing designs and potentially offer more efficient and safer solutions.

Future improvements

Although the algorithm has demonstrated correct operation and is able to generate and optimize the selected EC and indicated parameters, there are a number of improvements that can be implemented in the future studies. Firstly, the freedom of the algorithm could be increased by allowing it to select FFAs from a pre-defined set. Then it could also consider using assemblies that are in the pool and come from previous cycles. Another important aspect for testing the algorithm is to allow FA transfer between zones A and B for 1/8 symmetry. Such a solution would combine the advantages of both symmetries - reducing the solution space and at the same time not excluding solutions assuming such cassette movement. A large area of possible change is further work on the form of the objective function, various definitions and weights and coefficients, the penalty function, and above all the list of parameters that it takes into account, such as BC, MTC, linear power density as well as well as mechanical properties such as deformations and bending of FA. Due to the multi-objectivity of this problem, it would also be possible to use other methods that do not require the explicit form of FF (eg. NSGA-II, see Section 1.3.4). Another area is the use of other optimisation techniques, Tabu Search, Particle Swarm Optimisation or Reinforced Learning [44, 165] and the use of the Surrogate Model to replace time-consuming calculations, e.g. with an artificial neural network [166].

5 Optimization of the irradiation scheme

5.1 MARIA reactor

5.1.1 Description of the MARIA reactor

MARIA reactor is an open pool type, research reactor with pressurized fuel channels and nominal thermal power of 30 MW, located at the Polish *National Centre for Nuclear Research* (NCBJ) in Otwock/Świerk. The reactor core consists of a beryllium moderator matrix with pressurized fuel elements, control rods, and irradiation channels (see an example configuration in Figure 5.1). The matrix is surrounded by graphite blocks that act as a reflector. The entire assembly, arranged in a conical shape, is placed in a structure called a basket (see Figure 5.2). Adjacent to the reactor pool is a storage pool (fuel pool) mainly used for the temporal storage of spent fuel and various types of probes. It also serves as an underwater transport route to the hot cells, particularly to the disassembly chamber. A vertical cross-section of the reactor is presented in Figure 5.2.

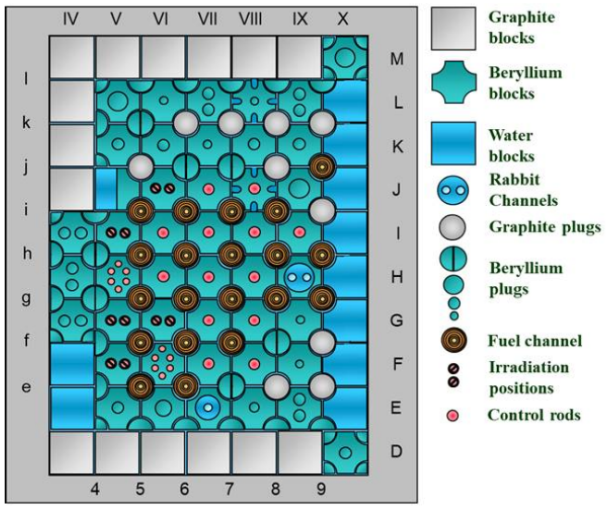


Figure 5.1: Beryllium matrix of the MARIA reactor core (an example configuration) [167].

In the core, there are vertical channels for the production of radioactive isotopes, as well as testing probes and loops. Additionally, for neutron beam research, horizontal channels are extended from the graphite matrix [167,168]. In the matrix, there are fuel elements in the shape of concentric cylinders containing low-enriched uranium (LEU) fuel enriched to 19.7%, in the form of a UO2 dispersion in aluminum. The concentric cylinders are designed as pressurized channels through which water flows, providing a separate cooling system for each fuel element

[168]. The core typically contains about 20-25 such fuel elements. The beryllium and graphite blocks are shaped as truncated cones with square bases, some featuring chamfered corners. The beryllium blocks function as moderators (together with water) and include isotope channels, typically with diameters ranging from 25 to 60 mm. Graphite blocks that surround the beryllium matrix, serve as reflectors and are encased in 2mm thick aluminum. Control rods, safety rods, and the automatic compensation control rod are made of B_4C and are placed in beryllium blocks in channels with 28mm diameter (see an example configuration in Figure 5.1) [167, 168].

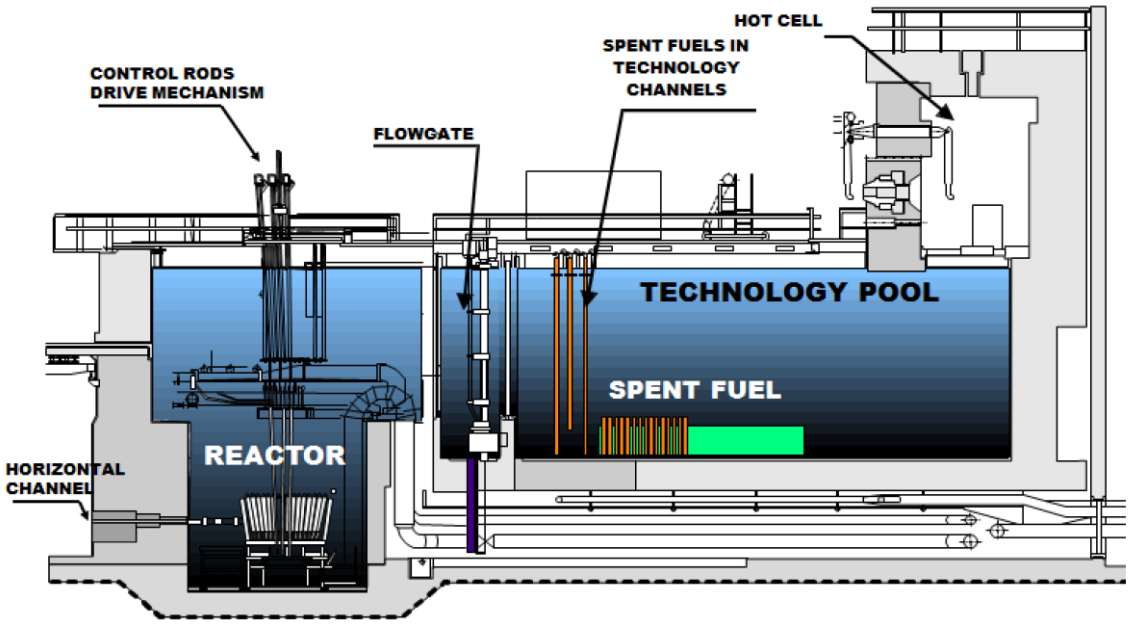


Figure 5.2: Vertical cross-section of the MARIA reactor pool with the core and technology pool [167].

5.1.2 Radionuclides production

The MARIA reactor's design allows different types of irradiation, each designed to aim specific requirements related to neutron flux properties, irradiation duration, and the physical characteristics of target materials. MARIA reactor accommodates several irradiation setups and neutron sources, capable of delivering a neutron flux up to 3×10^{14} n/cm^{2}s for thermal neutrons and 2×10^{14} n/cm^{2}s for fast neutrons.

MARIA contains multiple vertical irradiation channels placed within the region of the reactor core and reflector, mainly used for radionuclide production. A key target material is tellurium dioxide, with around 2000 containers (nearly 200 kg in total), irradiated annually, generating over 1 PBq of ^{131}I. Other important radioisotopes produced in the MARIA facility are S-35, P-32, Sm-153, Yb-169, and Co-60 [169].

Especially important for the MARIA research reactor is ^{99}Mo, radionuclide essential in nuclear medicine (see Section 1.1.2). MARIA plays a crucial role in the irradiation of uranium targets for the production of this isotope and its daughter isotope ^{99m}Tc. In response to a global shortage of ^{99}Mo in 2010 [24], the management of the MARIA reactor committed to irradiating newly designed LEU targets and MARIA has been engaged in the commercial irradiation of uranium targets for the production of ^{99}Mo [24, 170]. Figure 5.1 presents the main sources of Mo-99 production worldwide in the year 2016. MARIA reactor is one of the key suppliers of this nuclide as in 2016, it accounted for 9% of its global production [171]. Despite the listed sites producing Mo-99, shortages of this isotope still occur regularly due to accidents, inspections, and renovations of other reactors in Europe and around the world. One of the solutions that could help improve the situation related to the production of this radioisotope is improving the production process, therefore, the further part of the study focuses on the production of this isotope.

Table 5.1: List of reactor facilities that irradiate targets for Mo-99/Tc-m99 production, as of 2016 [171].

Reactor	**Country**	**Power (MWt)**	**Fuel Type**	**Start of Operation**	**License Expiration**	**Mo-99 Production (Ci/year)**	**Global Capacity (%)**
BR-2	Belgium	100	HEU	1961	2026	210,600	21
HFR	Netherlands	45	HEU	1961	2024	228,000	23
LVR-15	Czech Republic	10	HEU	1957	2028	72,000	7
MARIA	Poland	30	HEU	1974	2030	95,000	9
NRU	Canada	135	HEU	1957	2018	187,200	19
OPAL	Australia	20	LEU	2006	2055*	75,250	8
SAFARI-1	South Africa	20	HEU/LEU	1965	2030	130,700	13
Total						998,750	100

*estimated date

5.2 Theoretical considerations

5.2.1 Introduction to the problem

Before the implementation of the optimization framework was carried out, simple theoretical studies were conducted with the aim of preliminary analysis of a potential optimization problem. Therefore, in this chapter, we undertake a multifaceted analysis to address the question of optimizing the production of a certain nuclide, focusing on the efficiency of utilizing

the target material in a nuclear reactor core. To explore this optimization problem, we conduct three distinct analyses: a direct comparison of irradiation strategies based on different numbers of samples, a theoretical examination of sample shuffling, and an economic analysis of the irradiation cycles.

To compare different methods, we introduce a measure determining the efficiency of utilizing the target material, namely the ratio of the final concentration of isotope B to the initial concentration of the utilized target material A:

$$\eta_B(n) = \frac{N_B(n)}{N_A}. \tag{5.1}$$

In the first problem we wonder whether, from the standpoint of efficiency in utilizing the target material, it is more effective to place a smaller number of samples in the core for a longer period (e.g., for two cycles) or a larger number of samples for a shorter period (e.g., one cycle each). On one hand, irradiating one sample for a longer period will lead to greater utilization of material A, but on the other hand, the amount of material A will decrease with irradiation, slowing down the production rate. Insertion of two samples will, of course, lead to the production of a larger amount of material B but also at the expense of using a larger amount of material A (see Figure 5.3). The outcome will, of course, depend on the value of the flux, the value of the cross-section, and the decay constant of the produced isotope.

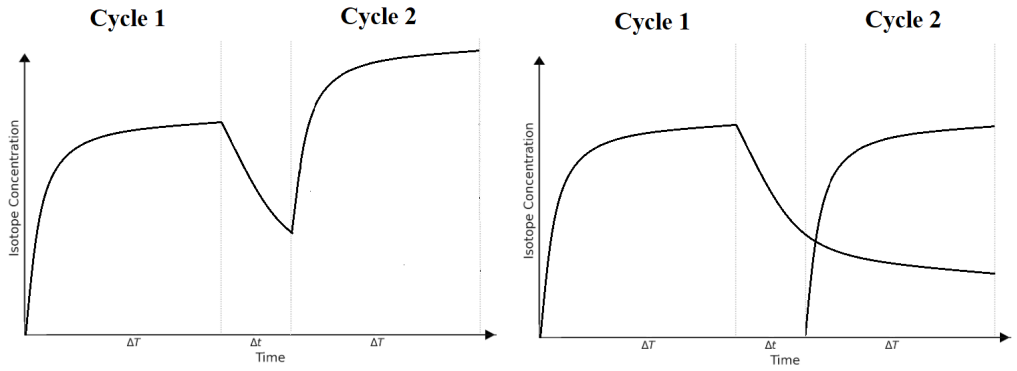


Figure 5.3: A schematic concept of two different irradiation schemes. Irradiation of one sample for two cycles (left) and irradiation of two samples for one cycle each (right).

In the second problem, we try to answer the question of whether, in the case of irradiating many samples over many cycles, it is more efficient to place the samples in given positions for the entire duration of the cycles or rather to exchange the positions of the samples between the cycles. Assuming we have two positions in the reactor: the first one characterized by a high

reaction rate and high production efficiency (black curve in Figure 5.4), and the second position characterized by a low reaction rate (red curve in Figure 5.4), swapping the samples' positions between cycles should increase production efficiency. It can be expected that since, throughout a cycle, nuclide A depletes faster in the first position than in the second, swapping the samples should effectively increase the exposure of nuclide A and enhance production efficiency.

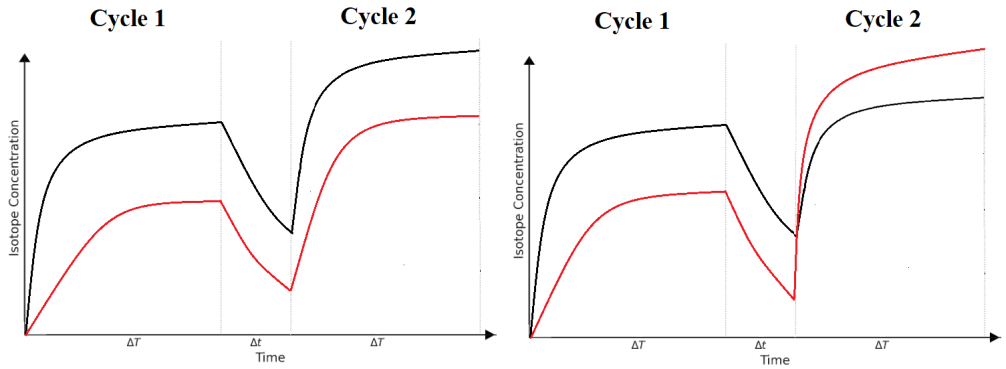


Figure 5.4: The schematic concept of irradiating two samples over two cycles in two positions - one with a high reaction rate (black curve) and one with a low reaction rate (red curve), for two scenarios: irradiation without swapping (left chart) and with swapping positions between cycles (right chart).

In the last problem, we tackle the issue of production costs and the most cost-effective number of cycles that a sample should spend in the reactor core. On one hand, a smaller number of cycles means lower irradiation costs, but on the other hand, it leads to worse utilization of the sample material, necessitating the purchase of greater quantities. Similarly, longer irradiation time means higher costs associated with the cycle but lower costs of the sample.

The questions considered become an optimization problem for which, in most cases, we are unable to easily predict the answer. In order to initially analyze the problems, we perform theoretical considerations using a simplified case where one isotope A is irradiated to produce another isotope B. This scenario assumes a starting concentration N_A subjected to a monoenergetic, constant in time neutron flux ϕ, with isotope B being produced solely and directly through the irradiation of isotope A. The production probability is proportional to the cross-section for the reaction $\sigma_{A\to B}$.

5.2.2 Simplification of the problem

For an infinite and homogenous material, the description of the evolution of the nuclide concentrations can be represented as a set of rate equations [172]:

$$\frac{dN_i}{dt} = \sum_j \left(\lambda_{ji} + \sigma_{ji}\phi(t)\right) N_j, \tag{5.2}$$

where:

N_i is the number of nuclides i at time t and ϕ is neutron flux $\left(\mathrm{cm}^{-2}\mathrm{s}^{-1}\right)$,

for $j \neq i$:

$+\lambda_{ji}$ is the decay constant of nuclide j producing i $\left(\mathrm{s}^{-1}\right)$,

$+\sigma_{ji}$ is the cross-section for reactions cucurring on j producing i $\left(\mathrm{cm}^2\right)$

and for $j = i$:

$-\lambda_{jj}$ is the total decay constant of nuclide j $\left(\mathrm{s}^{-1}\right)$,

$-\sigma_{jj}$ is the total cross-section of reactions occurring on j $\left(\mathrm{cm}^2\right)$.

If we assume that nuclide B is produced directly by irradiation of isotope A, only isotope B is produced from this isotope with production probability described by cross-section $\sigma_{A\to B}$, isotope B decays, with a decay constant λ_B, there is no burnup of isotope B and no other reactions occur, then we consider a simplified problem derived from the Equation 5.2:

$$\frac{dN_B(t)}{dt} = (N_A - N_B(t))\sigma_{A\to B}\phi - \lambda_B N_B(t), \tag{5.3}$$

which is a first-order linear ordinary differential equation that we can rewrite in the form:

$$N_B' + aN_B = b, \tag{5.4}$$

where:

$$a = \sigma_{A\to B}\phi + \lambda_B,$$

$$b = N_A\sigma_{A\to B}\phi.$$

We solve the ODE by multiplying the Equation 5.4 by a factor e^{at} and then integrating both sides. Assuming that $N_B(t=0) = 0$, the solution is:

$$N_B(t) = N_A\frac{\sigma_{A\to B}\phi}{\sigma_{A\to B}\phi + \lambda_B}\left[1 - e^{-(\sigma_{A\to B}\phi+\lambda_B)\cdot t}\right]. \tag{5.5}$$

Thus, the concentration of isotope B, created by irradiation of isotope A, after irradiation time ΔT and cooling time Δt can be written as:

$$N_B(\Delta T, \Delta t) = N_A\frac{\sigma_{A\to B}\phi}{\sigma_{A\to B}\phi + \lambda_B}\left[1 - e^{-(\sigma_{A\to B}\phi+\lambda_B)\cdot \Delta T}\right]\cdot e^{-\lambda_B \Delta t}. \tag{5.6}$$

To simplify, we introduce $\alpha_{\Delta\tau}$ and $\beta_{\Delta\tau}$:

$$\alpha_{\Delta\tau} = \frac{\sigma_{A\to B}\phi}{\sigma_{A\to B}\phi + \lambda_B}\left[1 - e^{-(\sigma_{A\to B}\phi+\lambda_B)\cdot\Delta\tau}\right], \tag{5.7}$$

$$\beta_{\Delta\tau} = e^{-\lambda_B\Delta\tau}. \tag{5.8}$$

Both $\alpha_{\Delta\tau} \in [0,1]$ and $\beta_{\Delta\tau} \in [0,1]$. The factor $\alpha_{\Delta\tau}$ describes both the creation and decay of isotope B over time interval $\Delta\tau$, and it can be interpreted as efficiency (defined in terms of Equation 5.1), which builds up during continous irradiation. $\beta_{\Delta\tau}$ describes the fraction of the isotope that did not undergo decay after time $\Delta\tau$. Then, the concentration of isotope B in Equation 5.6 can be written as:

$$N_B(\Delta T, \Delta t) = N_A\alpha_{\Delta T}\beta_{\Delta t}. \tag{5.9}$$

5.2.3 General solution for a sample irradiation for *n* cycles

Analyzing the solution from the previous section, we can determine a recursive formula for the number of nuclides of nuclide B, at the end of the $n-th$ cycle of length ΔT and a cooling time of Δt between each of them:

$$N_B^{(n)} = N_B^{(n-1)}\beta_{\Delta T+\Delta t} + (N_A - N_B^{(n-1)})\alpha_{\Delta T}. \tag{5.10}$$

We can simplify the equation to:

$$N_B^{(n+1)} = N_B^{(n)}a + b, \tag{5.11}$$

which is a nonhomogenous linear recurrence equation with constant coefficients:

$$a = \beta_{\Delta T+\Delta t} - \alpha_{\Delta T}, \quad b = N_A\alpha_{\Delta T}.$$

We find the steady state value $N_B^* = \frac{b}{1-a}$, and by adding this expression to both sides of Equation 5.11, we convert the problemn into a homogeneous form:

$$(N_B^{(n+1)} - N_B^*) = a(N_B^{(n)} - N_B^*) \implies \tilde{N}_B^{(n+1)} = a\tilde{N}_B^{(n)},$$

the solution of which is $\tilde{N}_B^{(n)} = C \cdot a^n$. Then, returning to the heterogenous equation, and substituting back the constants and assuming that the initial condition $N_B^{(0)} = 0$ (no nuclides B at the beginning of the irradiation), we solve the heterogenous problem and obtain the solution:

$$\boxed{N_B^{(n)} = \frac{N_A \alpha_{\Delta T}[1 - (\beta_{\Delta T + \Delta t} - \alpha_{\Delta T})^n]}{\alpha_{\Delta T} - \beta_{\Delta T + \Delta t} + 1}}. \tag{5.12}$$

This formula allows for direct calculation of built-up of the numbers of nuclides of isotope B, $N_B^{(n)}$ after irradiation for n cycles, using the specified parameters $\alpha_{\Delta T}$, $\beta_{\Delta T+\Delta t}$ related to the irradiation conditions, and initial number of atoms of the target isotope N_A.

