


# Experiments and fuel reloads optimization for the Jules Horowitz Reactor (JHR) using a genetic algorithm combined with a convolutional neural network

Douchka Dimitrijevic, and Guillaume Ritter\* 

Commissariat à l’Energie Atomique et aux Energies Renouvelables, Direction des Energies, IRESNE/DER/SERJH/LFSC, Cadarache 13108, St Paul Lez Durance, France

Received: 6 December 2024 / Received in final form: 11 July 2025 / Accepted: 13 November 2025

**Abstract.** This paper presents the experiments and fuel reloads optimization of the Jules Horowitz research Reactor (JHR) using a genetic algorithm coupled with a convolutional neural network. At the end of a cycle, the most burned fuel elements will be removed and replaced by fresh fuel elements. Since the purpose of this reactor is to conduct experiments, it is crucial to reach experimental performances. With this aim, the goal was to find the best location for each fuel element for the next cycle, i.e. the optimal core configuration that satisfies not only operations and safety constraints but also experimental goals. The analysis was based on an algorithm using two Python libraries: PyGAD for the genetic algorithm and TensorFlow for the convolutional neural network, thus providing a simple implementation of the method. It has been shown the convolutional neural network predicts the core parameters in 12 ms with a low error. Moreover, the genetic algorithm converges toward an optimal core configuration in a very short computation time ( $\sim 7.5$  minutes) as well. This paper demonstrates that the combination of a convolutional neural network and a genetic algorithm allows obtaining the optimal core configuration for a research reactor in particular to maximize its experimental performances.

## 1 Introduction

The Jules Horowitz Reactor (JHR), currently under construction at the Cadarache CEA site, is a unique 100 MW research reactor that will be used to test materials and fuels under irradiation in support of current and future nuclear reactors [1]. The production of radioisotopes for nuclear medicine and non-nuclear industry will also be ensured. At the end of an operation cycle in the JHR, the most depleted fuel elements of the core will be removed and replaced by fresh fuel elements in order to hold the duration of the next cycle. In order to minimize radioactive waste, the fuel elements have to be depleted as much as possible. Moreover, experimental performances that depend on the reload pattern have to be reached. Thus, the main challenge is to find the best location of each fuel element in the reactor core for the next cycle so that safety, economical and experimental constraints are satisfied.

This type of optimization problems was initially heuristically approached by “nuclear reload design engineers” with intuition and field computations [2]. However, the development of numerical approaches brought so sig-

nificant improvements that it now appears more efficient to rely on artificial methods [3].

Since 1990s, the researchers have proposed different optimization methods for nuclear core reload problems such as simulated annealing algorithms, tabu search algorithms, evolutionary algorithms, particle swarm optimization, and ant colony optimization [4]. Among these methods, genetic algorithms are commonly chosen for this optimization problem. It was introduced for the search of an optimal fuel arrangement within a power reactor core (e.g. [2,5]). Another study showed good results in terms of precision and computational cost for the fuel management optimization of a PWR [6]. The approach was also successfully adapted to research reactors where an objective function was used to maximize the effective multiplication factor and minimize the power peaking factor [7]. Besides, reference [8] presents the definitions of the basic concepts of genetic algorithms for the reload problem and applied them as an example to the BN type Russian reactor. Reference [9] proposed new geometric genetic operators and fitness function construction. Multiobjective optimization of the core reloading pattern of PARR-1 was also made using a genetic algorithm coupled with Monte Carlo methods [10]. The effective multiplication factor and

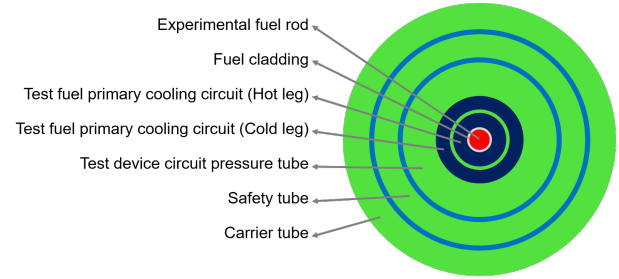
\* e-mail: [guillaume.ritter@cea.fr](mailto:guillaume.ritter@cea.fr)

thermal neutron flux were maximized while the power peaking factor and position of control fuel elements were constrained. Furthermore, a multiobjective optimization for a TRIGA type research reactor was successfully completed [11] where three parameters were optimized: the effective multiplication factor was maximized, the center fuel temperature was minimized, and the departure from nucleate boiling ratio was maximized. This last study used a genetic algorithm coupled with neutronics and thermal-hydraulics codes that was able to converge toward an optimal core configuration.

The use of machine learning methods has increased especially in nuclear engineering. These methods are used to solve different problems such as health and monitoring, radiation detection and optimization [12]. In particular, artificial neural networks have been used to predict PWR core parameters [13], fuel cycle scenarios [14] or some burnup calculations [15]. In another study, a neural network was also used to predict core characteristics such as peaking factor, cycle length and maximum burnup in order to improve optimization efficiency [16]. A recent study has also investigated various artificial neural network architectures to predict the cycle length of a typical PWR with a very good accuracy [17]. Convolutional neural networks, that are a type of networks used for computer vision, were introduced in the 2020s to predict core parameters [18]. They showed significantly higher performances than artificial neural network. Another study combined computer vision approaches and modular neural network to successfully predict the target neutronics parameters [19]. To current knowledge, convolutional neural networks have not yet been used for both core parameters and experimental performances prediction.

Furthermore, some studies already combined genetic algorithms optimization methods and machine-learning methods to reduce computation time for the core reload problem. A study developed a genetic algorithm to search for the optimal core configuration of PWR and an artificial neural network to predict several core parameters [20]. Recently, a study used this time a convolutional neural network and a genetic algorithm to optimize the fuel-reloading pattern of a PWR [21]. The study showed that the optimal configuration was obtained in a very short time ( $\sim 20$  minutes). Lastly, [22] used a similar method to rapidly obtain the fuel reloading patterns for HPR-1000. However, to current knowledge, the method coupling a genetic algorithm with a convolutional neural network has never been used for research reactors, especially to meet experimental expectations.