Comparison of the solution with FISPACT-II calculations:

To determine the reliability of the simplified model, calculations were performed using Solution 5.12 and the FISPACT-II nuclide inventory code, which solves the exact nuclide evolution equations [172] (all major pathways and reactions). The calculations were carried out for the case of producing ^{99}Mo from irradiation of uranium samples, up to 5 cycles. In the calculations, the following assumptions were made:

- The sample contains $N_{U235} = 10^{24}$ atoms of ^{235}U,
- It is assumed that the target is irradiated with a monoenergetic neutron flux of density $\Phi = 10^{13}\ cm^{-2}s^{-1}$,
- The irradiation occurs over five cycles lasting $\Delta T = 5\ d$ each, with a break between them of $\Delta t = 2\ d$ (typical values for the MARIA reactor [167]),
- The cross-section for the production of ^{99}Mo from the ^{235}U is $\sigma_{U235 \to Mo99} = 37\ b$ [20],
- The decay constant of ^{99}Mo is $\lambda_{Mo99} = 2.9 \cdot 10^{-6}\ s^{-1}$ [173].

Figure 5.5 presents a comparison of the evolution of nuclide B numbers for the simplified model and using the FISPACT-II code for the first two cycles, for which most analyses are performed. As can be seen from the chart, the curves are very close to each other, deviating by about 4% (Table 5.2) at the end of the second cycle.

Table 5.2 presents a comparison of the results for five cycles. One can see that for more cycles, the error propagates for a larger number of cycles but most of the time does not exceed 5%. Since in this part, we are only interested in preliminary analysis and comparison of methods, not precise results, and we mostly consider the first two cycles, it can be concluded that such calculation accuracy will be sufficient for the considered case. Additionally, from

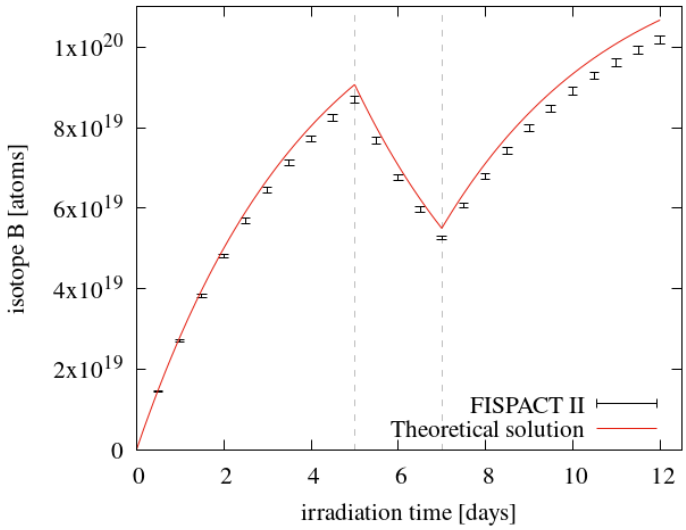


Figure 5.5: Comparison between Equation 5.12 results and FISPACT-II calculations*, for the first two cycles.

*Uncertainties for the FISPACT-II code calculations come from the analysis of production pathways and uncertainties in nuclear data [172]. Most of the uncertainties are on the order of tenths of a percent, but for visibility, the uncertainties have been rounded to one full percent.

Table 5.2: Comparison between Equation 5.12 results and FISPACT-II calculations.

n	Equation 5.12	FISPACT-II	Δ [%]
1	$9.079 \cdot 10^{19}$	$8.705 \cdot 10^{19}$	4.29
2	$1.063 \cdot 10^{20}$	$1.019 \cdot 10^{20}$	4.34
3	$1.089 \cdot 10^{20}$	$1.041 \cdot 10^{20}$	4.62
4	$1.094 \cdot 10^{20}$	$1.043 \cdot 10^{20}$	4.92
5	$1.094 \cdot 10^{20}$	$1.040 \cdot 10^{20}$	5.22

Table 5.2, it can be inferred that for ^{99}Mo production, irradiation for more than about 4 cycles will not yield further effect (the nuclide concentration saturates). However, in considerations related to the economics of production, for comparative purposes, we consider irradiation for up to five cycles.

5.2.4 Optimal number of samples

In this part, using calculations from the previous section, we will conduct the first of the analyses considered. For this purpose, we will try to determine the final concentration N_B at the end of each of the two cycles, depending on the chosen irradiation scheme, and assess the efficiency of utilizing the target material (Equation 5.1).

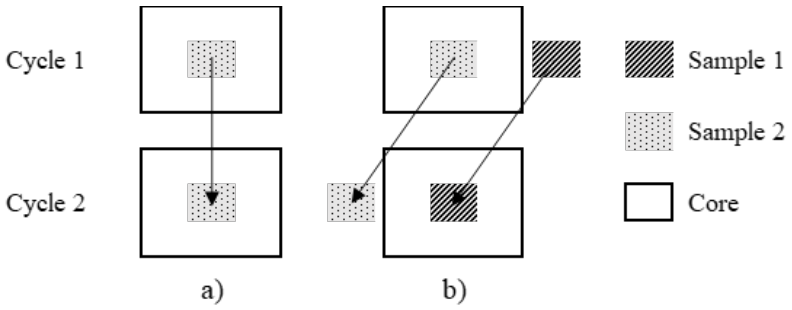


Figure 5.6: The schematic concept of irradiating one sample through two cycles a) and replacing the sample with a new one after the first cycle b).

Case 1: one sample irradiated for two cycles (Figure 5.6a)

We calculate the final concentration of nuclide B, for **one sample** irradiated for two cycles and cooling in between (1+1):

$$N_B^{(1+1)} = N_A \alpha_{\Delta T} \cdot \beta_{\Delta t+\Delta T} + (N_A - N_A \alpha_{\Delta T}) \cdot \alpha_{\Delta T}. \tag{5.13}$$

$$N_B^{(1+1)} = N_A \alpha_{\Delta T} (\beta_{\Delta t+\Delta T} + 1 - \alpha_{\Delta T}), \tag{5.14}$$

and corresponding efficiency of nuclide production:

$$\eta_B^{(1+1)} = \frac{1}{N_A} \cdot N_A \alpha_{\Delta T} (\beta_{\Delta t+\Delta T} + 1 - \alpha_{\Delta T}) = \alpha_{\Delta T} (\beta_{\Delta t+\Delta T} + 1 - \alpha_{\Delta T}). \tag{5.15}$$

Case 2: two samples irradiated for one cycle (Figure 5.6b)

For **two samples** irradiated, the final concentration for the first (1) and second (2) sample will equal:

$$N_B^{(1)} = N_A \alpha_{\Delta T} \beta_{\Delta t+\Delta T},$$

$$N_B^{(2)} = N_A \alpha_{\Delta T}.$$

Thus, the total concentration (1+2):

$$N_B^{(1+2)} = N_A \alpha_{\Delta T}(1+\beta_{\Delta t+\Delta T}), \tag{5.16}$$

and the associated nuclide production efficiency:

$$\eta_B^{(1+2)} = \frac{1}{2N_A} \cdot N_A \alpha_{\Delta T}(1+\beta_{\Delta t+\Delta T}) = \frac{1}{2} \cdot \alpha_{\Delta T}(1+\beta_{\Delta t+\Delta T}). \tag{5.17}$$

Comparison of Case 1 and Case 2:

Here, we calculate **the ratio of the efficiencies** of Case 1 and Case 2 to determine when irradiation by the first method (one sample for two cycles) leads to a lower efficiency:

$$\frac{\eta^{(1+1)}}{\eta_B^{(1+2)}} < 1, \tag{5.18}$$

$$\frac{\eta^{(1+1)}}{\eta_B^{(1+2)}} = \frac{\alpha_{\Delta T}(\beta_{\Delta t+\Delta T}+1-\alpha_{\Delta T})}{\frac{1}{2} \cdot \alpha_{\Delta T}(1+\beta_{\Delta t+\Delta T})} = 2\frac{\beta_{\Delta t+\Delta T}+1-\alpha_{\Delta T}}{1+\beta_{\Delta t+\Delta T}},$$

$$2\frac{\beta_{\Delta t+\Delta T}+1-\alpha_{\Delta T}}{1+\beta_{\Delta t+\Delta T}} < 1,$$

$$2(\beta_{\Delta t+\Delta T}+1-\alpha_{\Delta T}) < 1+\beta_{\Delta t+\Delta T},$$

$$1+\beta_{\Delta t+\Delta T} < 2\alpha_{\Delta T}.$$

We introduce the reaction rate:

$$r = \sigma_{A\to B}\phi. \tag{5.19}$$

Then:

$$\frac{2r}{r+\lambda_B}\left[1-e^{-(r+\lambda_B)\cdot\Delta T}\right] > 1+\beta_{\Delta t+\Delta T},$$

$$2r\left[1-e^{-(r+\lambda_B)\cdot\Delta T}\right] > (1+\beta_{\Delta t+\Delta T})(r+\lambda_B). \tag{5.20}$$

We can compare two functions $f(r)$ and $g(r)$ of the left and right side of the inequality (Figure 5.7):

$$f(r) = 2r\left[1-e^{-(r+\lambda_B)\cdot\Delta T}\right],$$

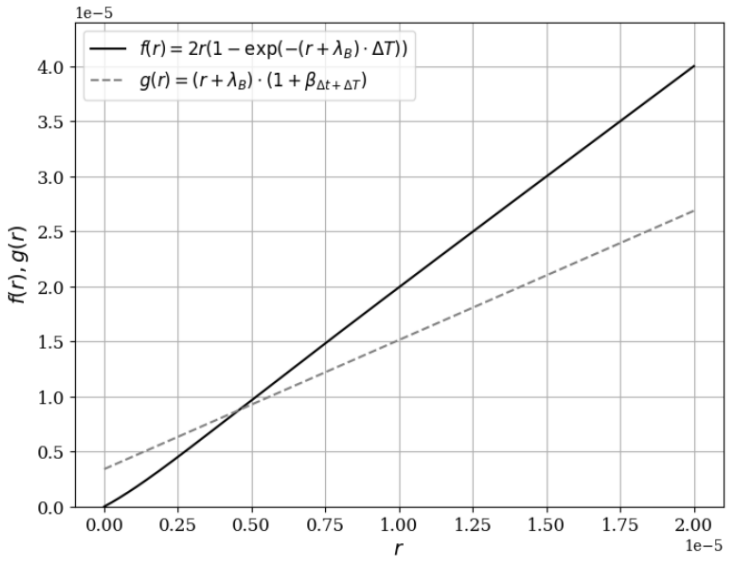


Figure 5.7: Comparison of left and right side of the Inequality 5.20, $\lambda_B = 2.9 \times 10^{-6} s^{-1}, \Delta T = 4.3 \times 10^5 s, \beta_{\Delta t+\Delta T} = 0.173$.

$$g(r) = (1+\beta_{\Delta t+\Delta T})(r+\lambda_B).$$

Because $e^{-(r+\lambda_B)\cdot\Delta T} > 0$:

$$2r > 2r\left[1 - e^{-(r+\lambda_B)\cdot\Delta T}\right] > (1+\beta_{\Delta t+\Delta T})(r+\lambda_B). \tag{5.21}$$

Thus:

$$2r > (r+\lambda_B)(1+\beta_{\Delta t+\Delta T}),$$

$$2r - r(1+\beta_{\Delta t+\Delta T}) > \lambda_B(1+\beta_{\Delta t+\Delta T}),$$

$$r(1-\beta_{\Delta t+\Delta T}) > \lambda_B(1+\beta_{\Delta t+\Delta T}),$$

$$r > \lambda_B \frac{1+\beta_{\Delta t+\Delta T}}{1-\beta_{\Delta t+\Delta T}}. \tag{5.22}$$

Due to the upper estimate of the inequality, introduced in Equation 5.21, the threshold value of the reaction rate r, for which the first method is characterized by lower efficiency, equals at least:

$$\boxed{r_{min} \geq \tilde{r}_{min} = \lambda_B \frac{1+e^{-\lambda_B(\Delta t+\Delta T)}}{1-e^{-\lambda_B(\Delta t+\Delta T)}}}. \tag{5.23}$$

For the given flux and neutron spectrum, the minimal value r_{min} depends only on the material type, as the irradiation and cooling times are constant for the MARIA reactor ($\Delta T = 5\ d$ and $\Delta t = 2\ d$). Therefore, for example for ^{99}Mo production from ^{235}U ($\sigma_{A\to B} \approx 37\ b$, $\lambda_B = 2.92 \cdot 10^{-6}\ s^{-1}$ [20]), the minimum value of $r_{min} > \tilde{r}_{min} = 3.56 \cdot 10^{-6}\ s^{-1}$ which corresponds to a neutron flux $\tilde{\phi}_{min} = 9.6 \cdot 10^{20}\ cm^{-2}s^{-1}$. The MARIA reactor core offers irradiation with the thermal neutron flux up to $2 \cdot 10^{15}\ cm^{-2}s^{-1}$ [167] which is several orders of magnitude less than the calculated lower boundary of the threshold value $\tilde{r}_{min}$. Therefore we can conclude that for this simplified problem, for the ^{99}Mo production, for the MARIA reactor conditions, the first irradiation scheme will always lead to a better target material usage in terms of the efficiency defined as Equation 5.1.

5.2.5 Shuffeling of the samples

In this part, we consider a scenario with two samples and two different available positions for the irradiation. Each of the samples contains an equal number of nuclides of element N_A, and the positions are characterized by reaction rates r_1 and r_2.

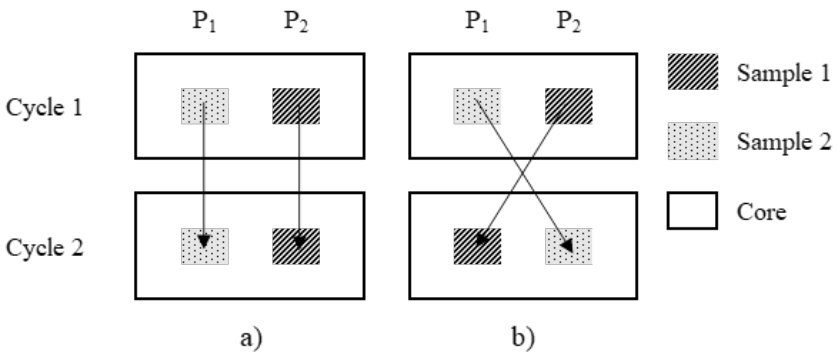


Figure 5.8: The schematic concept of irradiating samples in each position for two cycles a) and swapping the samples between cycles b).

In this analysis, we will also assume irradiation for two cycles. We will try to answer the question of whether, from the efficiency standpoint as defined by Equation5.1, it is better to irradiate each of the samples in one position for two cycles or to swap the samples after the first cycle (see Figure 5.8). Using the Equation 5.9 which is a solution to the simplified Equation 5.4, we will briefly consider these two scenarios.

To simplify the notation, the factor $\alpha_{\Delta T}$ (Equation 5.7) associated with the production and decay of isotope B during the cycle, will be denoted as follows:

For irradiation in position 1:

$$\alpha_1 = \alpha_{\Delta T,1} = \frac{r_1}{r_1 + \lambda_B}\left[1 - e^{-(r_1+\lambda_B)\cdot\Delta T}\right].$$

And in position 2:

$$\alpha_2 = \alpha_{\Delta T,2} = \frac{r_2}{r_2 + \lambda_B}\left[1 - e^{-(r_2+\lambda_B)\cdot\Delta T}\right].$$

The factor $\beta_{\Delta t}$ (Equation 5.8), associated with the decay of isotope B, does not depend on the position in the core and for both cases will be denoted as:

$$\beta = \beta_{\Delta T+\Delta t} = e^{-\lambda_B(\Delta T+\Delta t)}.$$

Case 1: no shuffling (Figure 5.8a)

Using the earlier analyses, we calculate the number of nuclides B produced by **irradiating for two cycles at the same position**, i.e. the first sample in the 1st position $(1 \to 1)$ and the second sample in the 2nd position $(2 \to 2)$:

$$N_B^{(1\to1)} = N_A \cdot \alpha_1(\beta + 1 - \alpha_1), \tag{5.24}$$

$$N_B^{(2\to2)} = N_A \cdot \alpha_2(\beta + 1 - \alpha_2). \tag{5.25}$$

We calculate the efficiency of such an irradiation scheme as the sum of the final number of nuclide B divided by the total number of nuclides of isotope A used:

$$\eta_B^{\binom{1\to1}{2\to2}} = \frac{N_B^{(1\to1)} + N_B^{(2\to2)}}{2N_A} = \frac{1}{2}(\alpha_1\beta + \alpha_1 - \alpha_1^2 + \alpha_2\beta + \alpha_2 - \alpha_2^2),$$

$$\eta_B^{\binom{1\to1}{2\to2}} = \frac{1}{2}\left[(\alpha_1 + \alpha_2)(\beta + 1) - (\alpha_1^2 + \alpha_2^2)\right]. \tag{5.26}$$

Case 2: with shuffling (Figure 5.8b)

Now we consider the case where **the samples are swapped between the cycles**. Then, the total number of nuclides B produced in the sample, irradiated in the 1st position and then in the 2nd position $(1 \to 2)$, equals:

$$N_B^{(1\to 2)} = N_A \left[\alpha_1 \cdot \beta + (1-\alpha_1)\cdot \alpha_2\right], \tag{5.27}$$

and for the sample irradiated in the second and then in the first position $(2 \to 1)$:

$$N_B^{(2\to 1)} = N_A \left[\alpha_2 \cdot \beta + (1-\alpha_2)\cdot \alpha_1\right]. \tag{5.28}$$

We calculate the efficiency of such an irradiation scheme:

$$\eta_B^{\binom{1\to 2}{2\to 1}} = \frac{N_B^{(1\to 1)} + N_B^{(2\to 2)}}{2N_A}, \tag{5.29}$$

$$\eta_B^{\binom{1\to 2}{2\to 1}} = \frac{1}{2}\left[(\alpha_1+\alpha_2)(\beta+1) - 2\alpha_1\alpha_2\right]. \tag{5.30}$$

Comparison of Case 1 and Case 2:

Now, we calculate **the ratio of the efficiencies** to compare these two irradiation schemes and determine when the first one (no change in the position of the samples) leads to a worse efficiency.

$$\frac{\eta_B^{\binom{1\to 1}{2\to 2}}}{\eta_B^{\binom{1\to 2}{2\to 1}}} = \frac{(\alpha_1+\alpha_2)(\beta+1) - (\alpha_1^2+\alpha_2^2)}{(\alpha_1+\alpha_2)(\beta+1) - 2\alpha_1\alpha_2} < 1, \tag{5.31}$$

$$(\alpha_1+\alpha_2)(\beta+1) - (\alpha_1^2+\alpha_2^2) < (\alpha_1+\alpha_2)(\beta+1) - 2\alpha_1\alpha_2,$$

$$\alpha_1^2 + \alpha_2^2 - 2\alpha_1\alpha_2 > 0,$$

$$(\alpha_1-\alpha_2)^2 > 0. \tag{5.32}$$

Analyzing the inequality in Equation 5.32, we conclude that it is always satisfied except for one case when $\alpha_1 = \alpha_2$, i.e., when the reaction rates in those positions are the same (assuming the same isotopes and irradiation times). This is a very specific case, and then swapping the samples does not affect the irradiation and leads to exactly the same efficiency.

Therefore, we can conclude that swapping or shuffling the samples always leads to better (or at least the same) production efficiency, in terms of the definition in Equation 5.1.

5.2.6 The most cost-effective number of cycles

Beyond physical considerations, which involve searching for irradiation conditions (time, position) of samples in order to achieve maximum efficiency, factors related to the economics of producing a given isotope are also important. It is understandable that the cost of producing a given nuclide includes the cost of the nuclide from which it is derived (the sample material), but also significant is the cost associated with the time the sample material spends under irradiation (the cost of reactor operation). While longer irradiation may lead to better utilization of the material of the sample, it can turn out to be economically unfeasible. Unfortunately, while it's relatively straightforward to determine the cost of the sample, defining the cost of reactor operation is a challenging task. Due to numerous variables (operational costs, fuel costs, spent fuel management, decommissioning, service, staffing) and non-obvious financing mechanisms, determining the actual costs of reactor operation is in itself a research topic [174]. In this part, however, we will attempt a simple analysis of cost relationships and potential optimal solutions.

In this analysis, we assume that the goal is to produce a constant, expected number of N_B^{EXP} atoms (number, mass, or concentration) of nuclide B from nuclide A. In a simple model, we assume that the cost $C(n)$ of producing N_B^{EXP} nuclides consists of two components: the first one is related to cycle costs, proportional to the time the sample spends in the core, and proportional to its mass, as we assume that providing the required number of atoms N_A is done by increasing the mass of the sample (so the volume it occupies in the core), not its enrichment (enrichment remains constant). The second component is related to the costs of purchasing or producing N_A nuclides of the sample.

The number of nuclides A (number, mass, or concentration), N_A, required to produce N_B^{EXP} depends on the chosen scenario and efficiency $\eta_B(n)$:

$$N_A(n) = \frac{N_B^{EXP}}{\eta_B(n)}. \tag{5.33}$$

Therefore, the total production cost we write as a sum of the two components:

$$C(n) = C_1 \cdot n \cdot N_A(n) + C_2 \cdot N_A(n), \tag{5.34}$$

where C_1 is the unit cost of occupying space in the core per cycle, and C_2 is the cost of the target material. By substituting the expression from Equation 5.33, we obtain:

$$C(n) = C_1 \cdot n \cdot \frac{N_B^{EXP}}{\eta_B(n)} + C_2 \cdot \frac{N_B^{EXP}}{\eta_B(n)}.$$

Given the fixed number of expected nuclides N_B^{EXP} to be produced and costs of the sample C_2, we can scale $C(n)$ as:

$$\hat{C}(n) = \frac{C(n)}{C_2 \cdot N_B^{EXP}} = \frac{C_1}{C_2} \cdot \frac{n}{\eta_B(n)} + \frac{1}{\eta_B(n)}, \tag{5.35}$$

$$\hat{C}(n) = \frac{1}{\eta_B(n)}(\xi \cdot n + 1), \tag{5.36}$$

where:

$$\xi = \frac{C_1}{C_2} \tag{5.37}$$

is a ratio between the unit **costs of the cycle** C_1 and unit **costs of the sample** C_2. Because the reference case is irradiation for one cycle, we will analyze all cases using a relative cost, so compared to the costs for $n = 1$:

$$\hat{C}_{rel}(n) = \frac{\hat{C}(n)}{\hat{C}(1)}. \tag{5.38}$$

Now we can treat ξ (Equation 5.37) as a control parameter and consider what is the optimum number of cycles, in terms of the economics of producing isotope B, depending on the ratio of cycle costs and material A. Figure 5.9 shows the relative cost function $C_{rel}(n)$ (Equation 5.38) for the parameter $\xi \in [10^{-3}, 2.5 \cdot 10^{-1}]$. In addition, through analysis of the case when $C_{ref}(k) = 1$, a critical value was calculated for the parameter ξ_k, for which the cost production of nuclides B through k cycles, is the same as for the one-cycle irradiation:

$$\boxed{\xi_k = \frac{\frac{1}{\eta_B(1)} - \frac{1}{\eta_B(k)}}{\frac{k}{\eta_B(k)} - \frac{1}{\eta_B(1)}}.} \tag{5.39}$$

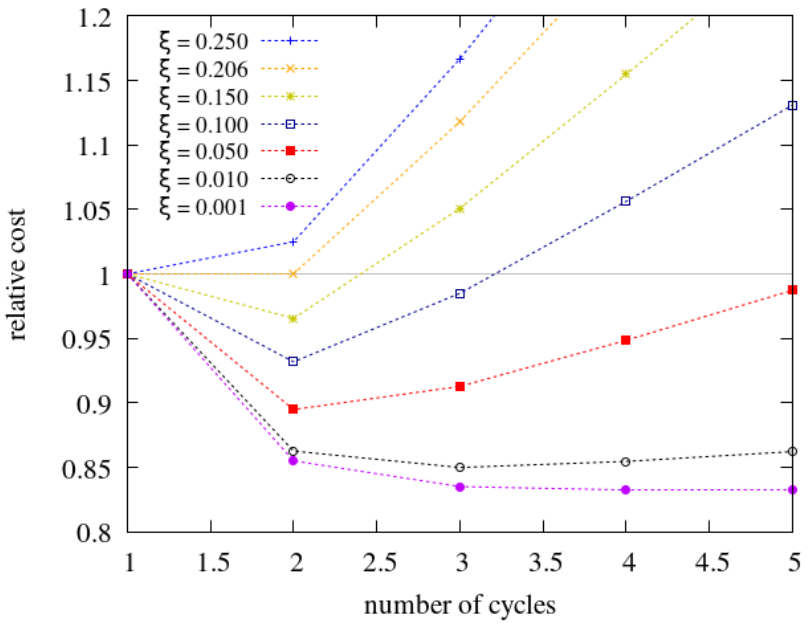


Figure 5.9: Relative cost of irradiation (Equation 5.38) for n cycles, with different ratios of cycle to sample costs ξ.

By substituting the data for the production of ^{99}Mo, we are able to determine the critical value $\xi_2 = 0.206$ for which is the marginal value of the ratio of cycle and sample costs below which it may be more profitable to produce nuclides using more than 1 cycle. Analyzing the obtained result, we can conclude that if the cost of irradiating a given mass of the sample for 1 cycle is less than about 20% of the cost of purchasing this sample, it may be economically more profitable to produce it by irradiation for two or more cycles (see Figure 5.9). Above this value, the costs of the cycle exceed the costs of the sample, and it becomes more cost-effective to use the irradiation scheme with one cycle (or even part of a cycle).

5.2.7 The most cost-effective length of the cycle

It is possible that the cost of placing a given sample per cycle will be well over 20% of its production or purchasing cost. In this case, one can analyze the case of continuous irradiation for a time other than the one assumed earlier and search for a more optimal cycle length (in terms of the production of a given isotope).

For this purpose, we will define continuous efficiency, which is the efficiency that builds up during continuous irradiation, without interruption. This efficiency is defined identically to the alpha coefficient in Equation 5.7 with continous variable t, and takes the following form:

$$\varepsilon_B(t) = \frac{r}{r+\lambda_B}\left[1 - e^{-(r+\lambda_B)\cdot t}\right]. \tag{5.40}$$

With analogy to Equation 5.36, we define the irradiation cost function, for continuous time:

$$\tilde{C}(t) = \frac{1}{\varepsilon_B(t)}(\xi \cdot t + 1), \tag{5.41}$$

And the relative cost function:

$$\tilde{C}_{rel}(t) = \frac{\tilde{C}(t)}{\tilde{C}(5d)}. \tag{5.42}$$

The relative cost function is presented in Figure 5.10 for defferent values of the cycle-to-sample cost ξ, for continous irradiation up to 14 days. As can be expected, for higher values of ξ (so the higher relative cycle costs), the optimal irradiation time shifts towards smaller values.

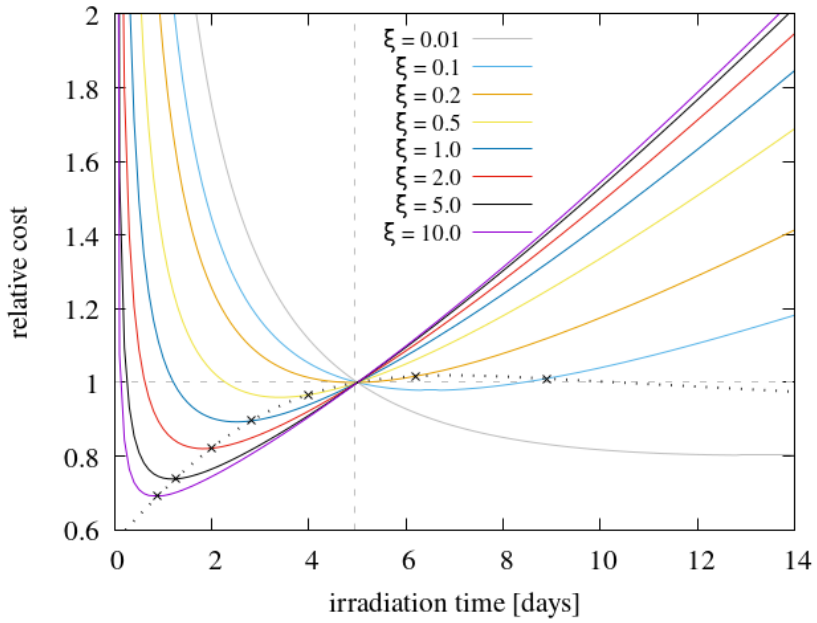


Figure 5.10: Relative cost of continous irradiation (Equation 5.42) for t days, with different ratios of cycle to sample costs ξ.

To estimate how the optimal number of continuous irradiation days depends, we will calculate the derivative of the cost function defined in Equation 5.41.

$$\tilde{C}(t)' = \left[\frac{1}{\varepsilon_B(t)}(\xi \cdot t + 1)\right]' = \left[\frac{(r+\lambda_B)\cdot(\xi \cdot t + 1)}{r \cdot \left[1 - e^{-(r+\lambda_B)\cdot t}\right]}\right]', \tag{5.43}$$

$$\tilde{C}(t)' = \frac{e^{(r+\lambda_B)t}\left[\xi(e^{(r+\lambda_B)t}-1)-(r+\lambda_B)\cdot(\xi t+1)\right]}{\frac{r}{r+\lambda_B}\cdot[e^{(r+\lambda_B)t}-1]^2}. \tag{5.44}$$

Assuming that $t > 0$, we are looking for the extremum of this function. Equation 5.44 will equal to zero when:

$$\xi(e^{(r+\lambda_B)t}-1) = (r+\lambda_B)\cdot(\xi t+1). \tag{5.45}$$

To simplify we introduce parametrization:

$$x = (r+\lambda_B)\cdot t, \tag{5.46}$$

therefore, the equation becomes:

$$e^x - x = A + 1, \tag{5.47}$$

where $A = \frac{r+\lambda_B}{\xi}$. Because Equation 5.47 does not have an analytical solution, we expand the e^x function into a series, applying an approximation to the second term, $e^x \approx 1 + x + \frac{x^2}{2}$ (which is a good approximation for small x). Then the equation simplifies to:

$$1 + x + \frac{x^2}{2} - x = A + 1,$$

and we obtain a physical solution:

$$x = \sqrt{2\cdot A}.$$

Returning to the substitution in Equation 5.46 and substituting the constant A, we obtain:

$$\boxed{t_\xi = \sqrt{\frac{2}{\xi(r+\lambda_B)}}.} \tag{5.48}$$

The Equation 5.48 describes the theoretical, optimal (in terms of the cost defined by Equatrion 5.41) time t_ξ of continous irradiation of sample A to produce sample B at the given ratio of cycle-to-sample costs ξ. In Figure 5.10, the theoretical optimal irradiation times for the given ξ were marked with a dotted line. It can be seen that the applied approximation of the exponential function works well for shorter irradiation times (below 2 days), but for longer times, the theoretical optima are shifted toward larger values.

5.3 Implementation of the optimization algorithm

5.3.1 Representation of the irradiation scheme

One of the most important steps in the optimization process is the determination of a simple transition between the optimized system and the optimization algorithm, in this case, genetic algorithm (GA, Section 1.3.1). In order to optimize the irradiation scheme it was necessary to define its simple representation, which would allow easy processing by the algorithm but also be able to reflect all the possible scenarios.

Definition of the irradiation scheme:

If we analyze P available positions in the core, and irradiation for maximum K cycles, we should use up to $m = K \cdot P$ samples to consider all possible scenarios. Therefore, each irradiation scheme (IS) we can code as:

$$[IS]^{K\times P} = \begin{bmatrix} S_{1,1} & \dots & S_{1,P} \\ \dots & S_{k,p} & \dots \\ S_{K,1} & \dots & S_{K,P} \end{bmatrix} \tag{5.49}$$

where:

$$S_{k,p} \in \{1, 2, ..., K \cdot P\}, \tag{5.50}$$

and $S_{k,p}$ denotes number of the sample placed for cycle k at position p. In this way, it is very simple to represent all the possible irradiation schemes. For example, if we consider an irradiation scheme with 4 available positions and a maximum of 2 cycles of irradiation. The irradiation scheme will be represented by a 2×4 matrix. For example, a matrix:

$$[IS]_1^{2\times 4} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} \tag{5.51}$$

will describe an irradiation scheme for which we place one sample in each position and irradiate

them for two cycles, so we use four out of eight available samples. On the other hand, a matrix:

$$[IS]_2^{2\times 4} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \tag{5.52}$$

will represent a scheme for which we are utilising all eight available samples, and irradiate each of them for one cycle. And a matrix:

$$[IS]_3^{2\times 4} = \begin{bmatrix} 5 & 1 & 2 & 3 \\ 5 & 2 & 1 & 4 \end{bmatrix} \tag{5.53}$$

describes a mixed strategy, i.e. sample no 5 stays at the same position for two cycles, samples no 1 and no 2 are swapped between the cycles, and in sample no 3 is exchanged with a new sample no 4.

In this way, we are able to easily represent the irradiation schemes, that takes into account irradiation for different number of cycles, shuffeling and even taking a sample outside the core for a few cycles and irradiating it again. If assumed all the $K \cdot P$ samples can be placed in the core we are able to consider all the possible irradiation schemes.

5.3.2 Optimization framework

To conduct the analysis of the irradiation scheme optimization, a framework depicted in the diagram in Figure 5.11 was used, and it combined several codes and measurements. The utilized optimization program consists of a genetic algorithm coupled with the FISPACT-II package. The algorithm searches for optimal irradiation schemes whereas FISPACT-II assesses the solutions by running the activation calculations. To run the FISPACT-II calculations, neutron flux densities and neutron energy spectra had to be determined at selected locations in the core. The parameters were determined using both, simulations and measurements. First, the Serpent2 Monte Carlo code was used to simulate reaction rate distribution for Co59(n, gamma)Co60 and neutron energy spectra. The serpent2 model was prepared using output data (material compositions) from the REBUS depletion code, that is used to perform real-time calculations at the MARIA reactor facility. The reaction rates, calculated with Serpent2, were compared with real measurements of Al-Co activation foils placed in selected locations in the core and used

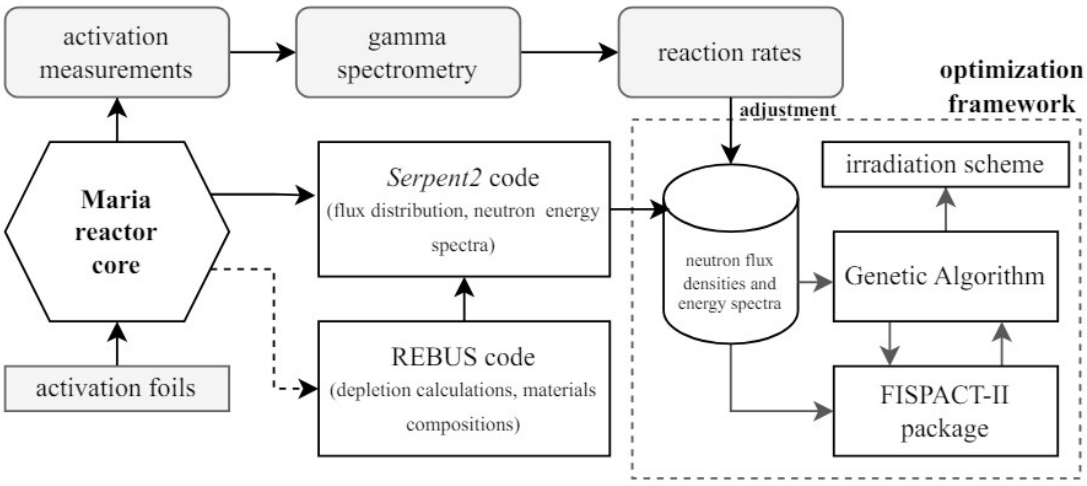


Figure 5.11: Irradiation scheme optimization framework.

to correct the simulated values. After these several steps, the flux parameters (densities and energy spectra) as close as possible to the actual were obtained at the locations considered for the production of Mo-99 and the optimization process. Further sections provide a more detailed description of the subsequent steps of the analysis.

5.3.3 Serpent2 numerical model of the MARIA reactor

Since the reactor cycle of the MARIA reactor is short, typically lasting about 5 to 10 days, the matrix as well as the arrangement of fuel elements systematically change. Consequently, the numerical model also requires updates, both in terms of geometry (new arrangement of elements in the core) and materials compositions, which have changed over the last cycle. At the MARIA reactor research facility, real-time cycle calculations are conducted using the deterministic code REBUS. The results of these calculations (calculated nuclide concentrations) are used to update Monte Carlo models (MCNP, Serpent2), which are mainly for steady-state calculations (flux distributions, energy spectra, rod weights, safety coefficients).

To simplify the process of updating the Monte Carlo model, a framework was developed to automatically generate the input file for the Serpent2 code, based solely o two *.json* files: one file containing the current configuration and one file containing the current nuclide concentrations, generated based on calculations from the REBUS code. The gray region represents the main part of the program, supported by the technological file `core_translator.py`, which contains definitions of cells in the model, transpositions of elements, and calculations of positions in the reactor grid into spatial variables.

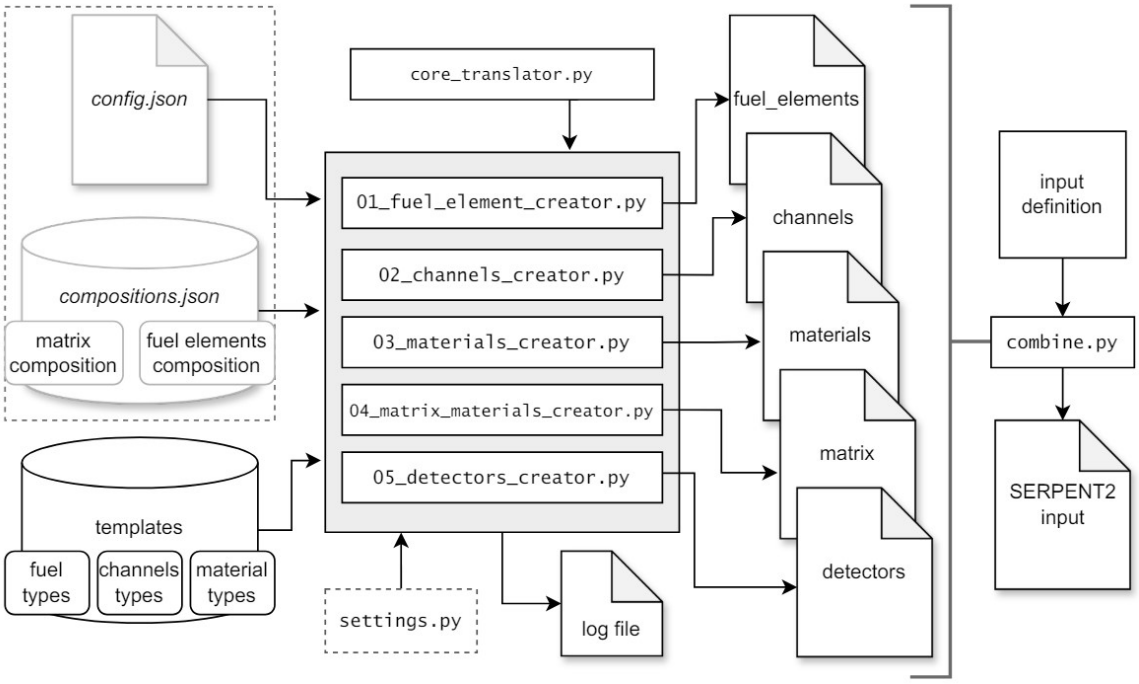
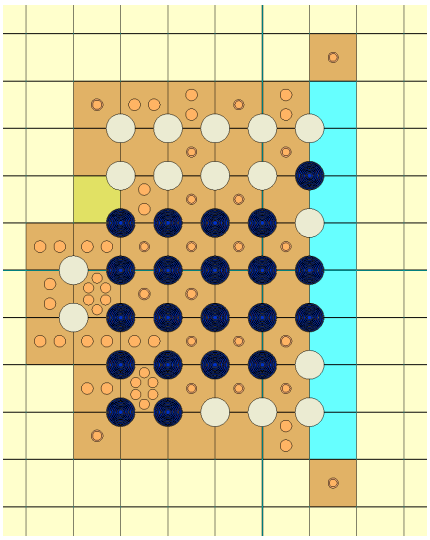


Figure 5.12: The block diagram of the framework for generating input of the MARIA reactor core model for the Serpent code. The dotted region represents the *.json* batch files containing the current configuration and material compositions, defined in `settings.py`. The gray region represents the main part of the program, supported by the technological file `core_translator.py`.