Facing an optimization problem, several options can be envisaged [4]. A cognitive and heuristic approach would be sound in the first place. It uses physics observations and heuristics rules that provide helpful guidance toward an operational configuration and above all a justification for the final choice. However, this strategy can seldom be exhaustive and is not necessarily closest to the optimum [2]. A formal mathematical approach would thus go beyond a handful of intuitions. Previous studies have converged on Genetic Algorithms which has been used as a basis for this paper. It has been coupled with a surrogate model to find an optimal JHR core configuration. A



**Fig. 1.** Outline of the ADELIN test device (not to scale).

core configuration is defined by the position of each fuel element in the core. Since there are 34 fuel elements in the JHR, there are  $34!/10! = 81.36 \times 10^{30}$  possible configurations obtained by shuffling 10 new fuel elements in the core. The choice of using a genetic algorithm coupled with a fast and reliable surrogate model was made because it allows converging toward a good enough configuration with very low computational time [8]. It was assumed convolutional neural networks would best match the constraints of speed and reliability in this study. An optimal configuration is one that optimizes the fuel use: the Beginning of Cycle effective multiplication factor  $K_{\text{eff}}$  should be maximized to ensure the cycle duration. The JHR includes different locations, where test devices will be placed to test experimental fuels. The JHR core is surrounded by a large neutron reflector made of beryllium and water, providing an intense thermal flux to experiments. In particular, the ADELIN irradiation loop, which is located in this reflector, will contain one PWR experimental rod [23]. This device will be placed on a so called displacement system in order to create power ramps in order to experimentally reproduce PWR incidents. Thus, an optimal core configuration is one that also maximizes the linear power in the experimental rod  $P_{\text{exp}}$ . Eventually, an optimal configuration is one that satisfies safety constraints: the maximal power peaking Factor  $F3D$  should be lower than 2.76.

## 2 Problem description

The ADELIN irradiation loop, that will be placed in the reflector, is dedicated to study the behaviour of fuel rods undergoing power ramps transients [23]. Since the device is placed on a displacement system, different types of power ramps can be performed with this device. They allow defining reliable Stress Corrosion Cracking-Pellet Cladding Interaction failure thresholds [23]. Figure 1 shows a radial cut of the ADELIN test device. The experimental fuel rod, in red on Figure 1, is placed at the center of the device and surrounded by a dark blue cooling circuit that reproduces PWR conditions. Pressure, safety and carrier tubes, in green, are nested inside each other.

The JHR operates on a 27 Equivalent Full Power Days cycle length basis. The reactivity is controlled with the help of 23 control rods. These neutron absorbers include 2 separate groups. Power steering absorbers stand in the middle of the core and are used for divergence and in any power adjustment phase. Moreover, the consecutive

extraction of each among 19 compensation absorbers is used to keep the reactor critical in spite of the reactivity loss caused by fuel burnup. Eventually, additional safety absorbers are always out of the core during operations and will be triggered by any safety event. For the search of an optimal core configuration, the extraction sequence of the compensation absorbers was fixed.

The JHR uses 34 mechanically identical high-enriched fuel elements. The depletion and isotopic composition of a fuel element at the end of a cycle depends on its position in the core during each new cycle. A typical fuel element configuration is arbitrarily set after several out-in reloads providing an equilibrium situation as shown in Figure 2. It shows a typical heatmap of burnups for each fuel element in the JHR core. Fuel cell numbers are shown at the top of each fuel element location, while fuel element burnup is added below. It shows yellow fresh elements that have reached a maximum burnup of 32 GWd/t at the End Of their first Cycle. Fuel elements with one, two or three additional cycles have burnups of about 60, 90 or 120 GWd/t respectively. The JHR core has room for 37 fuel cells, but 3 of them will be occupied by experiments instead of fuel elements. In this study, these 3 locations are filled with dummy test devices and leave empty spaces on visual representations.

From this End Of Cycle core configuration, the most burned fuel elements are removed and replaced by 10 fresh fuel elements. The “reload engineer” challenge is to find the optimal core configuration for the next cycle by determining the best location for each fuel element for a particular objective.

In our case, the fuel reload pattern optimization is multi-objective. In order to hold the cycle duration, the  $K_{\text{eff}}$  is maximized. Furthermore, to reach the desired experimental performances, the linear power in the experimental rod is maximized. Since the safety heat removal limits could be exceeded by the fuel reloading pattern, the power peaking factor  $F3D$  is minimized and constrained by the safety limit value. These three parameters are jointly optimized: an optimal configuration of the core would consist in a high  $K_{\text{eff}}$ , a high linear power  $P_{\text{exp}}$  in the experimental rod and a low power peaking factor  $F3D$  below the safety limit value.

## 3 Methodology

### 3.1 Core physics calculations

#### 3.1.1 Main configuration

In order to analyze on a physics point of view the optimal core configuration, a preliminary neutronics study was made. The goal is to investigate, understand and justify the impact of the core configuration on experimental performance. For this purpose, core physics calculations were made with TRIPOLI-4<sup>®</sup>, a continuous-energy radiation transport Monte Carlo code developed by CEA [24]. For this preliminary study, the reference core is filled with 34 initially high-enriched fuel elements each with a burnup of 50.7 GWd/t which corresponds to an average burnup

of core fuel elements in a typical equilibrium situation. The reflector is filled with beryllium and dummy experiments except for the ADELINe test device. This device is placed next to the core and contains one experimental rod. The experimental rod is fresh and 5% <sup>235</sup>U mass enriched corresponding to a typical PWR rod enrichment used in experiments. The compensation control rods are placed at their critical position. 10<sup>9</sup> neutron histories are simulated.