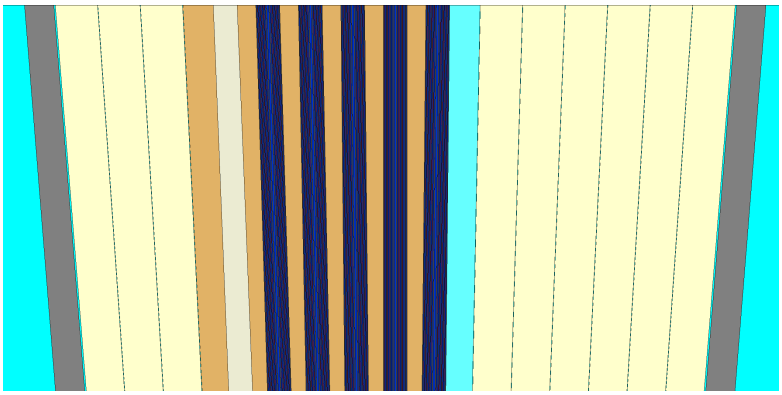
Using the described framework, an input model of the MARIA reactor core was prepared, representing the configuration from the period when activation measurements were taken, i.e., from February 2022 (see the configuration in Appendix B). Figure 5.13 shows a top view and side view of the model, with visible fuel elements, isotope channels, and beryllium blocks.

5.3.4 Theoretical neutron energy spectra and reaction rates

To calculate the flux parameters at selected locations in the core, the Monte Carlo code Serpent2 was used. The study considered 10 locations: *e-7, h-4, g-4, e-8, f-9, i-9, j-5, j-6, j-7*, and *j-8* (see configuration in Appendix B). These locations were chosen for three reasons: firstly, due to the design allowing for the placement of systems for the production of Mo-99, secondly, these are possible locations closest to the measurement points, hence the flux parameters are quite well known, and thirdly, these locations are as close to the center of the core as possible, ensuring the highest flux density values, and in practice, the highest possible reaction rates.



(a) top view



(b) side view

Figure 5.13: Serpent2 model of the MARIA reactor core.

Detector definition in Serpent2

In the Serpent2 code, values of interest can be calculated using detectors, which record the response generally defined as [29]:

$$R = \frac{1}{V} \int_V \int_{E_i}^{E_{i+1}} f(r,E)\phi(r,E)\, d^3r\, dE, \tag{5.54}$$

where V is the volume of analysed region, $f(r,E)$ is the response function, $\phi(r,E)$ is the flux for particles with energy between E_i and E_{i+1}. Depending on the quantity of interest, the response function can take various forms. In the simplest case of measuring the flux density, $f(r,E) = 1$. When determining the reaction rate, it corresponds to the cross section for a given reaction. Serpent2 uses the collision flux estimator (CFE) for estimating particle flux. It is based on the sampling of the ratio of detector response to the total reaction cross-section Σ of the given

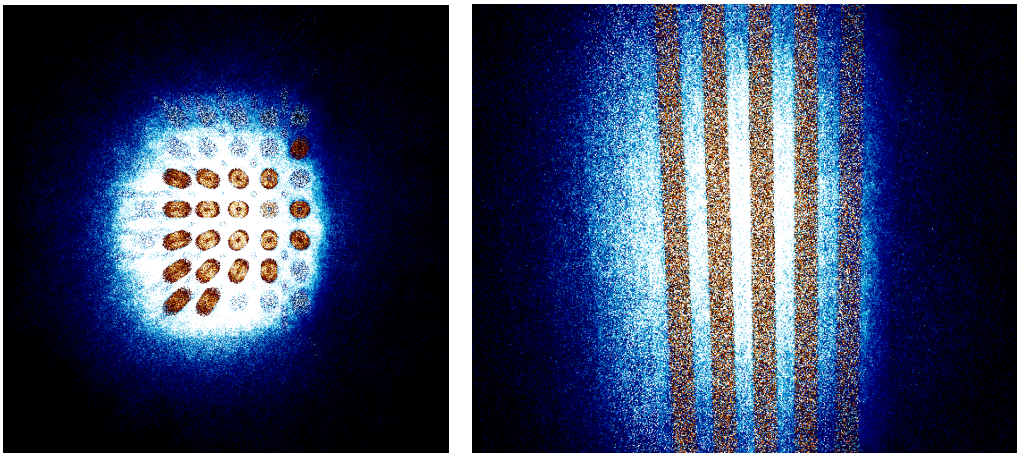
material:

$$s_i = \frac{f(\mathbf{r},E)}{\Sigma(\mathbf{r},E)}. \tag{5.55}$$

As the total number of collisions exceeds the number of occurrences of the reaction of interest, this method provides better statistics. The detector response is computed as a sum over all collisions:

$$x_n = \sum_i s_i. \tag{5.56}$$

In this way, theoretical values of neutron flux density and energy spectra were calculated for the selected location in the core, for the CCFE-709 [175] group structure, which are compatible with the FISPACT-II code. Figure 5.14 presents the visualization of the thermal neutrons for 10^5 histories. Figure 5.15 presents calculated reaction rates and Figure 5.16 the obtained neutron energy spectra (based on 10^9 neutron histories) at the considered locations.



(a) top view. (b) side view

Figure 5.14: Serpent2, visualization of the thermal neutron flux in the MARIA reactor core.

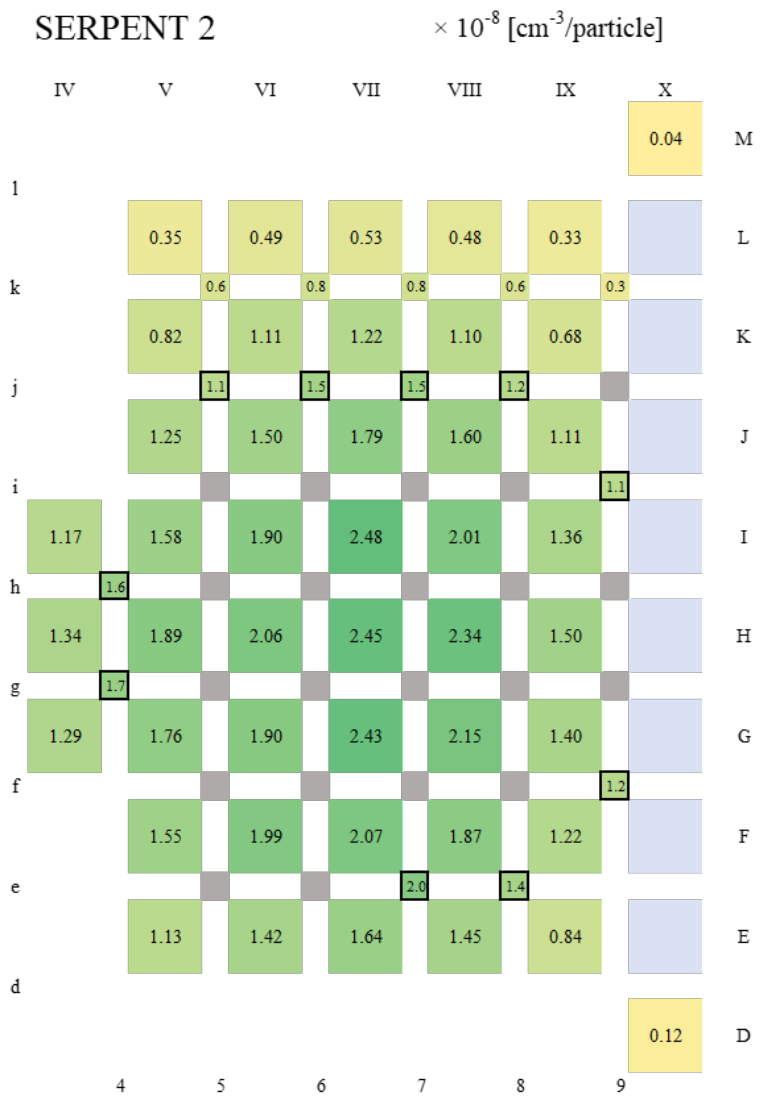


Figure 5.15: Calculated reaction rates, per cm^3, per particle history, registered in the beryllium matrix of the MARIA reactor core. Considered positions for Mo-99 production are marked by black frames. Statistical uncertainty (standard deviation) for all the values is below 1%.

5.3.5 Determination of the thermal neutron flux density

The MARIA reactor undergoes periodic activation measurements to determine the flux distribution and validate internal models. To validate the core model and determine the correction coefficients, the reaction rates determined by activation methods are compared with those obtained from simulations (see Section 5.3.4). In this section, we briefly discuss how the model was validated and how correction coefficients for the neutron flux density values in selected core locations were determined, based on activation measurements conducted in February 2022.

To map the distribution of neutron flux in the core of the MARIA reactor, activation measurements are used, employing activation foils, including an aluminum-cobalt alloy (Al-Co). Cobalt contains the isotope Co-59, activated by Co-59(n, gamma)Co-60 reaction, followed

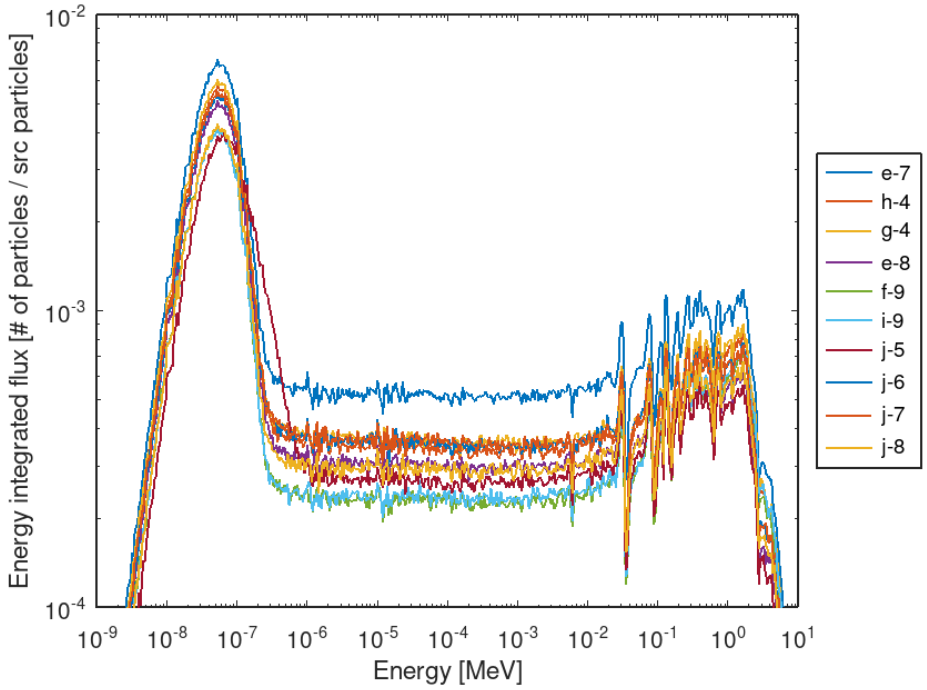


Figure 5.16: Energy spectra of neutrons at various locations in the core of the MARIA reactor, determined using the Serpent2 code.

by measurements of sample activity, allow for the determination of the flux density causing activation (see Section 1.2.3). Of course, there are measurement points in the reactor core (fission chambers and self-powered neutron detectors), but they can only be placed in a few selected locations within the core and serve as a reference and provide ongoing control measurements.

Since the Serpent2 code is capable of simulating steady states, that is, analyzing selected parameters (such as flux distribution and safety coefficients) based on a given number of particle histories, which correspond to a period of time that cannot be determined solely based on Monte Carlo simulations. Then it is not possible to directly correlate the obtained results with time-based values. To obtain coefficients that allow for the transition from model values to actual values (e.g., flux density values or reaction rates) and for model validation and corrections, results from actual measurements are used.

Determining the reaction rate based on the measured activities

Activation foils irradiated at selected locations within the reactor core are measured using a germanium gamma spectrometer to determine their activity. By knowing the irradiation time, cooling time (the interval between irradiation and activity measurement), decay constant, and mass (or number of nuclides in the detector), we can determine the reaction rates. Assum-

ing similar effective cross-sections (see Section 1.2.3), these rates will be proportional to the thermal flux density.

Table 5.3: Results of the activation measurements for the central segment of the core and reaction rates (RR) calculated using Equation 1.18.

Position	A [Bq]	$u(A)$[1]	Irradiation Time [s]	RR measured [s^{-1}]	$u(RR)$[2]
F-V	$8.98 \cdot 10^3$	1.1%	$2.59 \cdot 10^5$	$4.29 \cdot 10^{-9}$	1.2%
G-V	$1.19 \cdot 10^4$	0.9%	$2.59 \cdot 10^5$	$5.69 \cdot 10^{-9}$	1.0%
G-VI	$1.41 \cdot 10^4$	0.8%	$2.59 \cdot 10^5$	$6.74 \cdot 10^{-9}$	1.0%
H-V	$1.17 \cdot 10^4$	0.9%	$2.59 \cdot 10^5$	$5.60 \cdot 10^{-9}$	1.1%
g-4	$2.89 \cdot 10^4$	0.6%	$6.77 \cdot 10^5$	$5.29 \cdot 10^{-9}$	0.8%
h-4	$2.42 \cdot 10^4$	0.6%	$6.77 \cdot 10^5$	$4.43 \cdot 10^{-9}$	0.8%
E-VII	$4.05 \cdot 10^3$	1.6%	$1.08 \cdot 10^5$	$4.65 \cdot 10^{-9}$	1.7%
H-IX	$3.98 \cdot 10^3$	1.6%	$1.08 \cdot 10^5$	$4.57 \cdot 10^{-9}$	1.7%

[1]statistical uncertainty $u(N) \approx \sqrt{N}$, [2]calculated with Equation 1.19.

Then, the relative reaction rates were calculated in comparison to the reaction rates at two locations within the core where actual measurements of thermal neutron flux density were performed, namely at positions H-IX and E-VII:

$$RR_{rel,p}^{H-IX}(x,y) = \frac{RR_p(x,y)}{RR_p(H-IX)}, \qquad RR_{rel,p}^{E-VII}(x,y) = \frac{RR_p(x,y)}{RR_p(E-VII)} \tag{5.57}$$

And $p \in \{s,m\}$ where s means simulation and m - measurement. Under ideal conditions, these values would lead to equal flux densities. However, in reality, due to measurement uncertainties, different spectra, and effective cross-sections together with statistical variations, the relative distributions of reaction rates show slight differences from one another. Then, the measured relative distributions of reaction rates were compared with the relative distributions of reaction rates determined using the Serpent2 model, determining the correction factors for the thermal flux density in the locations where the measurements were conducted (see Table 5.4). Three types of correction factors (measured to simulated) were calculated: the correction factor $c_{m/s}^{H-IX}$ for the distribution relative to position H-IX, the correction factor $c_{m/s}^{E-VII}$ relative to position E-VII, and their average value $\bar{c}_{m/s}$.

It is important to note that the reaction rates determined through measurements and those obtained from Monte Carlo simulations have different units. In activation detectors, the reaction

rate is given as the number of reactions per nucleus per second. In calculations performed with the Serpent2 code, this value is time-independent and is expressed as the number of reactions per cubic centimeter per neutron history. The difference arises because, in the Serpent model, the actual detectors are not simulated. Instead, the reaction rates are calculated as the integral of the cross-section and neutron flux in a given location per cubic centimeter. In the simulation, there is no time dependence; only the number of simulated neutron histories is considered. However, in real detectors, the density of nuclei is constant, and the selected number of histories corresponds to a specific period of time. Thus, despite having different units, these values are simply scaled differently, and their relative distributions will correspond to each other. Table 5.4 presents the distribution of relative reaction rates for both, measurements and Serpent2 model. Analyzing the results, we observe that the distribution determined using the model does not significantly deviate from the measured reaction rate distribution. The correction factors generally differ from unity by no more than about 0.1. It can be noted that the model tends to underestimate the relative reaction rate in the central part of the core and overestimate it towards the edges of the core.

Table 5.4: Comparison of the relative reaction rates measured in the core and those determined using the Serpent2 model.

Position	RR measured (m)			RR Serpent2 (c)			Correction coefficients		
(x,y)	RR_m [s^{-1}]	$RR^{H-IX}_{m,\,rel}$	$RR^{E-VII}_{m,\,rel}$	RR_c [cm^{-3}]	$RR^{H-IX}_{s,\,rel}$	$RR^{E-VII}_{s,rel}$	$c^{H-IX}_{m/s}$	$c^{E-VII}_{m/s}$	$\bar{c}_{m/s}$
F-V	$4.29 \cdot 10^{-9}$	0.94	0.92	$1.55 \cdot 10^{-8}$	1.03	0.95	0.91	0.97	0.94
G-V	$5.70 \cdot 10^{-9}$	1.25	1.22	$1.76 \cdot 10^{-8}$	1.17	1.07	1.07	1.14	1.10
G-VI	$6.74 \cdot 10^{-9}$	1.47	1.45	$1.90 \cdot 10^{-8}$	1.26	1.16	1.17	1.25	1.21
H-V	$5.61 \cdot 10^{-9}$	1.23	1.21	$1.89 \cdot 10^{-8}$	1.26	1.16	0.97	1.04	1.01
g-4	$5.30 \cdot 10^{-9}$	1.16	1.14	$1.67 \cdot 10^{-8}$	1.11	1.02	1.04	1.11	1.08
h-4	$4.44 \cdot 10^{-9}$	0.97	0.95	$1.59 \cdot 10^{-8}$	1.06	0.97	0.92	0.98	0.95
E-VII	$4.65 \cdot 10^{-9}$	1.02	1.00	$1.64 \cdot 10^{-8}$	1.09	1.00	0.93	1.00	0.97
H-IX	$4.57 \cdot 10^{-9}$	1.00	0.98	$1.50 \cdot 10^{-8}$	1.00	0.92	1.00	1.07	1.04

The uncertainties for the measured values in the table were assessed to be <1% (Equation 1.19) and statistical uncertainty (standard deviation) of Serpent2-based values were <1%.

Determination of the neutron flux density

To calculate the values of the thermal neutron flux density at the selected locations, the following formula was applied:

$$\phi_c(x,y) = \frac{1}{2}\left[\phi_c^{H-IX}(x,y) + \phi_c^{E-VII}(x,y)\right] \tag{5.58}$$

Where $\phi_c^{H-IX}(x,y)$ and $\phi_c^{E-VII}(x,y)$ are the fluxes calculated using simulated relative reaction rates distributions $RR_{rel,s}^{H-IX}(x,y)$, $RR_{rel,s}^{E-VII}(x,y)$ and the reference measurements: $\phi_m(H\text{-}IX)$, $\phi_m(E\text{-}VII)$ measured at positions H-IX and E-VII:

$$\phi_c^{H-IX}(x,y) = RR_{rel,s}^{H-IX}(x,y)\cdot\phi_m(H\text{-}IX), \qquad \phi_c^{E-VII}(x,y) = RR_{rel,s}^{E-VII}(x,y)\cdot\phi_m(E\text{-}VII) \tag{5.59}$$

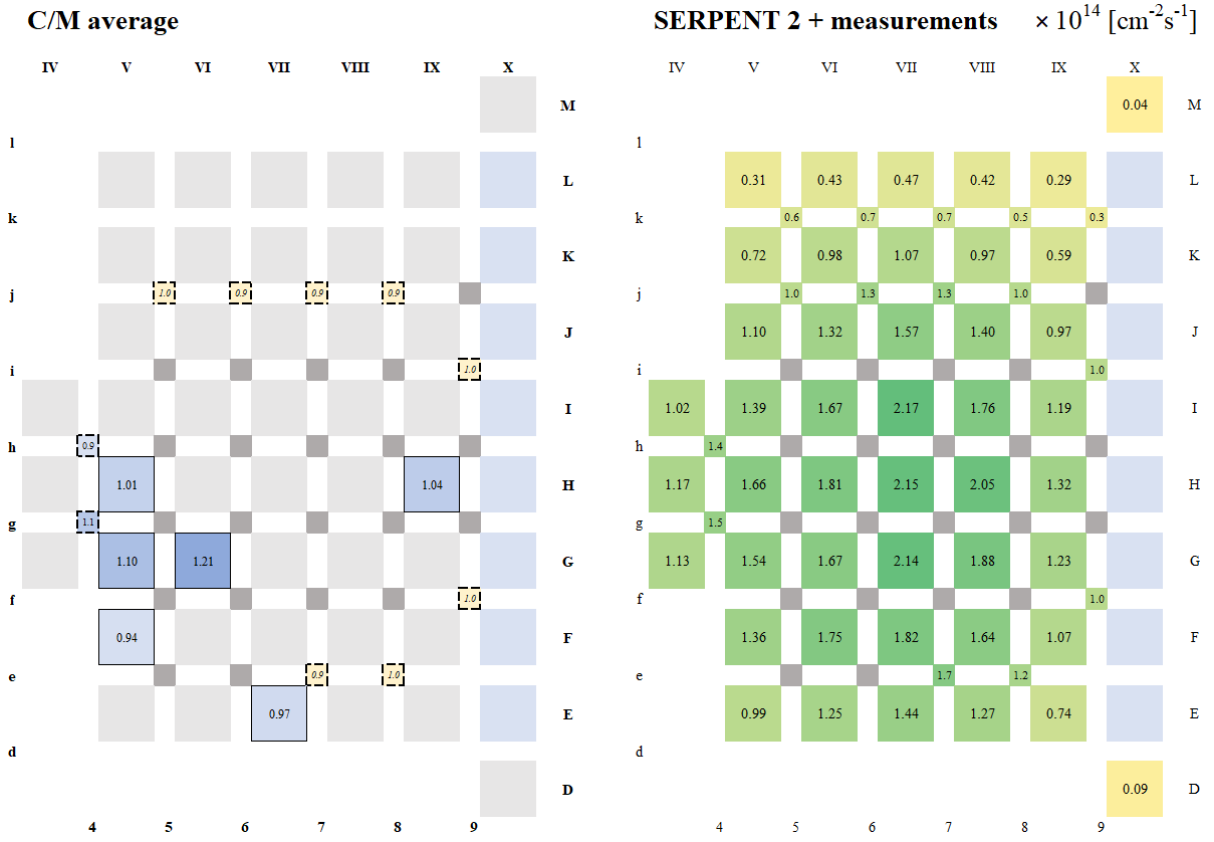


Figure 5.17: The neutron flux density distribution determined using the formula in Equation 5.58 (right side) and the average correction factors for the measurement locations (left side). Yellow dashed squares in the left figure indicate the specific correction factors for the locations considered for Mo-99 production.