#### 3.1.2 Sensitivity studies

Numerically optimizing core parameters will use several different ways. However, it seemed necessary to start the old fashion way, with the basic physics observations, like a “reload design engineer” [2]. This preliminary way included the following sensitivity studies. The sensitivity of target parameters is developed according to either the absorbers insertion in the core or the burnup of each fuel elements. The sensitivity to absorbers insertion is applied to compensation and power steering control rods. This case corresponds to Beginning Of Cycle conditions. The sensitivity to local fuel elements burnup considers several configurations. This case corresponds to Middle Of Cycle conditions. These studies will also contribute to a better understanding and justification of the results provided by numerical methods, some of which may be Physics blind.

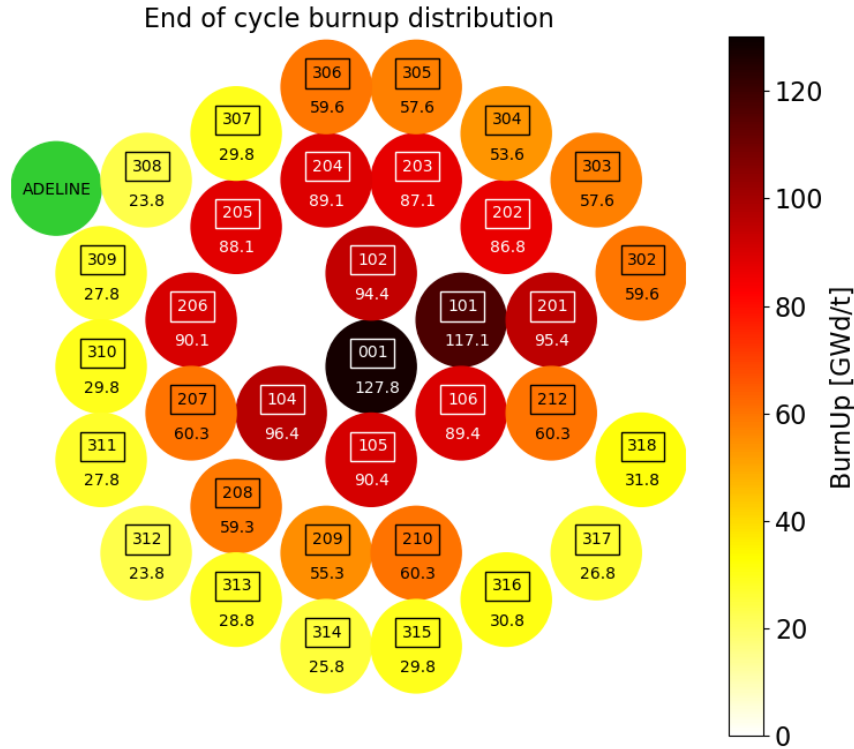
### 3.2 Genetic algorithm

In order to find the optimal core configuration, a genetic algorithm was used, considering it provides satisfactory results for reload problems [8,11]. The general principle, the operators and the parameters of the algorithm, as well as the fitness function are presented below.

#### 3.2.1 General principle

Genetic algorithms are used to solve optimization problems by applying biologically inspired processes based on the theory of evolution by natural selection [25]. Biological terminology is traditionally used to describe these algorithms. An individual is a possible solution to the optimization problem. A group of individuals constitutes a population. Each individual is unique and characterized by its genotype, which contains the properties, i.e. the genes, of this individual. Genetic algorithms converge toward better solutions at each iteration using the principles of inheritance and adaptation. Inheritance is the transmission of the properties to offspring while adaptation characterizes the fitness of an individual to its environment. At each iteration of the algorithm, the population is called a generation.

The algorithm starts by randomly selecting individuals to create the initial population. The fitness of each individual in this population is evaluated through a fitness function that depends on the optimization problem. The individuals with the highest fitness values are then selected from the population. A new population is generated from these selected individuals through genotypes



**Fig. 2.** Burnup map of the JHR core at the End Of a typical equilibrium Cycle and location of the ADELINe test device. The fuel cell location number is framed above while the burnup of the fuel element is specified below.

crossover and mutation. During the crossover operation, the genotype of the selected individuals are combined to generate new offspring, while during the mutation operation, the genotype of the offspring is randomly modified. The next iteration of the algorithm uses the population made from the offspring. This new generation then goes through the evaluation, selection, crossover, and mutation operations. The algorithm goes on until the desired number of generations are reached.

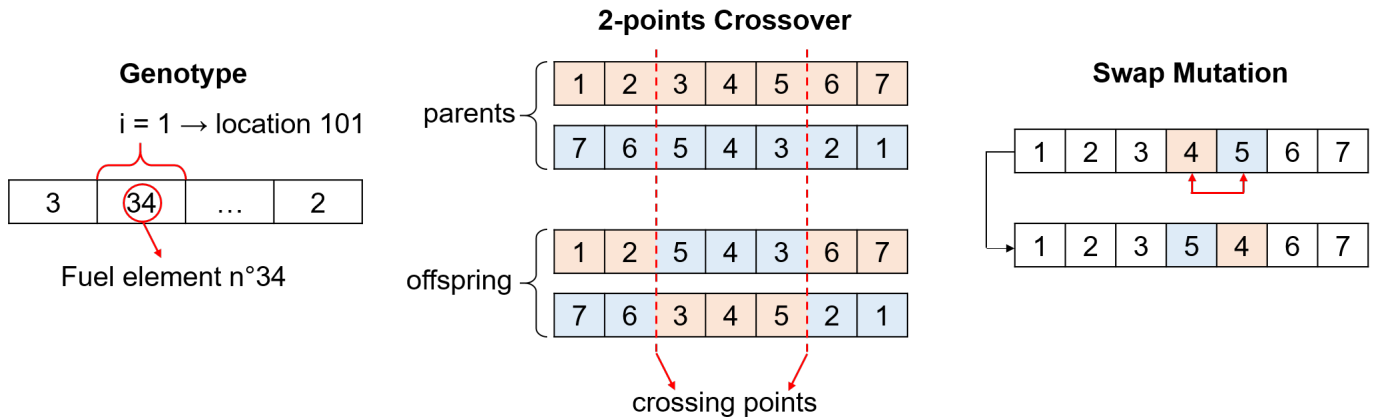
For this study, the open-source Python library PyGAD was used [26]. PyGAD allows building genetic algorithms for both single-objective and multi-objective optimization problems. Different types of selection, crossover, and mutation operators are implemented in PyGAD.

### 3.2.2 Genetic algorithm operators and parameters

In this study, a genetic algorithm built with the PyGAD library was used to converge toward optimal core configurations. In this genetic algorithm, different core configurations are represented each by a unique genotype. Each of the 24 burned and 10 fresh fuel elements is numbered from 1 to 34 and can be inserted in any of the 34 fuel cells depicted in Figure 2. These numbers are the genes of the individuals. Each individual owns 34 genes. The genotype is a list of the fuel elements numbers where the

indices of the numbers correspond to the fuel cell locations defined in Figure 2 (Fig. 3). For example, the genotype [3, 34, ..., 2] means that location 001 in the core is filled by fuel element numbered 3, location 101 is filled by fuel element numbered 34 and location 318 is filled by fuel element numbered 2. Duplicate genes are not allowed: one fuel element cannot be placed at two locations at the same time.

The parents are selected through the tournament selection technique. Several tournaments among randomly selected individuals in the population are made and the individuals with the best fitness are selected. The selected individuals, i.e. the parents, are not kept in the new generation. A two-points crossover operation technique [11] is performed as follows: the genes between two randomly selected points in the parent's genotype are swapped to create two children as shown in Figure 3. The used mutation operation is the swap mutation where two randomly selected genes are swapped as shown in Figure 3. In other words, the location of two random fuel elements are interchanged in the core. The mutation rate is fixed at 5%. The best solution in the population is kept in the next generation. It means that the fitness of the best solution in the population will only remain stable or increase after each generation.



**Fig. 3.** Genotype and operators of the genetic algorithm.

The size of the population is fixed to 20 since the size of this present reload problem is smaller than reload problems in other published works where the population size was fixed from 40 to 200 [11]. The number of parents mating is fixed to 10. It means that 10 solutions in the population will be selected as parents. It has been observed that less parents mating increased the execution time of the algorithm and lead to less optimal core configurations. The number of generations are fixed at 500, where the algorithm seemed to have converged over an optimal core configuration.

### 3.2.3 Fitness function

The best solution to this reload problem consists of a core configuration that maximizes  $K_{\text{eff}}$  and  $P_{\text{exp}}$  and minimizes  $F3D$ . The fitness function is constructed with the weighting method as follows (Eq. 1):

$$F = W_1(K_{\text{eff}} - 1) + W_2P_{\text{exp}} + W_3(2.76 - F3D) \quad (1)$$

where  $W_1$ ,  $W_2$  and  $W_3$  are the weights factors of  $K_{\text{eff}}$ ,  $P_{\text{exp}}$  and  $F3D$  respectively.  $K_{\text{eff}}$  and the  $F3D$  are constrained:  $K_{\text{eff}}$  should be larger than 1 to hold the cycle duration and safety limits impose a  $F3D$  lower than 2.76 for the JHR. The weights influence the impact of the parameters on the fitness value. Adjustments were made to find weights that allow each of the three parameters to have an equal impact on the fitness value. The following weights were chosen:  $W_1 = 500$ ,  $W_2 = 0.03$ , and  $W_3 = 50$ .

## 3.3 Convolutional neural network

In order to predict the three neutronics parameters  $K_{\text{eff}}$ ,  $P_{\text{exp}}$  and  $F3D$  for a particular core configuration, a surrogate model is used. This surrogate model is a convolutional neural network. The general principle, the implementation and parameters of the model, and the training and validation procedures are presented below.

### 3.3.1 General principle

A convolutional neural network is a type of neural network that learns features of an image through filter optimization.

It is traditionally used for image processing and computer vision [3,27]. In a typical neural network, each neuron in the input layer is fully connected to the ones in the hidden layers, whereas, in a convolutional neural network, the neurons in the hidden layers are only connected to a small part of the neurons in the input layer. In other words, in a convolutional neural network each neuron in the hidden layers represents only a small region of the input image. Using a convolutional neural network for image processing allows reducing the number of weights that would have been used in a classical neural network. Thus, overfitting is prevented and training is easier. Furthermore, because convolutional neural networks detect features in an image, the geometric structure of the input image is taken into account by the network to do the final prediction.

A convolutional neural network is made up of different layers presented in Figure 4. In input, a convolutional neural network takes a tensor with shape [number of images, image height, image width, image depth]. These inputs then pass through a convolution layer to become feature maps. In a convolution layer, input images are convoluted with filters: the filters are slid across the image and all pixels of the covered region are summed and weighted by the filter. The feature map obtained after a convolution exactly reflects where an input image is being most similar to the filter. The feature map is then activated with an activation function to introduce non-linear properties in the model. A Rectified Linear Unit (ReLU) activation function is commonly used in a convolutional neural network. This function is defined as follows:  $\text{ReLU}(x) = \max(0, x)$ . After convolution and activation layers, the feature map then passes through a pooling layer. Pooling is a down sampling operation and is used to reduce the dimensionality of feature maps. Small regions of pixels in the feature maps are condensed into a single pixel. The dimensionality of input images is thus reduced by passing through the successive layers of the model. The complexity of learned features increases after each layer: first layers detect simple features whereas last layers detect complex shapes. In a convolutional neural network, each neuron is a pixel of a feature map and the weights the network is learning are the values in the