Values of the measured reference thermal fluxes densities, provided by the MARIA reactor facility are:

$$\phi_m(H\text{-}IX) = 1.37(0.10)\cdot 10^{14} cm^{-2}s^{-1}, \qquad \phi_m(E\text{-}VII) = 1.39(0.10)\cdot 10^{14} cm^{-2}s^{-1}$$

Table 5.6: Calculated flux ϕ_c and corrected flux $\hat{\phi}_c$ at various positions, along with correction factors $\hat{C}$.

Position (x,y)	$\phi_c(x,y) \cdot 10^{14}$ [cm^{-2}s^{-1}]	$\bar{c}(x,y)$	$\hat{\phi}_c(x,y) \cdot 10^{14}$ [cm^{-2}s^{-1}]
e-7	1.72(0.17)	0.9	1.55(0.16)
h-4	1.39(0.14)	0.9	1.30(0.13)
g-4	1.47(0.15)	1.1	1.55(0.16)
e-8	1.24(0.12)	1.0	1.24(0.12)
f-9	1.01(0.10)	1.0	1.01(0.10)
i-9	0.99(0.10)	1.0	0.99(0.10)
j-5	0.97(0.10)	1.0	0.97(0.10)
j-6	1.31(0.13)	0.9	1.18(0.12)
j-7	1.34(0.13)	0.9	1.21(0.12)
j-8	1.04(0.10)	0.9	0.93(0.09)

The main source of uncertainty for the above values is the measurement of the flux, which was determined to have an uncertainty of 8%, hence the uncertainties of the determined values were set at the level of 10%.

Additionally, the positions *e-7, h-4, g-4, e-8, f-9, i-9, j-5, j-6, j-7*, and *j-8*, considered in this study, the correction factors for the flux density values were determined based on the neighboring correction factors $\bar{c}(x,y)$ for the locations where measurements were performed (see yellow squares in left scheme in Figure 5.17):

$$\hat{\phi}_c(x,y) = \phi_c(x,y) \cdot \bar{c}(x,y) \tag{5.60}$$

For positions *g-4* and *h-4*, the values of $\bar{c}(x,y)$ were calculated based on the measurements. For other locations, it was determined expertly, based on the correction factors and the flux density values in the vicinity of the considered position. The final values of the neutron flux density are presented in Table 5.6.

The uncertainties of the values determined in Table 5.6 are primarily based on the uncertainty of measuring the reference flux values, which was about 8%. Due to the fact that RR are not independent variables, and their strong mutual correlation as well as correlation with the flux density value, the final determination of uncertainties is not a straightforward task. In this study, due to the applied method, the uncertainty in measuring reference values, and the estimated correction coefficients used, the final uncertainty of the flux was estimated at 10%.

Despite the uncertainty values being estimated and seeming significant in the context of optimization, the uncertainty of the flux density value does not play the most crucial role here. Since the optimization is carried out with respect to a reference scheme (one cycle), the shape of the flux distribution is more important in this study, and it has been determined with relatively good accuracy (see Table 5.4). Finally, while the flux value is uncertain, it only scales the problem but should not affect the optimization outcome.

5.4 Results

5.4.1 Estimation of the ratio of cycle costs to target material costs

Determining the operating costs of a reactor core is an extremely challenging task and constitutes a separate research topic that has been addressed numerous times [176–179], particularly in terms of nuclear power plant insurance and securing sufficient funds to ensure financial liquidity both during the reactor's operation and after the facility's decommissioning and waste disposal. A variety of models are employed, including detailed data analyses and probabilistic assessments. In the case of research reactors, the situation is even more complex to analyze than for commercial reactors. This complexity arises from the fact that research reactors often operate under different economic conditions, which can involve direct governmental support [178, 180], impacting the financial models used to calculate their operational costs.

It is easier to determine the costs of the sample, as they are usually standardized. However, the costs can also vary depending on contracts, material type and the nature of the facility. In this study, we are only trying to determine the orders of magnitude of these costs to get a rough approximation of the value of the ratio ξ in Equation 5.37 that we might be dealing with.

Irradiation costs

In this study, we assume that in the MARIA reactor, there is a certain volume in which target material can be placed for the production of radioisotopes, and the irradiation costs of a given sample are proportional to the portion of this volume occupied. In this very simplified analysis, aimed only at determining the order of magnitude of the value ξ, we take into account:

- Reactor MARIA maintenance costs of 3.5 million EUR annually [180],
- Fuel costs, about 20-25 fuel elements replaced every three years [180], which equates to approximately 8 fuel elements annually, 100,000 EUR each [178], which results in around 0.8 million EUR annually,

- Personnel costs (assuming about 100 people, an average of 1200 EUR of costs per person per month), approximately 1.5 million EUR annually.

The above calculations amount to approximately 5.8 million EUR annually for the operational costs of the MARIA reactor core. The operation time projected for MARIA reactor in 2024 is 4200 hours [181]. Assuming 5 days per cycle, which is 120 hours per cycle, resulting in 35 cycles per year, and around 170,000 EUR per cycle. Of course, these calculations are very simplified and do not include all costs, such as the disposal of spent fuel and its storage, or the decommissioning of the power plant. Nevertheless, it shows the order of magnitude of expenses associated with the cycle.

To calculate the irradiation cost per cm^3 of a sample, we determine the volume in the MARIA reactor that can be used for irradiating samples and producing radioisotopes. For simplicity, we assume that there are about 5 channels with a diameter of 60 mm and about 15 channels with a diameter of 30 mm in the beryllium matrix of the core, additionally, there are 15 pegs in the matrix that can also be replaced with an irradiation system and used for production (see configuration in Appendix B). The active part of the core is approximately 100 cm in length. This gives a total of about $1.23 \cdot 10^5$ cm^3 of available volume. However, for technical reasons (appropriate range of reactivity, no samples at the very ends of the active part of the core, volume of the container, and supporting structures), not all of this volume can be effectively used for production. The effective volume is very difficult to determine, however, in this very simple estimate, we assume that the target materials can occupy up to 10% of the available volume, which results in $1.23 \cdot 10^4$ cm^3 (the order of magnitude is in line with the total active volume of the main part of the irradiation system of 100 containers $\times$ 50 cm^3, there are also other available parts, with less precisely defined volume). Finally, the irradiation costs would be equal to around **13.5 EUR/cm^3** per cycle.

Target material costs

For the production of Mo99 in the MARIA reactor, uranium targets are used. Analyzing the price list prepared by the *New Brunswick Laboratory*, material containing 5.4g (or approximately 0.49cm^3) of uranium dioxide enriched to 4% (certified reference material ID C125A) costs approximately 630 EUR [182]. It results in **1283 EUR/cm^3** of target material costs.

Estimated ratio of cycle costs to target material costs

In this simplified analysis, the irradiation costs of the sample were found to be: $C_1 \approx 13.5$ EUR/cm^3 per cycle, and the costs associated with purchasing the sample were: $C_1 \approx 1283$ EUR/cm^3. This gives, according to Equation 5.37, a value of:

$$\bar{\xi} = \frac{C_1}{C_2} \approx 0.01. \tag{5.61}$$

The above analysis is very approximate and is not intended for cost analysis, and it is burdened with a great deal of uncertainty, both from the approximations used and parameters and costs that were not included. Therefore, in the further part of the study, within the framework of a sensitivity study, values 10 times smaller and 10 greater than the calculated value will be used.

It's worth noting that the calculated value is about 20 times smaller than the threshold value $\xi_2 \approx 0.2$ calculated in Section 5.2.6 (Equation 5.39). This means that despite the significant uncertainty in determining the ratio of irradiation costs to sample costs, the actual value can be below the threshold value, which potentially indicates the possibility of profit from irradiating samples for more than one cycle.

5.4.2 Reference cases

In this section, the results of calculations for reference cases will be presented, i.e., for 1, 2, 3, and 4 cycles, for each sample remaining in the same position through successive cycles, and with full shuffling of samples between cycles.

One cycle

Since all the samples placed in the selected locations in the core have the same mass, it does not matter how we place them in the core. Using the FISPACT-II package, calculations were performed for the irradiation of 10 samples of the target material for one cycle. The average production yield (i.e., the sum of produced Mo99 nuclides per total number of used U235 nuclides) is:

$$\eta(1) \approx 5.35 \cdot 10^{-4}.$$

The efficiency value $\eta(1)$ will further serve as the reference value to which we will compare other loading schemes and with which we will calculate the relative value of irradiation costs $\hat{C_{rel}}$ (Equation 5.38).

Multi-cycle

For scenarios with 2, 3, and 4 irradiation cycles, we consider two cases - without shuffling and with shuffling. In the first case, we assume that we place the target material in each of the considered positions and irradiate it for the given number of cycles, with no shuffling between cycles (marked as *'ns'*). According to the nomenclature described in Section 5.3.1, for example, for 3 cycles, it could be:

$$[IS]_{ns}^{3\times 10} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}.$$

The next case considered was multiple cycles but with shuffling applied (marked as *'ws'*). In this reference case, random shuffling is used after each cycle, with the assumption that no sample can repeat its location in all the cycles. For an example of three cycles it could be:

$$[IS]_{ws}^{3\times 10} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 9 & 10 & 4 & 3 & 7 & 8 & 5 & 6 & 1 & 2 \\ 10 & 8 & 7 & 6 & 1 & 5 & 9 & 2 & 3 & 4 \end{bmatrix}.$$

Table 5.7 presents the results of efficiency calculations and relative irradiation costs for the reference cases considered for 1 to 4 cycles, 10 samples, and nominal value of $\bar{\xi} = 0.01$ as well as the sensitivity analyses with $0.1 \cdot \bar{\xi} = 0.001$ and $10 \cdot \bar{\xi} = 0.1$.

The summary shows that for a greater number of cycles, a higher efficiency (i.e., the number of Mo-99 nuclei produced per number of used U-235 nuclei) can be achieved than for a single cycle. Using two cycles leads to higher efficiency, even without shuffling samples.

The highest efficiency for irradiation without shuffling (*'ns'*) is obtained for 3 cycles. This means that for a greater number of cycles, the concentration of the nuclide Mo-99 approaches the saturation level and further irradiation makes no sense, while the final efficiency may decrease due to the decay of the produced nuclei during breaks between cycles. This corresponds to the predictions described in Table 5.2.

The cases with shuffling (*'ws'*) were generated randomly, with the assumption of full

Table 5.7: The results of calculations for reference cases, 1-4 cycles, 10 samples.

				$\hat{C_{rel}}[\%]$ (Equation 5.38)		
Cycles	Shuffling	Scheme	η	$\xi = 0.001$	$\xi = 0.01$	$\xi = 0.1$
1	-	$[IS]^{1\times10}$	5.35×10^{-4}	100.0	100.0	100.00
2	no	$[IS]^{2\times10}_{ns}$	6.17×10^{-4}	86.8	87.6	94.6
2	yes	$[IS]^{2\times10}_{ws}$	6.19×10^{-4}	86.4	87.2	94.2
3	no	$[IS]^{3\times10}_{ns}$	6.21×10^{-4}	86.3	87.8	101.8
3	yes	$[IS]^{3\times10}_{ws}$	6.21×10^{-4}	86.3	87.9	101.8
4	no	$[IS]^{4\times10}_{ns}$	6.12×10^{-4}	87.6	90.0	111.2
4	yes	$[IS]^{4\times10}_{ws}$	6.15×10^{-4}	87.2	89.5	110.6

shuffling, meaning that no sample could be found in the same location twice. This random shuffling does not guarantee the certainty of the best possible solution but allows for a preliminary comparison of solutions with and without shuffling. In the cases considered, there is a negligible impact of swapping samples between subsequent cycles. For 4 cycles, it increases the efficiency from $6.12 \cdot 10^{-4}$ to $6.15 \cdot 10^{-4}$. In other cases, it practically does not affect efficiency.

5.4.3 Optimization

In the previous Section, we considered only 10 samples and extreme cases, i.e., leaving samples in the same positions through subsequent cycles, or randomly swapping their places between cycles. In this part, we allow the optimization algorithm to select the number of samples, that is to consider all scenarios, including placing a new set of samples with each new cycle as well as the scenario of removing a given sample for a selected number of cycles and then reinserting it. In this way, for a scenario with two cycles, the GA could consider using up to 2×10 samples, in the scenario with three cycles 3×10 samples and respectively 4×10 samples for the scenario with four cycles. Theoretical considerations in Section 5.2.4 preliminarily showed that with the flux values we are dealing with in the core of the MARIA reactor, this strategy should not lead to better efficiency. However, there is no basis to reject the possibility of exisiting a certain mixed strategy of irradiation (e.g., swapping part of the samples, removing them for selected cycles and/or reinserting them after some time) which could potentially lead to better efficiency.

Four cycles

In this scenario, it was considered that the irradiation of samples would occur over four consecutive cycles. The algorithm had complete flexibility in selecting the number of samples. Optimization was done using a population of 50 specimens, $p_m = 0.1$ and $p_c = 0.9$. The process starts with the random selection of 50 individuals for the initial population, i.e., 50 matrices of $[IS]_{ws}^{4\times 10}$ containing numbers from 1 to 40.

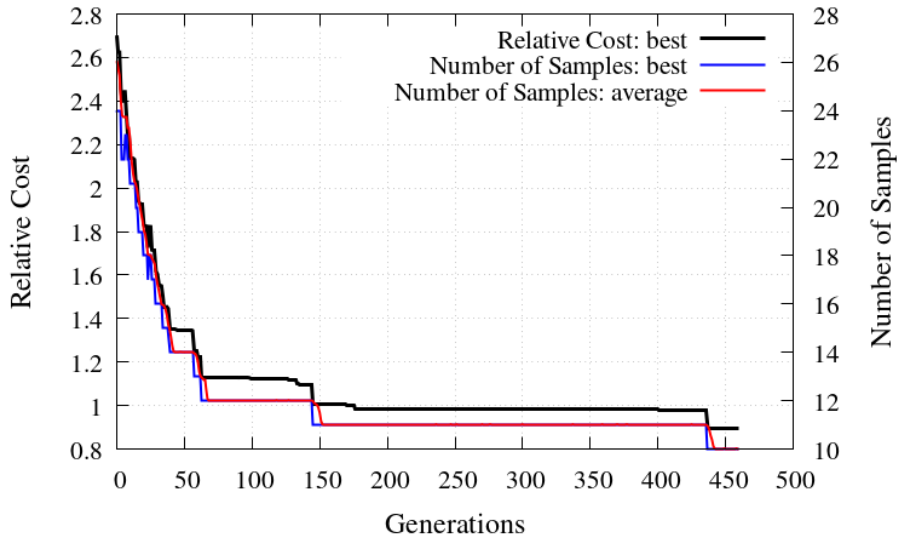


Figure 5.18: Evolution curves for the 4-cycles scenario. Relative production cost and the number of samples for the best individual in the population, and the average number of samples in the population. Genetic Algorithm, a population of 50 individuals, $p_m = 0.1$, $p_c = 0.9$.

Figure 5.18 shows the evolution of the relative cost and the number of samples used for the best individual (i.e., the one with the lowest relative cost). Additionally, the graph also presents the average number of samples in the population. The graph shows that the algorithm very quickly removes additional samples and promotes solutions in which the same samples are irradiated over several cycles rather than introducing new ones. It is evident that the shape of the relative cost curve matches the shape of the curve describing the number of samples. This indicates that the improvement in process efficiency is mainly due to reducing the number of samples rather than their optimal shuffling in the core. After 439 generations, the algorithm reduces to the minimum number of samples (i.e., 10), and the relative cost practically no longer decreases, with differences below tenths of a percent (see Table 5.8).

Table 5.8: Evolution of results for the 4-cycles scenario, for a random population (generation 1), the optimization midpoint (250), the point of minimization of the number of samples (439), and last generation (450).

			$\hat{C_{rel}}[\%]$ (Equation 5.38)		
Generation	Samples	η	$\xi = 0.001$	$\xi = 0.01$	$\xi = 0.1$
0	24	2.04×10^{-4}	263.2	270.2	334.9
250	11	5.60×10^{-4}	95.8	98.4	121.6
439	10	6.16×10^{-4}	87.0	89.4	110.5
450	10	6.16×10^{-4}	87.0	89.3	110.4

Table 5.9 presents the best irradiation scheme found by the algorithm after 450 generations. Analyzing the solution, it can be observed that for several locations the algorithm maintains the sample in the same place for two cycles. It was checked (by manually modifying the scheme) that introducing shuffling for these locations slightly worsens the final result (below <0.1%, but to a disadvantage). This shows that it is a deliberate action of the algorithm, likely caused by a greater gain from exchanging samples in another location. It confirms the speculation that the optimal solution may be a non-trivial mix of different irradiation strategies.

Another observation might be that samples are typically moved across two different sides of the core (e.g., from location *e-7* to *g-4* or *g-4* to *j-8*), especially during the first two cycles, likely due to greater differences in flux parameters. Although exchanges of neighboring locations do occur, especially between cycles 2, 3, and 4, such as moving from *j-8* to *j-7* and finally to *j-6*.

Three cycles

Simulations were repeated considering 3 cycles, which means up to 30 samples in the core as part of the search for the scheme $[IS]_{ws}^{3x10}$. Again, a population of 50 individuals was utilized. Due to the smaller space of possible solutions, the optimization was performed for 150 generations. Similarly to the 4-cycle scenario, Figure 5.19 presents the evolution of relative cost and the number of samples for the best individual, as well as the average number of samples in the population.

Table 5.9: Best irradiation scheme for the 4-cycle scenario.

Sample	Cycle 1	Cycle 2	Cycle 3	Cycle 4
1:	e-7	g-4	i-9	i-9
2:	h-4	j-5	g-4	j-7
3:	g-4	j-8	j-7	j-6
4:	e-8	i-9	e-7	g-4
5:	f-9	j-7	j-8	e-7
6:	i-9	f-9	e-8	h-4
7:	j-5	h-4	j-5	j-5
8:	j-6	j-6	h-4	j-8
9:	j-7	e-8	j-6	e-8
10:	j-8	e-7	f-9	f-9

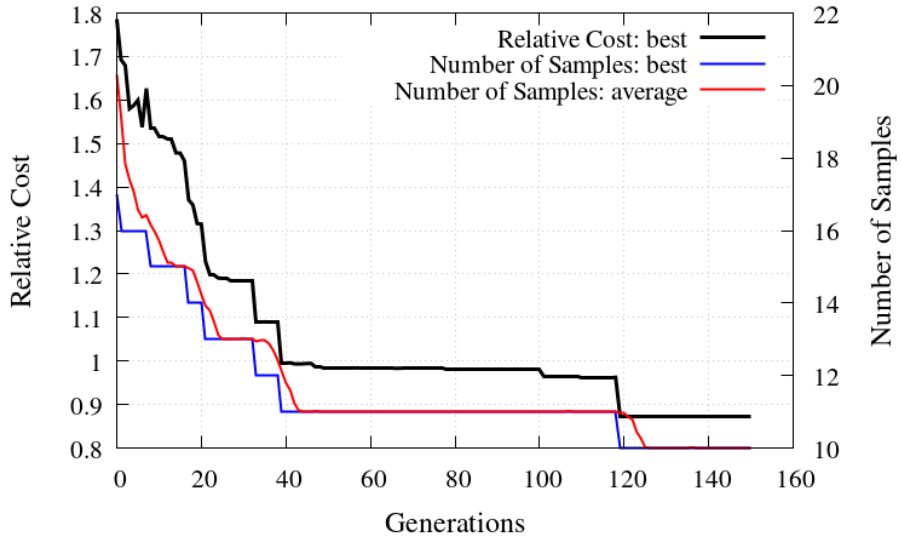


Figure 5.19: Evolution curves for the 3-cycles scenario. Relative production cost and the number of samples for the best individual in the population, and the average number of samples in the population. Genetic Algorithm, a population of 50 individuals, $p_m = 0.1$, $p_c = 0.9$.