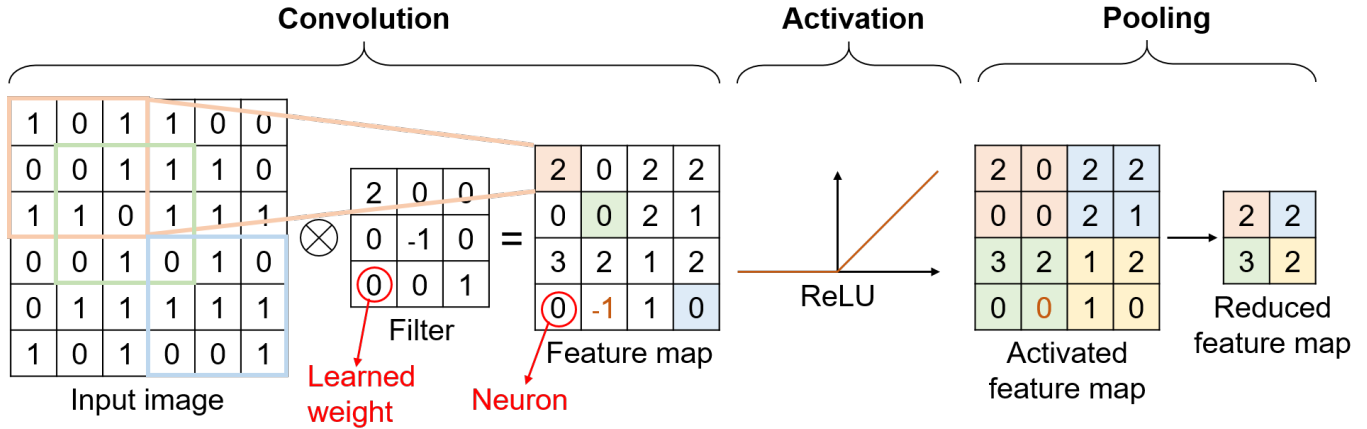


Fig. 4. Principle of a convolutional neural network.

filters. After several convolution, activation and pooling layers, the feature maps are flattened. Dense fully connected layers are placed before the output layer. The shape and activation of the output layer depend on the problem.

During the learning process, the network updates the weights by passing the training dataset through the network at each iteration or so called epoch. During an epoch, a loss function that returns the difference between predicted and target value is minimized with an optimizer. This optimizer adjusts the weights and learning rate to reduce the loss of the model.

For this study, the open-source Python library Keras [28] was used within the open-source framework TensorFlow [29]. Keras allows building standard and convolutional neural networks since commonly used neural network blocks such as layers, activation functions, and optimizers are implemented in it.

### 3.3.2 Network architecture

In this study, a convolutional neural network was built to predict the continuous variables  $K_{\text{eff}}$ ,  $P_{\text{exp}}$ , and  $F3D$  from burnup maps of the core. This is a multiple output regression problem that takes a matrix in input. The burnup maps are images of the core where a fuel element is represented by one pixel whose color depends on the burnup of the fuel element. The pixels are placed to be representative of the core geometry (Fig. 5). The size of the images, or so called burnup maps, is  $13 \times 13$ . The outputs are three continuous scalars: normalized  $K_{\text{eff}}$ ,  $P_{\text{exp}}$ , and  $F3D$ .

The architecture of the network is presented in Figure 6. The network takes in input tensors which shape is  $[13, 13, 1]$ . Then, there is a convolution layer with 34 filters of size  $3 \times 3$  each activated with the ReLU function. A pooling layer downscaling by a factor of 2 is placed after this convolution layer. Then, another ReLU activated convolution layer with 68 filters of size  $3 \times 3$  is placed. After a flattening layer, there is a dense layer containing 34 neurons. These neurons are again ReLU activated. There are three output layers of one neuron each that are fully connected to the ones in the dense layer and not activated as the activation function is linear.

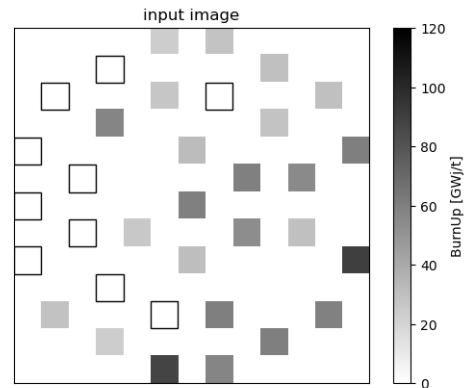


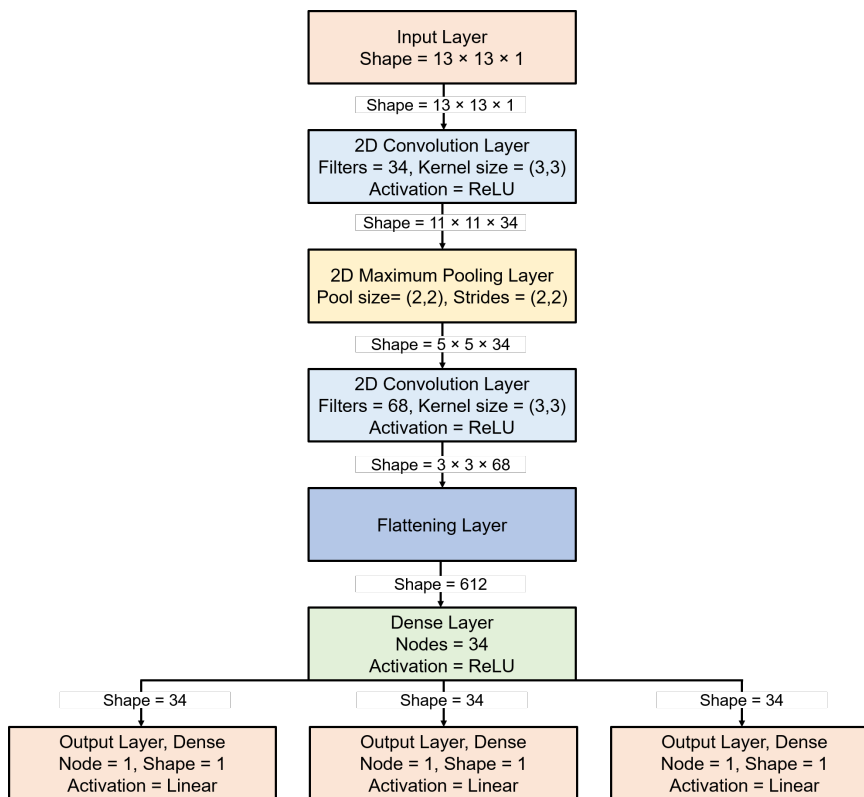
Fig. 5. Input image.

The Adam optimizer was chosen. It is a stochastic gradient descent method where first and second order moments are adaptively estimated [28]. It is traditionally used for machine learning, as it is computationally efficient. The Mean Squared Error (MSE) loss function was chosen (Eq. 2). This is a standard loss function for regression problems that returns the squared difference between target and predicted values [30].

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N \left( y_i^{\text{targeted}} - y_i^{\text{predicted}} \right)^2. \quad (2)$$

### 3.3.3 Training, validation and evaluation

In order to train the model and evaluate its performances, the dataset is divided in three groups: training, validation and evaluation. During the learning process, only the training data passes through the network. At each epoch, in order to watch for overfitting, the network is “validated” i.e. the network predicts the outputs of the validation data with temporary weights and compares them with the targeted outputs. Once the network parameters



**Fig. 6.** Chosen convolutional neural network architecture.

are converged after several epochs, the model is evaluated: the network predicts the outputs of the evaluation data and compares them with the targeted outputs.

The total computation power available for this study corresponds to 250 TRIPOLI-4<sup>®</sup> calculations belonging to 5 different families. Each family corresponds to a different BOC fuel equilibrium configuration each following a different out-in reload process. There are therefore 5 sets of 34 fuel elements that have each a given burnup. For each set, 50 different BOC core configurations were created by placing each of 34 fuel elements of the set at a random location in the core. The size of the dataset available for training, validation and evaluation is thus 250. The 5 families are generated from a first cycle JHR core starting with 34 fresh fuel elements and operating with regular out-in reloads. The following assumptions determine each of the 5 specific sets: the first set corresponds to BOC n°6 with 10 fresh fuel elements. The second corresponds to BOC n°7 with 12 fresh fuel elements. The third, fourth and fifth correspond to BOC n°8 with respectively 8, 10 and 12 fresh fuel elements. For each set, the burnup distribution is determined with core burnup calculations. Depending on the situation, there are more or less depleted fuel elements. The model was trained with these 5 different sets in order to predict the three parameters of interest ( $K_{\text{eff}}$ ,  $P_{\text{exp}}$ , and  $F3D$ ) in the largest range of situations. The global algorithm can thus be used to generate any reload from any EOC situation.

For training and validation, 90% of the data is randomly selected among which 10% is dedicated to validation. The remaining 10% is used for the evaluation of the model. The model is trained on 100 epochs.

The training data were obtained through TRIPOLI-4<sup>®</sup> core physics calculations. For each calculation, the core is filled with 34 initially high-enriched fuel elements each with a burnup adapted to the desired configuration. The reflector is filled with beryllium and dummy experiments except for the ADELINe test device. This device is placed next to the core and contains one experimental rod. The experimental rod is fresh and 5%  $^{235}\text{U}$  mass enriched. The calculations are made at the Beginning Of the Cycle. The control rods dedicated to reactivity loss compensation are placed at their Beginning Of Cycle position.  $2.5 \times 10^8$  histories are simulated. One calculation takes on average 1.8 hours on 256 CPU running @ 2.45 GHz.  $K_{\text{eff}}$ ,  $P_{\text{exp}}$  and  $F3D$  are automatically extracted from the calculation results with a dedicated Python script. These calculations provide the value of these three parameters corresponding to as many core configurations.

In order to enhance the performances of the convolutional neural network, all training, validation and evaluation data are normalized. The inputs, that is to say the burnup maps, are divided by 120 since no fuel element with a burnup higher than 120 GWd/t will be loaded in the core. The outputs are normalized with the following method:  $y_{\text{norm}} = \frac{y - \min}{\max - \min}$ . The value of the maximum and minimum selected for each parameter are gathered in [Table 1](#).

**Table 1.** Minimum and maximum for the normalization of the three parameters.

	$K_{\text{eff}}$	$P_{\text{exp}}$ [W/cm]	$F3D$
Min	1.0	1000	1.0
Max	1.1	1800	2.8

To evaluate the model, the Mean Absolute Percentage Error (MAPE) was observed [30]. It is a measure of the prediction accuracy and it is defined as follows (Eq. 3):

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i^{\text{targeted}} - y_i^{\text{predicted}}|}{y_i^{\text{targeted}}} \times 100. \quad (3)$$