Table 5.10 presents selected points of the optimization process. Identically to the 4-cycle scenario, the algorithm eliminates samples, reaching the minimum number of 10, within 119 generations. Further optimization practically does not improve the relative cost of irradiation anymore. Table 5.11 presents the best irradiation scheme proposed by the algorithm. On the scheme, it can also be observed that positions are usually changed to those on the opposite side of the core. Additionally, it can be noted that the algorithm decided to keep sample no. 7 in position *j-5* for the first two cycles. A similar analysis was applied and the position of this sample was randomly changed in the second cycle, resulting in a slightly worse result (<0.1%).

Table 5.10: Evolution of results for the 3-cycles scenario, for a random population (generation 1), the optimization midpoint (80), the point of minimization of the number of samples (119), and last generation (150).

			$\hat{C_{rel}}[\%]$ (Equation 5.38)		
Generation	Samples	η	$\xi = 0.001$	$\xi = 0.01$	$\xi = 0.1$
0	17	3.05×10^{-4}	175.5	178.6	207.0
80	11	5.55×10^{-4}	96.5	98.2	113.8
119	10	6.25×10^{-4}	85.8	87.3	101.2
150	10	6.25×10^{-4}	85.8	87.3	101.1

Table 5.11: Best irradiation scheme for the 3-cycle scenario.

Sample	Cycle 1	Cycle 2	Cycle 3
1:	e-7	j-8	j-5
2:	h-4	i-9	j-7
3:	g-4	j-7	j-6
4:	e-8	f-9	e-7
5:	f-9	j-6	g-4
6:	i-9	e-8	h-4
7:	j-5	j-5	i-9
8:	j-6	g-4	f-9
9:	j-7	h-4	e-8
10:	j-8	e-7	j-8

Two cycles

Similar analyses were conducted for the 2-cycle scenario. In this scenario, the algorithm considered up to 20 samples placed in 10 locations over two cycles, i.e., it searched for the scheme $[IS]_{ws}^{2\times10}$. Figure 5.20 presents the evolution of the relative cost and the number of samples. Due to a much smaller solution space, the algorithm converges much faster. After just 12 generations, it selects the minimum number of samples. Subsequent iterations do not improve the result significantly (see Table 5.12).

Table 5.12: Evolution of results for the 2-cycles scenario, for a random population (generation 1), the optimization midpoint (10), the point of minimization of the number of samples (12), and last generation (30).

			$\hat{C_{rel}}[\%]$ (Equation 5.38)		
Generation	Samples	η	$\xi = 0.001$	$\xi = 0.01$	$\xi = 0.1$
0	13	4.33×10^{-4}	123.7	124.8	134.8
10	11	5.49×10^{-4}	97.6	98.5	106.4
12	10	6.18×10^{-4}	86.7	87.5	94.5
30	10	6.20×10^{-4}	86.4	87.1	94.1

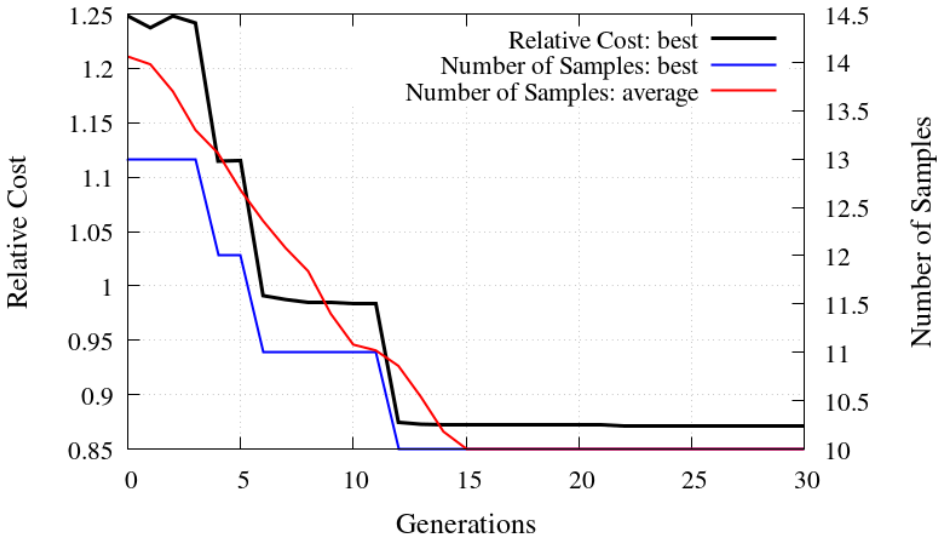


Figure 5.20: Evolution curves for the 2-cycles scenario. Relative production cost and the number of samples for the best individual in the population, and the average number of samples in the population. Genetic Algorithm, a population of 50 individuals, $p_m = 0.1$, $p_c = 0.9$.

Table 5.13 presents the obtained irradiation scheme. In this scheme, it is hard to discern any pattern - it includes both exchanges of neighboring locations (e.g. *g-h, h-4*) and those farther apart (e.g. *j-8, e-8*). However, this scheme does not assume keeping a sample in one location for two cycles. Probably no combination necessitates leaving a sample in the same place and swapping all samples might be the most optimal.

Table 5.13: Best irradiation scheme for the 2-cycle scenario.

Sample	Cycle 1	Cycle 2
1:	e-7	f-9
2:	h-4	j-7
3:	g-4	h-4
4:	e-8	e-7
5:	f-9	g-4
6:	i-9	j-6
7:	j-5	i-9
8:	j-6	j-5
9:	j-7	j-8
10:	j-8	e-8

5.4.4 Summary and conclusions

Table 5.14: Summary of the 4-cycles, 3-cycles, and 2-cycles scenarios compared with the reference calculation of 1-cycle scenario.

			$\hat{C_{rel}}[\%]$ (Equation 5.38)			
Case	Scheme	η	$\xi = 0.001$	$\xi = 0.01$	$\xi = 0.1$	ξ_c
4-cycles	$[IS]_{ws}^{4\times10}$	6.16×10^{-4}	87.0	89.3	110.4	0.05
3-cycles	$[IS]_{ws}^{3\times10}$	6.25×10^{-4}	85.8	87.3	101.1	0.09
2-cycles	$[IS]_{ws}^{2\times10}$	6.20×10^{-4}	86.4	87.1	94.1	0.19
1-cycle	$[IS]^{1\times10}$	5.35×10^{-4}	100.0	100.0	100.0	-

Table 5.14 presents a summary of the results for the optimization of the irradiation scheme, i.e., production efficiency η (Equation 5.1) and the relative production cost $\hat{C_{rel}}$ (Equation 5.38) for scenarios with four, three, and two cycles, compared to reference irradiation through one cycle. Additionally, based on the calculated efficiencies, the marginal cost coefficient values $\xi_c(n)$ were determined, beyond which irradiation through n cycles ceases to be cost-effective. Equation $\xi_c(n)$ was derived by calculating the condition $\hat{C}_{rel} = 1$, and equals to:

$$\xi_c(n) = \frac{\eta(1) - \eta(n)}{(n-1) \cdot \eta(1)}. \tag{5.62}$$

From the results, we can derive several following conclusions:

- The highest production efficiency $\eta = 6.25 \times 10^{-4}$ was obtained for irradiating samples over 3 cycles. Breaks between cycles for a larger number of cycles begin to lower efficiency and cause that irradiation over a greater number of cycles is not optimal.

- For the nominal value of the cost coefficient $\xi = 0.01$, the lowest relative production cost is also obtained for irradiation over 3 cycles - about 86% of the costs of irradiation compared to irradiation through 1 cycle, which is in line with the theoretical predictions presented in Figure 5.9 (Section 5.2.6). However, if cycle costs are higher than determined, irradiation over 2 cycles is more cost-effective. Another policy could be to extend the cycle length. In accordance with Figure 5.10 (curve for $\xi = 0.01$), presented in Section 5.2.6, around 15% lower cost production could be obtained using 8-days cycle (a longer cycle does not improve the efficiency much more).

- The determined marginal value $\xi_c(4) = 0.05$ shows that for irradiation through 4 cycles, the nominal value is only 5 times lower. In the case of 2 and 3 cycles, it is respectively 9 and 19 times more. Assuming the fixed target sample costs, calculated in Section 5.4.1, in practice, this would mean operating costs of the reactor of about 50 million EUR and 100 million EUR annually, respectively. It seems that despite rough calculations, we are far from this limit. It's worth noting that the value of $\xi_c(2) = 0.19$ is close to the theoretical value of $\xi_2 = 0.21$ calculated using Equation 5.39 from Section 5.2.6.

- Comparing the results from Table 5.14 with the reference calculations from Table 5.7, one can conclude that the number of cycles used and shuffling samples is significant, the way of shuffling plays a minor role and improves the efficiency of the process by tenths of a percent, which is completely negligible, especially when compared with the uncertainties of the flux parameters in determined places and the uncertainties of calculations with the FISPACT-II package.

- Overall, from the conducted analyses, it seems that potentially the most cost-effective production might be through irradiation over two cycles, even without applying shuffling, reducing costs by about 13%, or with the application of shuffling, further reducing costs by approximately another 0.5%, which is in line with the theoretical considerations in Section 5.2.5, where is it shown that shuffling leads to the same or better efficiency.

- Increasing the number of cycles to three results just in a slight improvement but brings us closer to the critical value of cost parameters, which could potentially reduce the profitability of the process.

Appendix

A Description of the class `LoadingPatternOptimizer`

This Appendix describes selected technical details of the implemented class for Loading Pattern Optimization.

Methods of the `LoadingPatternOptimizer`

1. `SetParameters(params*)`

   This method sets the most important parameters of the optimization, i.e. fitness function and its parameters, assumed symmetry, optimization method, path to the calculations, and other most important tion (see Table .15). Variables of the method are listed and explained in Table .16. Table .17 presents an example of the FF definition that has to be called in the `SetParameters()` method.

2. `ModelInit(create_directory=[True])`

   This method initializes the model, the FF, optimization method, selection, mutation, etc. The variable `create_directory` (default `true`) defines if the program creates the directory to save the optimization progress and results or not.

3. `PopulationInit()`

   This method creates the population. It can be random (by default) or be created from a provided checkpoint or specimen banks.

4. `CalculationsInit()`

   This method initializes *Copilote* calculations. It copies the calculation plan from the provided project and initializes the models based on the created population.

5. `Run(test=[False], save=[True], verbose=[1])`

   This method runs the calculations. In this method `test` option can be selected (`false` by default) to disable SMART calculations and only test the optimizer, `save` option allows switching off saving the results (useful for tests), `verbose` veriable allows to select the level (0-2) of details printed in the output.

Main file of the `LoadingPatternOptimizer`

```
from LPO import LoadingPatternOptimizer

def main():
    LPO = LoadingPatternOptimizer()

    LPO.SetParameters(
        PARAMS = {'FDH_init':10, 'FDH_max':10, 'ASSBU':100000,
                  'CORBUCYC':15000, 'CORBUCYCJEPP':100, 'CORFUITES_avg':100},
        SYMMETRY = '1/8',
        TITLE = 'PSA+GA_octant_Case1',
        OPTIMIZER = 'PSAGA',
        CXPB = 0.2,
        MUTPB = 1.0,
        T_INIT = 1000,
        TOURNAMENT_SIZE = 20,
        NGEN = 100,
        POP_SIZE = 200,
        PM_PATH = "./Copilote_calculations/master_plan_EPR",
        CALCULATIONS_PATH = "./LPO/Case1",
        TIMEOUT = 15,
        FF = FF,
        IF_CALCULATIONS_FAIL = 'restart',
        POPULATIONS=["./populations_bank/population_1.csv",
                     "./populations_bank/population_2.csv"]
        )

    LPO.ModelInit(create_directory=True)
    LPO.PopulationInit()
    LPO.CalculationsInit()
    LPO.Run(test=False, save=True, verbose=1)
```

Table .15: Main file for the Loading Pattern Optimizer.

Parameters of the `SetParameters()` method

PARAMS = {"PARAMETER_*type*" : *default*, ...}, parameters that are used in the fitness function definition, can be FDH, burnup, and leakage, *type* can be *max, init* or *avg*. The *default* value is important to set as it is used when the calculations fail.
SYMMETRY = core symmetry, `'1/8'` or `'1/4'`
TITLE = title of the optimization
CXPB = crossover probability, default = 0.9
MUTPB= mutation probability, default = 0.1
T_INIT = initial temperature, default = 1000
TOURNAMENT_SIZE = size of the tournament, default = 20
POP_SIZE = size of the population, default = 200
PM_PATH = path to the *Copilote* calculations (the model to be optimized)
CALCULATIONS_PATH = path to the directory where the calculations will be stored
TIMEOUT = max time of a single calculation (in case it gets stuck, does not converge, it will be killed after this time)
OPTIMIZER = `'GA'`, `'PSA'` or `'PSAGA'`
FF = fitness function that needs to be defined separately (see example in Table .17)
IF_CALCULATIONS_FAIL = `'restart'` or `'new'`, determines what the program will do if the an error occurs, it can try to restart from the last point or start from a new random state
RESTORE = `'path'`, if a path is provided the program will start from the provided checkpoint (the program saves the results at every generation)
POPULATIONS = `['path1', 'path2', ...]`, if this variable is provided, the algorithm will create the initial population out of the populations provided in the list of paths, equally from each specimen bank

Table .16: Variables of the `SetParameters()` method of the `LoadingPatternOptimizer` class.

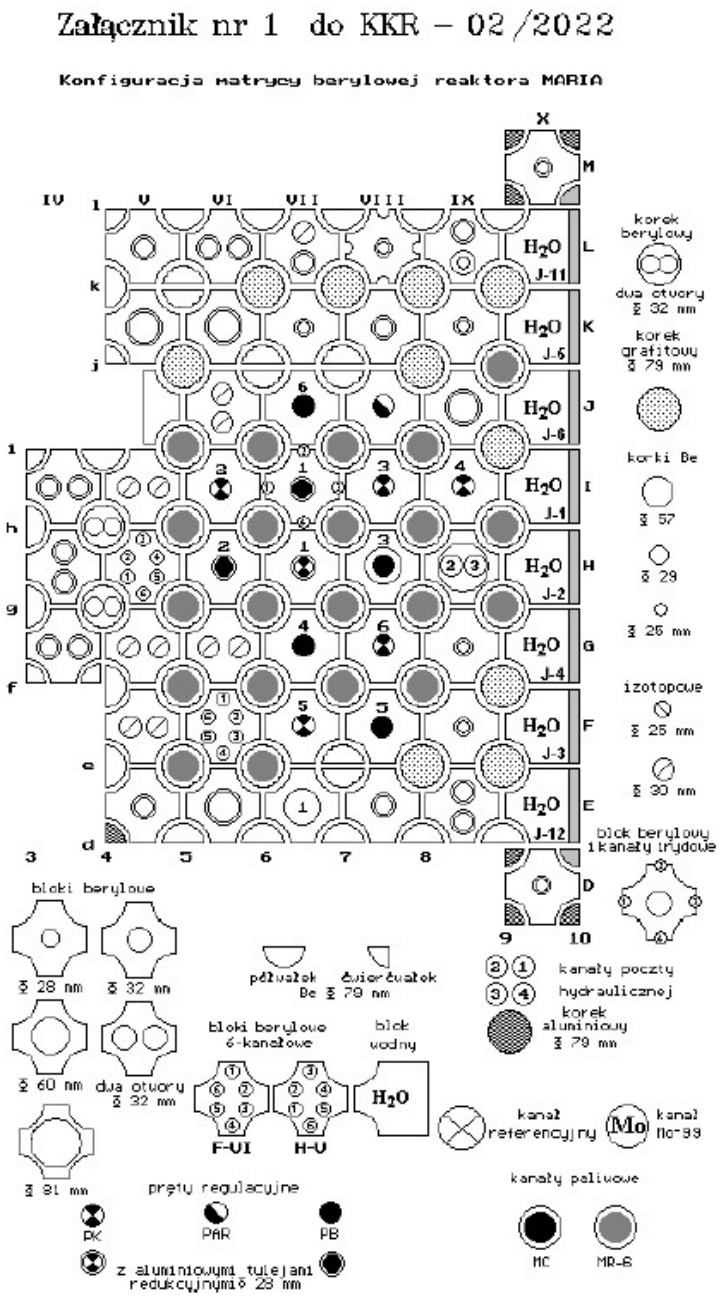
Definition of the `FF()` method

```
import maths as m

def FF(ind):
    FDH = ind.chromosome_parameters["FDH_max"]
    BU = ind.chromosome_parameters['ASSBU']
    CORBU = ind.chromosome_parameters['CORBUCYC']
    CORFUITES = ind.chromosome_parameters['CORFUITES_avg']

    if BU > 60_000:
        pB = m.exp(5e-5 * (60_000 - BU))
    else:
        pB = 1.0

    if  CORFUITES > 2.6169:
        pC = m.exp(0.1*(2.6169 - CORFUITES))
    else:
        pC = 1.0

    if FDH > 1.477:
        pF = m.exp(1.0*(1.4770 - FDH))
    else:
        pF = 1.0

    fitness_function_value =  (CORBU - 15000)/2229 * pB * pC * pF

    return fitness_function_value,
```

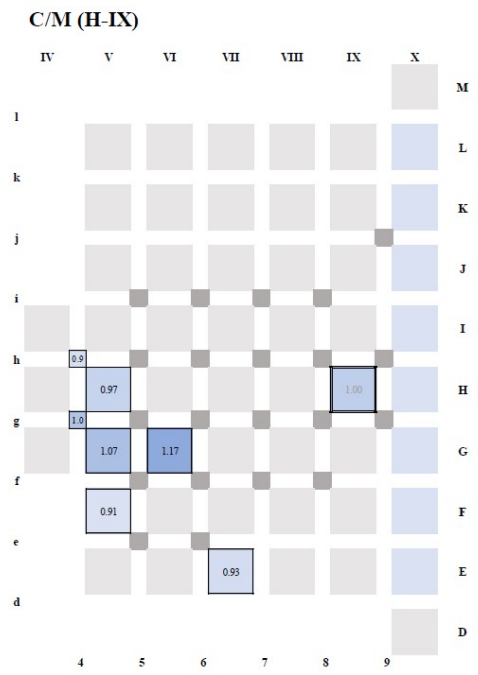
Table .17: Definition of the FF.

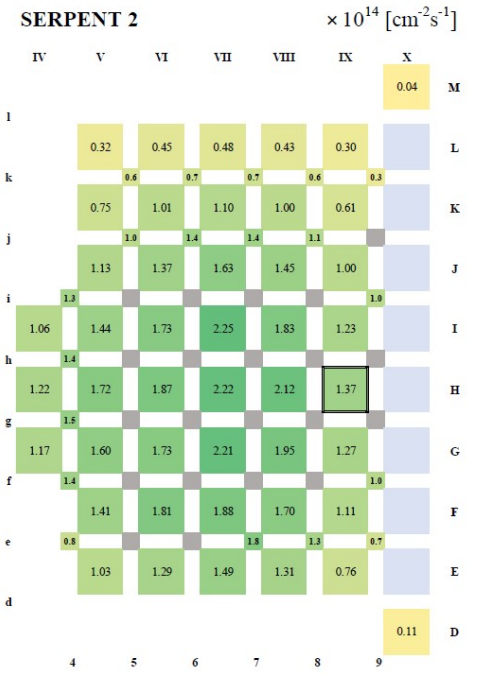
B Maria reactor core configuration

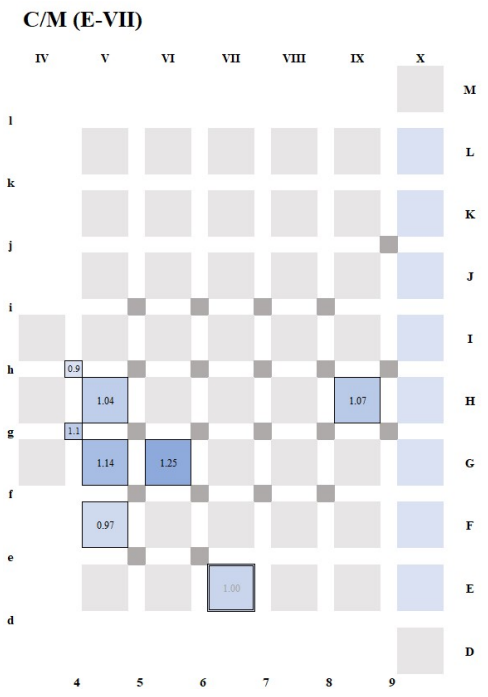
Core configuration of the Maria reactor form the period of the activation measurements in February 2022.

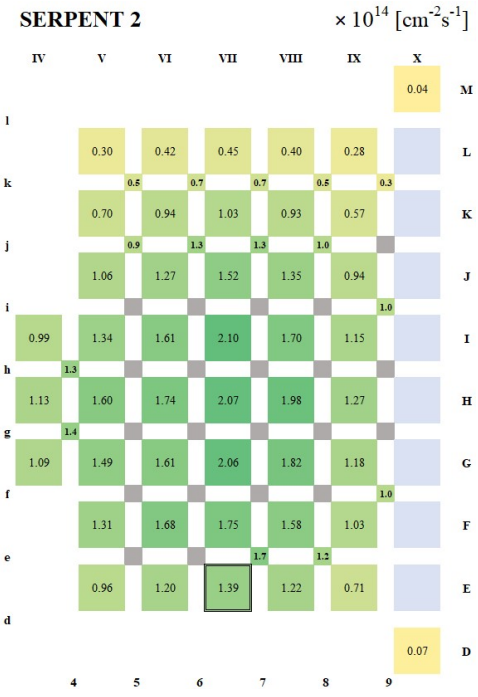


C Calculated neutron flux density distributions and correction factors relative to measurements at positions H-IX and E-VII









List of Figures

List of Tables

References

[1] N. Horelik and B. Herman. *MIT Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS)*. 2018. Rev 2.0.2, `https://www.osti.gov/biblio/22212933`.

[2] E. E. Lewis. *Fundamentals of Nuclear Reactor Physics*. Academic Press, 2008.

[3] *In-Core Fuel Management Programs for Nuclear Power Reactors: Final Report of the Co-ordinated Research Programme on Codes Adaptable to Small or Medium Size Computers in Developing Countries for In-Core Fuel Management*. IAEA-TECDOC-314, Vienna, 1984.

[4] Pedro H.S. Rodrigues, José Rubens Maiorino, Roberto Asano Jr, and Patrícia T.L. Asano. New approach to find the equilibrium cycle in PWR reactors using genetic algorithms. *Annals of Nuclear Energy*, 178:109333, 2022.

[5] W. Kubiński, P. Darnowski, and K. Chec. Optimization of the loading pattern of the PWR core using genetic algorithms and multi-purpose fitness function. *Nukleonika*, 66(4):147–151, 2021.

[6] W. Kubiński, P. Darnowski, and K. Chec. The development of a novel adaptive genetic algorithm for the optimization of fuel cycle length. *Annals of Nuclear Energy*, 155:108153, 2021.

[7] Samuel Glasstone and Alexander Sesonske. *Nuclear Reactor Engineering*, volume 2, 4th edition. Chapman & Hall, 1994.

[8] Tz. Haralampieva, A. Antonov, and M. Monev. An application of low leakage loading pattern to reduce fast neutrons Fluence on WWER-440 reactor pressure vessel in Kozloduy NPP. *Proceedings of the eleventh Symposium of Atomic Energy Research, (p. 853)*, 2001.

[9] Jake Mikouchi-Lopez, Gregory Delipei, and Jason Hou. Development and evaluation of parallel simulated annealing algorithm for reactor core optimization problems. *Nuclear Science and Technology Open Research*, 2:5, February 2024.

[10] W. Kubiński, G. Giorgi, and M. Segond. Multiobjective optimization of nuclear fuel multi-cycle management using genetic algorithms and parallel simulated annealing. In *Best Estimate Plus Uncertainty (BEPU) Conference*, Lucca, Italy, 2024.

[11] W. Kubiński, G. Giorgi, and M. Segond. Parallel Simulated Annealing, Genetic Algorithms and Hybrid Method applied to the Multiobjective Optimization of the Nuclear In-Core Fuel Management. In *International Conference on Physics of Reactors (PHYSOR)*, San Francisco, USA, 2024.

[12] W. Kubiński, P. Darnowski, and K. Chęć. Optimization of BEAVRS PWR Loading Pattern Using a Novel Genetic Algorithm Based on Population Variance Control. In *29th International Conference Nuclear Energy for New Europe (NENE-2020)*, 0214, 2020.

[13] International Energy Agency (IEA). Projected Costs of Generating Electricity 2020. `https://www.iea.org/reports/projected-costs-of-generating-\ electricity-2020`.

[14] U.S. Energy Information Administration. Country Analysis Brief: France. U.S. Department of Energy, October 2023.

[15] *Manual for Reactor Produced Radioisotopes*. Number 1340 in TECDOC Series. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2003.

[16] Nuclear Energy Agency (NEA). *Beneficial Uses and Production of Isotopes*. OECD Publishing, Paris, 1999.

[17] *Advances in Boron Neutron Capture Therapy*. Non-serial Publications. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2023.

[18] N. Knake, R. Prokopowicz, J. Cybowska, R. Kwiatkowski, A. Zawadka, and M.A. Gryziński. Initial Approach to Application of Gamma Spectroscopy to Alpha Particles Dosimetry in BNCT. *Radiation Protection Dosimetry*, 199(15-16):1926–1931, 2023.

[19] OECD/NEA. *The Supply of Medical Isotopes: An Economic Diagnosis and Possible Solutions*. OECD Publishing, Paris, 2019.

[20] *Medical Isotope Production Without Highly Enriched Uranium*. National Academies Press, Washington, D.C., May 2009. Pages: 12569.

[21] D.A. Brown, M.B. Chadwick, R. Capote, et al. ENDF/B-VIII.0: The 8th major release of the nuclear reaction data library with CIELO-project cross sections, new standards and thermal scattering data. *Nucl. Data Sheets*, 148:1, 2018.

[22] J. R. Ballinger. Short- and long-term responses to molybdenum-99 shortages in nuclear medicine. *British Journal of Radiology*, 83(995):899–901, November 2010.

[23] MARIA steps in to maintain Mo-99 supplies. 2023. World Nuclear News, https://world-nuclear-news.org/Articles/MARIA-steps-in-to-maintain-Mo-99-supplies, accessed: 2024-05-18.

[24] Matthew L. Wald. New Source of an Isotope in Medicine Is Found, February 16 2010. The New York Times, https://www.nytimes.com/2010/02/17/health/17isotope.html, accessed: 2024-05-18.

[25] Christophe Demazière. *Modelling of Nuclear Reactor Multi-physics: From Local Balance Equations to Macroscopic Models in Neutronics and Thermal-Hydraulics*. Elsevier, 1st edition, 2019.

[26] George I. Bell and Samuel Glasstone. *Nuclear Reactor Theory*. Van Nostrand Reinhold Company, New York, N.Y, 1 edition, 1970. Division of Technical Information, United States Atomic Energy Commission.

[27] A.J. Koning, D. Rochman, J. Sublet, N. Dzysiuk, M. Fleming, and S. van der Marck. TENDL: Complete Nuclear Data Library for Innovative Nuclear Science and Technology. *Nuclear Data Sheets*, 155:1, 2019.

[28] X-5 Monte Carlo Team. *MCNP - A General Monte Carlo N-Particle Transport Code, Version 5*. Los Alamos National Laboratory, Los Alamos, NM, 2008.

[29] Jaakko Leppänen. *Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code*, March 2013. User's Manual.

[30] MARIAn Kiełkiewicz. *Podstawy fizyki reaktorów jądrowych cz. I i II*. Wydawnictwo Politechniki Warszawskiej, Warszawa, 1977.

[31] International Atomic Energy Agency. Neutron Fluence Measurements. Technical Reports Series No. 107, International Atomic Energy Agency, Vienna, 1970.

[32] K. Pytel. Aktywacyjne pomiary neutronów. Technical Report Raport B IEA 10-95, Poland, 2003.

[33] J. Kenneth Shultis and Richard E. Faw. *Fundamentals of Nuclear Science and Engineering*. Marcel Dekker, Inc., New York, NY, USA, 2002.

[34] H. Anglart. *Applied Reactor Technology*. KTH Royal Institute of Technology, Stockholm, 2011.

[35] John H. Holland. Genetic Algorithms. *Scientific American*, 267(1):66–73, 1992.

[36] Zhan Li, Jincheng Wang, and Ming Ding. A review on optimization methods for nuclear reactor fuel reloading analysis. *Nuclear Engineering and Design*, 397:111950, 2022.

[37] E. Israeli and E. Gilad. Novel genetic algorithm for loading pattern optimization based on core physics heuristics. *Annals of Nuclear Energy*, 118:35–48, 2018.

[38] Dobroslav Grygar and René Fabricius. An Efficient Adjustment of Genetic Algorithm for Pareto Front Determination. *Transportation Research Procedia*, 40:1335–1342, 2019. TRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport.

[39] S. Chaimatanan D. Delahaye and M. Mongeau. Simulated annealing: From basics to applications. *International Series in Operations Research & Management Science*, 10 2019.

[40] D. J. Kropaczek. Concept for Multi-cycle Nuclear Fuel Optimization Based On Parallel Simulated Annealing With Mixing of States. *Proceedings of the PHYSOR Conference, Interlaken, Switzerland*, 2008.

[41] M. Louzazni, A. Khouya, K. Amechnoue, et al. Metaheuristic Algorithm for Photovoltaic Parameters: Comparative Study and Prediction with a Firefly Algorithm. *Applied Sciences*, 8(3):339, Feb 2018.

[42] A. Tjahjono, D. O. Anggriawan, A. K. Faizin, et al. Optimal Coordination of Overcurrent Relays in Radial System with Distributed Generation Using Modified Firefly Algorithm. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 7(4), Dec 2015.

[43] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. "a fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii". In Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI*, pages 849–858. Springer Berlin Heidelberg, 2000.

[44] Gregory K. Delipei, Jake Mikouchi-Lopez, Pascal Rouxelin, and Jason Hou. Reactor Core Loading Pattern Optimization with Reinforcement Learning. *M&C 2023 - Niagara Falls, Ontario, Canada*, 09 2023.

[45] Peter Wolf, Christian Hubschneider, Michael Weber, André Bauer, Jonathan Härtl, Fabian Dürr, and J. Marius Zöllner. Learning how to drive in a real world simulation with deep Q-Networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 244–250, 2017.

[46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[47] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic Algorithms and Applications, 2019.

[48] Gerald Goertzel. Minimum Critical Mass and Flat Flux. *J. Nuclear Energy*, 2:193–201, 1956. (Received 18 August 1955).

[49] Ian Wall and Henri Fenech. The Application of Dynamic Programing to Fuel Management Optimization. *Nuclear Science and Engineering*, 22(3):285–297, 1965.

[50] Daniel Tabak. Optimization of Nuclear Reactor Fuel Recycle via Linear and Quadratic Programming. *IEEE Transactions on Nuclear Science*, 15(1):60–64, February 1968.

[51] M. Melice. Pressurized Water Reactor Optimal Core Management and Reactivity Profiles. *Nuclear Science and Engineering*, 37:451–477, 1969.

[52] Koichi Sekimizu. Optimization of In-Core Fuel Management and Control Rod Strategy in Equilibrium Fuel Cycle. *Journal of Nuclear Science and Technology*, 12(5):287–296, 1975.

[53] T. O. Sauar. Application of linear programming to in-core fuel management optimization in light water reactors. *Nuclear Science and Engineering*, 46(2):274–283, 1971.

[54] J. O. Mingle. In-core fuel management via perturbation theory. *Nuclear Technology*, 27(2):248–257, 1975.

[55] R. R. Stout and A. H. Robinson. Determination of optimum fuel loadings in pressurized water reactors using dynamic programming. *Nuclear Technology*, 20(2):86–102, 1973.

[56] J. R. White, D. M. Chapman, and D. Biswas. Fuel management optimization based on generalized perturbation theory. In *Proceedings of the Topical Meeting on Advances in Fuel Management*, 1986.

[57] Tsutomu Hoshino. In-Core Fuel Management Optimization by Heuristic Learning Technique. *Nuclear Science and Engineering*, 49(1):59–71, 1972.

[58] D.J. Kropaczek. *In-Core Nuclear Fuel Management Optimization Utilizing Simulated Annealing*. Phd thesis, North Carolina State University, 1989.

[59] G. T. Parks. An Intelligent Stochastic Optimization Routine for Nuclear Fuel Cycle Design. *Nuclear Technology*, 89(2):233–246, 1990.

[60] Akio Yamamoto and Keiji Kanda. Comparison between Equilibrium Cycle and Successive Multicycle Optimization Methods for In-Core Fuel Management of Pressurized Water Reactors. *Journal of Nuclear Science and Technology*, 34(9):882–892, 1997.

[61] P. W. Poon and G. T. Parks. Application of genetic algorithms to in-core nuclear fuel management optimization. In *Proc. Joint Int. Conf. on Mathematical Methods and Supercomputing in Nuclear Applications*, volume 1, pages 777–786. ANS, 1993.

[62] E. Tanker and A. Z. Tanker. Application of a genetic algorithm to core reload pattern optimization. In *Proceedings of the International Conference on Reactor Physics and Reactor Computations*, 1994.

[63] G. T. Parks. Multiobjective Pressurized Water Reactor Reload Core Design by Nondominated Genetic Algorithm Search. *Nuclear Science and Engineering*, 124(1):178–187, 1996.

[64] C. Lin, J. I. Yang, K. J. Lin, and Z. D. Wang. Pressurized water reactor loading pattern design using the simple tabu search. *Nuclear Science and Engineering*, 129(1):61–71, 1998.

[65] Majdi I. Radaideh, Isaac Wolverton, Joshua Joseph, James J. Tusar, Uuganbayar Otgonbaatar, Nicholas Roy, Benoit Forget, and Koroush Shirvan. Physics-informed reinforcement learning optimization of nuclear assembly design. *Nuclear Engineering and Design*, 372:110966, 2021.

[66] Jun Zhao, Brian Knight, Ephraim Nissan, and Alan Soper. FuelGen: a genetic algorithm-based system for fuel loading pattern design in nuclear power reactors. *Expert Systems with Applications*, 14(4):461–470, 1998.

[67] Cláudio M. N. A. Pereira and Celso M. F. Lapa. Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem. *Annals of Nuclear Energy*, 30(5):555–565, 2003.

[68] CM Pereira and WF Sacco. A parallel genetic algorithm with niching technique applied to a nuclear reactor core design optimization problem. *Progress in Nuclear Energy*, 2008.

[69] Y. Kobayashi and E. Aiyoshi. Optimization of a Boiling Water Reactor Loading Pattern Using an Improved Genetic Algorithm. *Nuclear Technology*, 143(2):144–151, 2003.

[70] Wagner F. Sacco, Cassiano R. E. de Oliveira, and Cláudio M. N. A. Pereira. Two stochastic optimization algorithms applied to nuclear reactor core design. *Progress in Nuclear Energy*, 48(6):525–539, 2006.

[71] Fatih Alim, Kostadin N. Ivanov, Serkan Yilmaz, and Samuel H. Levine. New genetic algorithms (GA) to optimize PWR reactors: Part II: Simultaneous optimization of loading pattern and burnable poison placement for the TMI-1 reactor. *Annals of Nuclear Energy*, 35(1):113–120, 2008.

[72] Cecilia Martín-del Campo, Miguel-Ángel Palomera-Pérez, and Juan-Luis François. Advanced and flexible genetic algorithms for BWR fuel loading pattern optimization. *Annals of Nuclear Energy*, 36(10):1553–1559, 2009.

[73] Aneela Zameer, Sikander M. Mirza, and Nasir M. Mirza. Core loading pattern optimization of a typical two-loop 300MWe PWR using Simulated Annealing (SA), novel crossover Genetic Algorithms (GA) and hybrid GA(SA) schemes. *Annals of Nuclear Energy*, 65:122–131, 2014.

[74] Do Quang Binh, Ngo Quang Huy, and Nguyen Hoang Hai. A binary mixed integer coded genetic algorithm for multi-objective optimization of nuclear research reactor fuel reloading. *Kerntechnik*, 79(6):511–517, 2014.

[75] Yashar Rahmani. Reloading pattern optimization of VVER-1000 reactors in transient cycles using genetic algorithm. *Annals of Nuclear Energy*, 108:24–41, 2017.

[76] Zhan Li, Jie Huang, Jincheng Wang, and Ming Ding. Development and application of hybrid teaching-learning genetic algorithm in fuel reloading optimization. *Progress in Nuclear Energy*, 139:103856, 2021.

[77] Zhan Li, Jie Huang, and Ming Ding. Comparison and analysis of different selection strategies of genetic algorithms for fuel reloading optimization of Thorium-based HTGRs. *Nuclear Engineering and Design*, 373:110969, 2021.

[78] Chol So, Il-Mun Ho, Jong-Suk Chae, and Kwang-Hak Hong. PWR core loading pattern optimization with adaptive genetic algorithm. *Annals of Nuclear Energy*, 159:108331, 2021.

[79] Nadeem Shaukat, Ammar Ahmad, Bukhtiar Mohsin, Rustam Khan, Salah Ud-Din Khan, and Shahab Ud-Din Khan. Multiobjective Core Reloading Pattern Optimization of PARR-1 Using Modified Genetic Algorithm Coupled with Monte Carlo Methods. *Science and Technology of Nuclear Installations*, 2021:1802492, 2021.

[80] Chenghui Wan, Kaihui Lei, and Yisong Li. Optimization method of fuel-reloading pattern for PWR based on the improved convolutional neural network and genetic algorithm. *Annals of Nuclear Energy*, 171:109028, 2022.

[81] Yrobel Lima-Reinaldo and Juan-Luis François. Fuel loading pattern optimization of ALLEGRO fast reactor using genetic algorithms. *Annals of Nuclear Energy*, 180:109451, 2023.

[82] Jeffery Lewins and Martin Becker, editors. *Advances in Nuclear Science and Technology*, volume 26. Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2002. All rights reserved. No part of this eBook may be reproduced or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise, without written consent from the Publisher.

[83] R K Haling. OPERATING STRATEGY FOR MAINTAINING AN OPTIMUM POWER DISTRIBUTION THROUGHOUT LIFE. 10 1964.

[84] Y.P. Mahlers. Core reload optimization for equilibrium cycles using simulated annealing and successive linear programming. *Annals of Nuclear Energy*, 29(11):1327–1344, 2002.

[85] Y.P. Mahlers. Numerical study of optimal equilibrium cycles for pressurized water reactors. *Annals of Nuclear Energy*, 30(6):651–667, 2003.

[86] Davood Babazadeh, Mehrdad Boroushaki, and Caro Lucas. Optimization of fuel core loading pattern design in a VVER nuclear power reactors using Particle Swarm Optimization (PSO). *Annals of Nuclear Energy*, 36(7):923–930, 2009.

[87] Bahadır Mahmud Hudayi, Taekyun Lim, Bashir Kabir Ahmad, and Chang Joo Ha. Equilibrium Core Loading Pattern Search for APR1400 using Simulated Annealing Algorithm. In *Transactions of the Korean Nuclear Society Autumn Meeting*, Gyeongju, Korea, October 25-27 2023. Korean Nuclear Society.

[88] A. K. Ziver, J. N. Carter, C. C. Pain, C. R. E. de Oliveira, A. J. H. Goddard, and R. S. Overton. Multicycle Optimization of Advanced Gas-Cooled Reactor Loading Patterns Using Genetic Algorithms. *Nuclear Technology*, 141(2):122–141, 2003.

[89] A.K. Ziver, C.C. Pain, J.N. Carter, C.R.E. de Oliveira, A.J.H. Goddard, and R.S. Overton. Genetic algorithms and artificial neural networks for loading pattern optimisation of advanced gas-cooled reactors. *Annals of Nuclear Energy*, 31(4):431–457, 2004.