### 3.4 Global procedure to find the optimal core configuration

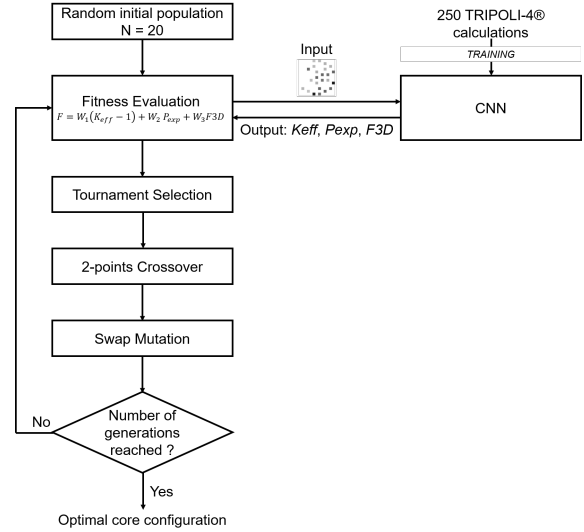
Eventually, in order to find the optimal core configuration corresponding to the defined criteria, the convolutional neural network is coupled with the genetic algorithm in a code written in Python. The flowchart presented in Figure 7 shows the different steps of the algorithm:

1. the convolutional neural network that serves as surrogate model to obtain  $K_{\text{eff}}$ ,  $P_{\text{exp}}$  and  $F3D$  from burnup maps is trained and evaluated.
2. The initial population is generated by randomly choosing possible core configurations among all possible configurations.
3. All core configurations present in the current population are evaluated through the fitness function. The fitness function calls the surrogate model to predict  $K_{\text{eff}}$ ,  $P_{\text{exp}}$  and  $F3D$  from the core configurations.
4. Parents are selected through tournament selection operator.
5. The offspring population is created by the two-points crossover and swap mutation operators.
6. The algorithm goes on and starts over at step n°2.
7. The algorithm stops when the number of generation is reached.

## 4 Results and discussion

### 4.1 Preliminary neutronics study

The preliminary neutronics study showed different interesting Physics results. Table 2 shows the results for the compensation and power steering control rods sensitivity study. It gives the linear power in the experimental rod and the percentage of contribution to experimental performance of the ADELINÉ test device compared to a reference case. Table 2 shows that extracting a compensation control rod in front of the test device and inserting one on the core side opposite to the test device increases the experimental performance by 23% in the affected test

**Fig. 7.** Flow diagram of the algorithm.

device. It also shows that extracting a power steering control rod within its operating range in front of the test device and inserting the other ones increases the experimental performance by 2.7% in the affected test device. In both cases, a flux tilt is performed which explains the increase of the linear power in the experimental rod.

Table 3 shows the results for the burnup sensitivity study and gives the linear power in the experimental rod and the percentage of contribution to experimental performance in the ADELINÉ test device compared to a reference case. It shows that placing fresh fuel elements in front of the test device increases its experimental performance by 13%, whereas placing depleted fuel elements decreases it by 13%. Moreover, fresh fuel elements located near the test device (locations 205 and 206) will increase experimental performance whereas putting them in the center and on the side opposite to the device (locations 101 and 106) will decrease it. On the contrary, depleted fuel elements located near the test device will decrease experimental performance whereas putting them in the center and on the side opposite to the device will increase it. Summarizing these results, it appears that placing fresh fuel elements in front of the test device and depleted fuel elements in the center and on the side opposite to the device will maximize the linear power in the experimental rod. It is confirmed by the fact that placing fresh fuel elements right in front of the device and depleted ones in the center increases the performance by 23%. It should be noted that other locations for fresh or depleted fuel elements have been explored but only the ones that maximize or minimize experimental performance have been kept in Table 3.

### 4.2 Surrogate model convergence and results

For all the 250 core configurations calculated with TRIPOLI-4<sup>®</sup> for the training, validation, and evaluation

**Table 2.** Linear power and contribution to experimental performance of the ADELINe test device for several core configurations.

Core configuration	Maximal linear power [W/cm]	Contribution [%]
Compensation rods: 309 inserted and 317 extracted	1558	Reference
Compensation rods: 309 extracted and 317 inserted	1912	+23%
Power steering rods: all placed at 40 cm	1891	Reference
Power steering rods: 206 at 50 cm and others at 35 cm	1942	+2.7%

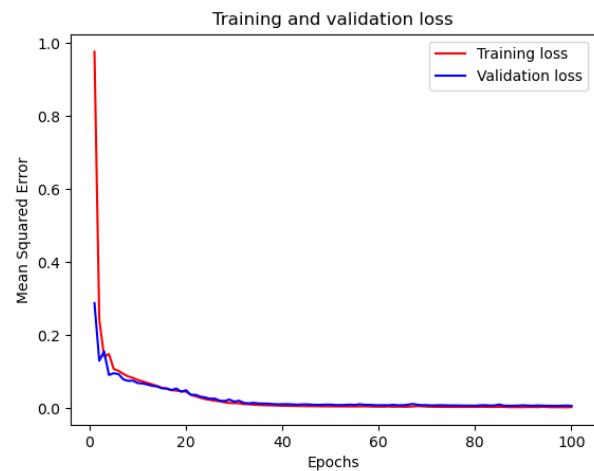
**Table 3.** Linear power and contribution to experimental performance of the ADELINe test device for several core configurations.

Core configuration	Maximal linear power [W/cm]	Contribution [%]
All 50.7 GWd/t	1548	Reference
308 and 309: 0 GWd/t others: 50.7 GWd/t	1756	+13%
308 and 309: 120 GWd/t others: 50.7 GWd/t	1353	-13%
205 and 206: 0 GWd/t others: 50.7 GWd/t	1667	+7.7%
101 and 106: 0 GWd/t others: 50.7 GWd/t	1433	-7.4%
205 and 206: 100 GWd/t others: 50.7 GWd/t	1458	-5.8%
101 and 106: 100 GWd/t others: 50.7 GWd/t	1617	+4.5%
308 and 309: 0 GWd/t 001, 101, 106 : 100 GWd/t others: 50.7 GWd/t	1876	+21%