[90] Peng Hong Liem, Hoai Nam Tran, and Tagor Malem Sembiring. Design optimization of a new homogeneous reactor for medical radioisotope mo-99/tc-99m production. *Progress in Nuclear Energy*, 2014.

[91] A. T. Khotbeh-Sara, F. Rahmani, F. Ghasemi, and S. Ahmadian-namin. Optimization study on production of mo-99 using high power electron accelerator linac. In *Proceedings of IPAC2017*, pages 4667–4668, Copenhagen, Denmark, 2017.

[92] Andrew Conant, Noel Nelson, Jorge Navarro, Terry Grimm, Chad Denbrock, and Robert Wahlen. Mcnp neutronic design optimization of an accelerator-driven subcritical assembly for mo-99 production. In *Conference on Neutronic Design Optimization*, August 2023.

[93] Daiyuan Chen, Ricardo dos Santos Augusto, Yuanhong Li, Zhi Qin, Jian Rong, Kaiqiang Yao, Huan Jia, Chenzhang Yuan, Juntao Liu, and Zhiyi Liu. Shielding assessment and optimization of the target station for medical isotope production based on superconducting proton linac. *Applied Sciences*, 13:1985, 2023.

[94] B. Meftah, T. Zidi, and A. Bousbia-Salah. Neutron flux optimization in irradiation channels at NUR research reactor. *Annals of Nuclear Energy*, 33(14):1164–1175, 2006.

[95] Afshin Hedayat. Developing a practical optimization of the refueling program for ordinary research reactors using a modified simulated annealing method. *Progress in Nuclear Energy*, 76:191–205, 2014.

[96] Susan Hogle. *Optimization of Transcurium Isotope Production in the High Flux Isotope Reactor*. PhD thesis, The University of Tennessee, Knoxville, 2012.

[97] Timothée Kooyman, Laurent Buiron, and Gérald Rimpault. Analysis and optimization of minor actinides transmutation blankets with regards to neutron and gamma sources. *EPJ Nuclear Sci. Technol.*, 3:7, 2017.

[98] A. Tsechanski, D. Fedorchenko, V. Starovoitova, and A. Galperin. Converter optimization for photonuclear production of mo-99. *Nuclear Inst. and Methods in Physics Research B*, pages 118–123, 2019.

[99] Charles R. Daily and Joel L. McDuffee. Design studies for the optimization of 238pu production in npo2 targets irradiated at the high flux isotope reactor. *Nuclear Technology*, 206(8):1182–1194, 2020.

[100] Cameron Ian Salyer. Meta-heuristic optimization techniques for the production of medical isotopes through special target design. Master's thesis, University of Tennessee, May 2022.

[101] Y Zhao, RM Edwards, and KY Lee. Hybrid feedforward and feedback controller design for nuclear steam generators over wide range operation using genetic algorithm. *IEEE Transactions on Energy Conversion*, 1997.

[102] M. Schlieck, H.-D. Berger, and A. Neufert. Optimized gadolinia concepts for advanced in-core fuel management in PWRs. *Nuclear Engineering and Design*, 205(1):191–198, 2001.

[103] Byeong Soo Kim and Joo Hyun Moon. Use of a genetic algorithm in the search for a near-optimal shielding design. *Annals of Nuclear Energy*, 37(2):120–129, 2010.

[104] Zhenping Chen, Zhenyu Zhang, Jinsen Xie, Qian Guo, and Tao Yu. Metaheuristic optimization method for compact reactor radiation shielding design based on genetic algorithm. *Annals of Nuclear Energy*, 134:318–329, 2019.

[105] Ying Dongchuan, Xiao Feng, Zhang Hongyue, Lv Huanwen, Tan Yi, Liu Jiajia, Jing Futing, and Tang Songqian. Study on optimization methods of nuclear reactor radiation shielding design using genetic algorithm. *Nuclear Power Engineering*, 37(4):160–164, 2016.

[106] John Pevey, Ondřej Chvála, Sarah Davis, Vladimir Sobes, and J. Wes Hines. Genetic Algorithm Design of a Coupled Fast and Thermal Subcritical Assembly. *Nuclear Technology*, 206(4):609–619, 2020.

[107] V. Solans, D. Rochman, C. Brazell, et al. Optimisation of used nuclear fuel canister loading using a neural network and genetic algorithm. *Neural Computing and Applications*, 33:16627–16639, 2021.

[108] Bin Liu, Huanwen Lv, Lan Li, Xiaoming Chai, Hu Xu, Yi Tan, Bangyang Xia, and Cong Chen. Multi-Objective Optimization Design of Radiation Shadow Shield for Space Nuclear Power With Genetic Algorithm. *Frontiers in Energy Research*, 10, 2022.

[109] C. Wang, C. Yan, J. Wang, L. Chen, and G. Li. Application of Dual-Adaptive Niched Genetic Algorithm in Optimal Design of Nuclear Power Components. In *Proceedings of the 2014 22nd International Conference on Nuclear Engineering*, volume 2A: Thermal Hydraulics, page V02AT09A013. ASME, 2014.

[110] Zheng Liu, Jianjun Wang, Sichao Tan, Shouxu Qiao, and Hu Ding. Multi-objective optimal design of the nuclear reactor pressurizer. *International Journal of Advanced Nuclear Reactor Design and Technology*, 1:1–9, 2019.

[111] João J. da Cunha, Antonio Carlos M. Alvim, and Celso Marcelo Franklin Lapa. Reduced scale PWR passive safety system designing by genetic algorithms. *Annals of Nuclear Energy*, 34(12):1014–1023, 2007.

[112] M. Rafiei and G.R. Ansarifar. Space nuclear reactor fuel design based on dynamic analysis and GA & PSO optimization. *Progress in Nuclear Energy*, 2022.

[113] Jiaoshen Xu, Hui Tang, Xin Wang, Ge Qin, Xin Jin, and Dongyuan Li. NSGA-II algorithm-based LQG controller design for nuclear reactor power control. *Annals of Nuclear Energy*, 169:108931, 2022.

[114] Brian Andersen and David J. Kropaczek. MOOGLE: A Multi-Objective Optimization tool for three-dimensional nuclear fuel assembly design. *Progress in Nuclear Energy*, 155:104518, 2023.

[115] W. Hongchun. Pressurized water reactor reloading optimization using genetic algorithms. *Annals of Nuclear Energy*, 2001.

[116] J.M. Gozalvez, S. Yilmaz, F. Alim, and K. Ivanov. Sensitivity study on determining an efficient set of fuel assembly parameters in training data for designing of neural networks in hybrid genetic algorithms. *Annals of Nuclear Energy*, 2006.

[117] Umarov Shokhmirzo, Jungseon An, and Chang Joo Hah. Fuel Batch Optimization for Extra Longer Initial Core Design of APR-1400. In *Transactions of the Korean Nuclear Society Autumn Meeting*, Changwon, Korea, October 19-21 2022. Korean Nuclear Society.

[118] C. M. N. A. Pereira, R. Schirru, and A. S. Martinez. Genetic algorithms applied to nuclear reactor design optimization. In *Fuzzy Systems and Soft Computing in Nuclear Engineering*, volume 38 of *Studies in Fuzziness and Soft Computing*, pages 315–334. Springer, 2000.

[119] Jerzy Kubowski. *Elektrownie jądrowe*. Wydawnictwo Naukowe PWN, WNT, Warszawa, wyd. 2 edition, 2017.

[120] Y. Kobayashi and E. Aiyoshi. Optimization of Boiling Water Reactor Loading Pattern Using Two-Stage Genetic Algorithm. *Nuclear Science and Engineering*, 142(2):119–139, 2002.

[121] Juan-Luis François, Juan-Rafael Guzmán, Cecilia Martín del Campo, and Miguel Ángel Palomera. Design and optimization of an equilibrium reload with MOX fuel with minor actinides. *Progress in Nuclear Energy*, 53(6):566–570, 2011. Select Papers from Advances in Nuclear Fuel Management IV, Hilton Head Island, SC, April 2009.

[122] Thi Thuy Giang Phan, Hoai Nam Tran, Quang Binh Do, Quang Huy Ngo, and Tuan Anh Tran. Comparative analysis of the differential evolution and genetic algorithm applied to the nuclear reactor fuel reloading optimization. In *Vietnam Conference on Nuclear Science and Technology VINANST-13*, page 8, August 2019.

[123] Afshin Hedayat. Conceptual design of a high-performance hybrid object for applications of the fast neutron irradiation in mtrs. *Research and Practice in Engineering*, 2020.

[124] A Kumar and PV Tsvetkov. A new approach to nuclear reactor design optimization using genetic algorithms and regression analysis. *Annals of Nuclear Energy*, 2015.

[125] W. Żurkowski, P. Sawicki, W. Kubiński, and P. Darnowski. Application of genetic algorithms in optimization of SFR nuclear reactor design. *Nukleonika*, 66(4):139–145, 2021.

[126] N. Horelik, B. Herman, B. Forget, and K. Smith. Benchmark for evaluation and validation of reactor simulations (BEAVRS). 7 2013. `https://www.osti.gov/biblio/22212933`.

[127] P. Darnowski and Michał Pawluczyk. Analysis of the BEAVRS PWR benchmark using SCALE and PARCS. *Nukleonika*, 64(3):87–96, 2019.

[128] Bradley T. Rearden and Matthew Anderson Jessee. SCALE Code System. 3 2018.

[129] T. Downar, Y. Xu, V. Seker, and N. Hudson. *PARCS v3.0 U.S. NRC Core Neutronics Simulator User Manual*. University of Michigan, 2010.

[130] B. J. Ade. *SCALE/TRITON Primer: A Primer for Light Water Reactor Lattice Physics Calculations*. ORNL, 11 2012.

[131] O. Tal, E. Israeli, P. Ravetto, and E. Gilad. The adjoint problem as physical heuristic for loading pattern optimization. *Annals of Nuclear Energy*, 134:226–234, 2019.

[132] Cecilia Martin del Campo, Juan Francois, Roberto Carmona, and Ivonne Oropeza. Optimization of BWR fuel lattice enrichment and gadolinia distribution using genetic algorithms and knowledge. *Annals of Nuclear Energy*, 34:248–253, 04 2007.

[133] F. Khoshahval and Amir Fadaei. Application of a hybrid method based on the combination of genetic algorithm and Hopfield neural network for burnable poison placement. *Annals of Nuclear Energy*, 47:62–68, 09 2012.

[134] A. Norouzi, A. Zolfaghari, A.H. Minuchehr, and F. Khoshahval. An enhanced integer coded genetic algorithm to optimize PWRs. *Progress in Nuclear Energy*, 53(5):449–456, 2011.

[135] C. Martín del Campo, M. A. Palomera-Pérez, and J. L. François. Advanced and flexible genetic algorithms for BWR fuel loading pattern optimization. *Annals of Nuclear Energy*, 36:1553–1559, 10 2009.

[136] Serkan Yilmaz, Kostadin Ivanov, and Samuel Levine. Application of genetic algorithm to optimize burnable poison placement in pressurized water reactors. volume 33, pages 1477–1483, 06 2005.

[137] Serkan Yilmaz, Kostadin Ivanov, Samuel Levine, and Moussa Mahgerefteh. Genetic Algorithm to Optimize the UO2/Gd2O3 Fuel Pin Designs in a Pressurized Water Reactor. *Nuclear Technology*, 156(2):168–179, 2006.

[138] Cláudio M.N.A. Pereira. Evolutionary multicriteria optimization in core designs: basic investigations and case study. *Annals of Nuclear Energy*, 31(11):1251–1264, 2004.

[139] Y. Kobayashi and E. Aiyoshi. Optimization of boiling water reactor loading pattern using an improved genetic algorithm. volume 143, pages 383 – 390, 02 2001.

[140] Juan José Ortiz and Ignacio Requena. An Order Coding Genetic Algorithm to Optimize Fuel Reloads in a Nuclear Boiling Water Reactor. *Nuclear Science and Engineering*, 146(1):88–98, 2004.

[141] D. E. Goldberg and J. H. Holland. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Mach. Learn., vol. 3, no. 2, pp. 95–99, 1979.

[142] E. Israeli and E. Gilad. Novel Genetic Algorithms for Loading Pattern Optimization Using State-of-the-Art Operators and a Simple Test Case. *ASME Journal of Nuclear Radiation Science*, 3(3):030901, 2017.

[143] Ben Desjardins, Rafael Falcon, Rami Abielmona, and Emil Petriu. *Planning Robust Sensor Relocation Trajectories for a Mobile Robot with Evolutionary Multi-objective Optimization*, chapter 1, pages 179–210. Computational Intelligence in Wireless Sensor Networks, 01 2017.

[144] Guido van Rossum et al. Python Setup and Usage, Release 3.8.16. `https://docs.python.org/3.8/`, 2023/03/30.

[145] H. W. Graves. *Nuclear Fuel Management*. John Wiley & Sons, Inc., 1979.

[146] J. A. Sefcik, M. J. Driscoll, and D. D. Lanning. Analysis of Strategies for Improving Uranium Utilization in Pressurized Water Reactors. Technical Report MIT-EL-80-132, M.I.T. Energy Laboratory, January 1981.

[147] New Horizon Scientific, LLC. Maine Yankee Decommissioning Experience Report: Detailed Experiences 1997 - 2004. Experience report, New Horizon Scientific, LLC, 661 Oakhurst Court, Naperville, IL 60540, 2004. Principal Investigator: R. Aker.

[148] W. T. Loh, M. J. Driscoll, and D. D. Lanning. The Use of Burnable Poison to Improve Uranium Utilization in PWRs. Energy Laboratory Report MITNE-250, Massachusetts Institute of Technology, Nuclear Engineering Department, Cambridge, MA, May 1982.

[149] Nicholas Tsoulfanidis. *The Nuclear Fuel Cycle*. American Nuclear Society, 3rd edition, 2013.

[150] Rüdiger Leverenz, Ludwig Gerhard, and Andreas Göbel. The European Pressurized Water Reactor: A Safe and Competitive Solution for Future Energy Needs. *Conference Nuclear Energy for New Europe (NENE), Portorož, Slovenia*, 2004.

[151] Rüdiger Leverenz. The EPR - A Safe and Competitive Solution for Future Energy Needs. *Conference Nuclear Energy for New Europe, Portorož, Slovenia*, 2006.

[152] IAEA. *Status report 78 - The Evolutionary Power Reactor (EPR)*. 04-04-2011, `https://aris.iaea.org/`.

[153] Areva. *EPR brochure* . `https://web.archive.org/web/20071129121411/http://www.areva-np.com/common/liblocal/docs/Brochure/EPR_US_%20May%202005.pdf`, 2005.

[154] Framatome ANP, Inc. *EPR Design Description*. `https://www.nrc.gov/docs/ML0522/ML052280170.pdf`, 2005.

[155] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. Deap: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13(70):2171–2175, 2012.

[156] Do Quang Binh, Ngo Quang Huy, and Nguyen Hoang Hai. A binary mixed integer coded genetic algorithm for multi-objective optimization of nuclear research reactor fuel reloading. *Kerntechnik*, 79(6):511–517, 2014.

[157] Ella Israeli and Erez Gilad. Novel Genetic Algorithms for Loading Pattern Optimization Using State-of-the-Art Operators and a Simple Test Case. *Journal of Nuclear Engineering and Radiation Science*, 3, 02 2017.

[158] Mark Deinert, Nadeem Shaukat, and et al. Multiobjective Core Reloading Pattern Optimization of PARR-1 Using Modified Genetic Algorithm Coupled with Monte Carlo Methods. *Science and Technology of Nuclear Installations*, 2021:1802492, 2021.

[159] Boyan Ivanov and David J. Kropaczek. Assessment of parallel simulated annealing performance with the NEXUS/ANC9 core design code system. *EPJ Web of Conferences*, 2021.

[160] David J. Kropaczek. COPERNICUS: A multi-cycle optimization code for nuclear fuel based on parallel simulated annealing with mixing of states. *Progress in Nuclear Energy*, 53(6):554–561, 2011. Advances in Nuclear Fuel Management IV, Hilton Head Island, SC, April 2009.

[161] Hongjing Wei, Shaobo Li, Houmin Jiang, Jie Hu, and Jianjun Hu. Hybrid genetic simulated annealing algorithm for improved flow shop scheduling with makespan criterion. *Applied Sciences*, 8(12), 2018.

[162] P. Girieud, L. Daudin, C. Garat, P. Marotte, and S. Tarle. SCIENCE Version 2: The Most Recent Capabilities of the Framatome 3D Nuclear Code Package. 2001.

[163] Iheb Toumi, André Bergeron, Gallo Danielle, Eric Royer, and D. Caruge. FLICA-4: A three-dimensional two-phase flow computer code with advanced numerical methods for nuclear applications. *Nuclear Engineering and Design*, 200:139–155, 08 2000.

[164] Richard Sanchez, Igor Zmijarevi, M. Coste-Delclaux, Emiliano Masiello, Simone Santandrea, Emanuele Martinolli, Laurence Villate, Nadine Schwartz, and Nathalie Guler. Apollo2 year 2010. *Nuclear Engineering and Technology*, 42(5):474–499, Oct 2010.

[165] Paul Seurin and Koroush Shirvan. Assessment of Reinforcement Learning Algorithms for Nuclear Power Plant Fuel Optimization, 2023.

[166] K. Palmi, W. Kubinski, and P. Darnowski. Prediction of the evolution of the nuclear reactor core parameters using artificial neural network, arXiv, 2304.10337, 2023.

[167] Marek Migdal, Emilia Balcer, et al. Maria reactor irradiation technology capabilities towards advanced applications. *Energies*, 14(23):8153, December 2021.

[168] K. Pytel, W. Bak, Borek-Kruszewska, et al. Eksploatacyjny Raport Bezpieczenstwa Reaktora MARIA. Technical report, National Centre for Nuclear Research, 2015.

[169] Narodowe Centrum Badań Jądrowych. Wytwarzanie izotopów promieniotwórczych, 2019. Accessed: 2024-06-12.

[170] Janusz Jaroszewicz, Zuzanna Marcinkowska, and Krzysztof Pytel. Production of Fission Product Mo using High-Enriched Uranium Plates in Polish Nuclear Research Reactor MARIA: Technology and Neutronic Analysis. *Nukleonika*, 59(2):43–52, 2014.

[171] National Academies of Sciences, Engineering, and Medicine. *Molybdenum-99 for Medical Imaging*. The National Academies Press, Washington, DC, 2016.

[172] Jean-Christophe Sublet. The FISPACT-II user manual. 2015.

[173] Nuclear Energy Agency. JANIS (Java-based Nuclear Information Software). `https://www.oecd-nea.org/jcms/pl_39910/janis`, 2023. Accessed: 2023-09-17.

[174] *Cost Aspects of the Research Reactor Fuel Cycle*. Number NG-T-4.3 in Nuclear Energy Series. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2010.

[175] FISPACT Wiki. CCFE-709 Group Structure, n.d. `https://fispact.ukaea.uk/wiki/CCFE-709_group_structure`, accessed: 2024-06-11.

[176] S.K. Kim, W.I. Ko, S.R. Youn, and R.X. Gao. Nuclear fuel cycle cost estimation and sensitivity analysis of unit costs on the basis of an equilibrium model. *Nuclear Engineering and Technology*, 47(3):306–314, 2015.

[177] Ioannis N. Kessides. Nuclear power: Understanding the economic risks and uncertainties. *Energy Policy*, 38(8):3849–3864, 2010.

[178] International Atomic Energy Agency. *Cost Aspects of the Research Reactor Fuel Cycle*. Number No. NG-T-4.3 in IAEA Nuclear Energy Series. Vienna, Austria, 2010. STI/PUB/1440.

[179] Matt Bowen, Emeka Ochu, and James Glynn. The Uncertain Costs of New Nuclear Reactors: What Study Estimates Reveal about the Potential for Nuclear in a Decarbonizing World. Research report, Columbia University, Center on Global Energy Policy, December 2023.

[180] Państwowa Agencja Prasowa. *Resort nauki nie ma pieniędzy na NCBJ. Kłopoty reaktora MARIA*. 2016. https://biznesalert.pl/resort-nauki-pieniedzy-ncbj-klopoty-reaktora-MARIA/, Accessed: 2024-06-09.

[181] Polskie Towarzystwo Nukleoniczne. Postępy Techniki Jądrowej. 67(1), 2024.

[182] National Nuclear Security Administration. 2021 CRM Price List, November 2021. `https://www.energy.gov/sites/prod/files/2020/11/f80/2021%20CRM%20Price%20List%20Draft%20V1_Nov%202020.pdf`, accessed: date-of-access.