**Table 4.** Different network architectures loss.

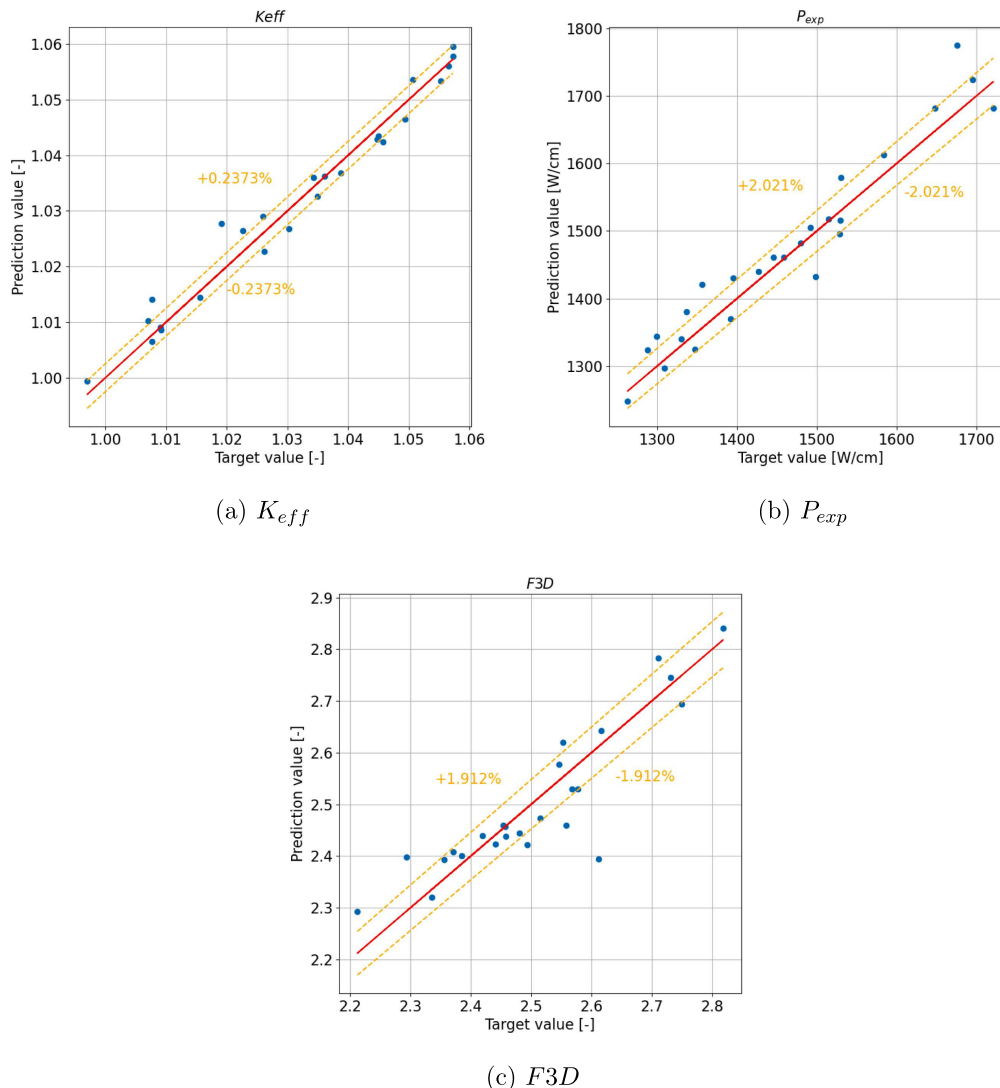
		Training loss	Validation loss
One 2D convolution layer 34 filters	$K_{\text{eff}}$	0.045E-3	0.41E-3
	$P_{\text{exp}}$	0.19E-3	2.0E-3
	$F3D$	0.34E-3	2.6E-3
Chosen network configuration	$K_{\text{eff}}$	0.055E-3	0.26E-3
	$P_{\text{exp}}$	0.19E-3	2.3E-3
	$F3D$	0.42E-3	1.7E-3
Two 2D convolution layers 34 and 68 filters	$K_{\text{eff}}$	0.34E-3	1.3E-3
	$P_{\text{exp}}$	0.15E-3	4.4E-3
	$F3D$	0.33E-3	1.5E-3
Six 2D convolution layers First layer: 34 filters Others: 68 filters	$K_{\text{eff}}$	0.34E-3	1.3E-3
	$P_{\text{exp}}$	0.15E-3	4.4E-3
	$F3D$	0.33E-3	1.5E-3

of the surrogate model,  $K_{\text{eff}}$  is between [0.993, 1.065] and on average the uncertainty at one sigma is 0.007%;  $P_{\text{exp}}$  is between [1181, 1924] W/cm and on average the uncertainty at one sigma is 1.093%;  $F3D$  is between [2.14, 2.84] and on average the uncertainty at one sigma is 1.022%. It can be observed that different core configurations result in different values for these three parameters. The convolutional neural network was trained with 203 TRIPOLI-4<sup>®</sup> calculations. The training takes 12 s for 100 epochs with a CPU running @ 3.60 GHz.

**Fig. 8.** Training and validation loss.

The number of filters and their size, and the number of layers have been chosen so that the predicted parameters of the test data are the closest to the target values. Table 4 presents the MSE loss (Eq. 2) for the three parameters of interest for different network configurations. With less filters or layers, the network is underfitting: it is not able to predict the parameters correctly. Indeed, Table 4 shows that the validation loss is greater for less layers or filters. On the other hand, if the network is more complex, with a lot of layers or a lot of filters, it becomes overfitting. It means that the network is able to well predict the training data but not the validation data. Indeed, for more layers or filters, Table 4 shows that the loss is low for training data but greater for validation data. Thus, it is a matter of finding a fair middle point between underfitting and overfitting by choosing the number of layers and filters of the network. This involves finding the network configuration for which the loss of the validation data is minimal, as it is the case for the chosen configuration.

Figure 8 shows the training and validation loss, that is the mean squared error, as a function of epochs. Both



**Fig. 9.** Training results of the convolutional neural network.

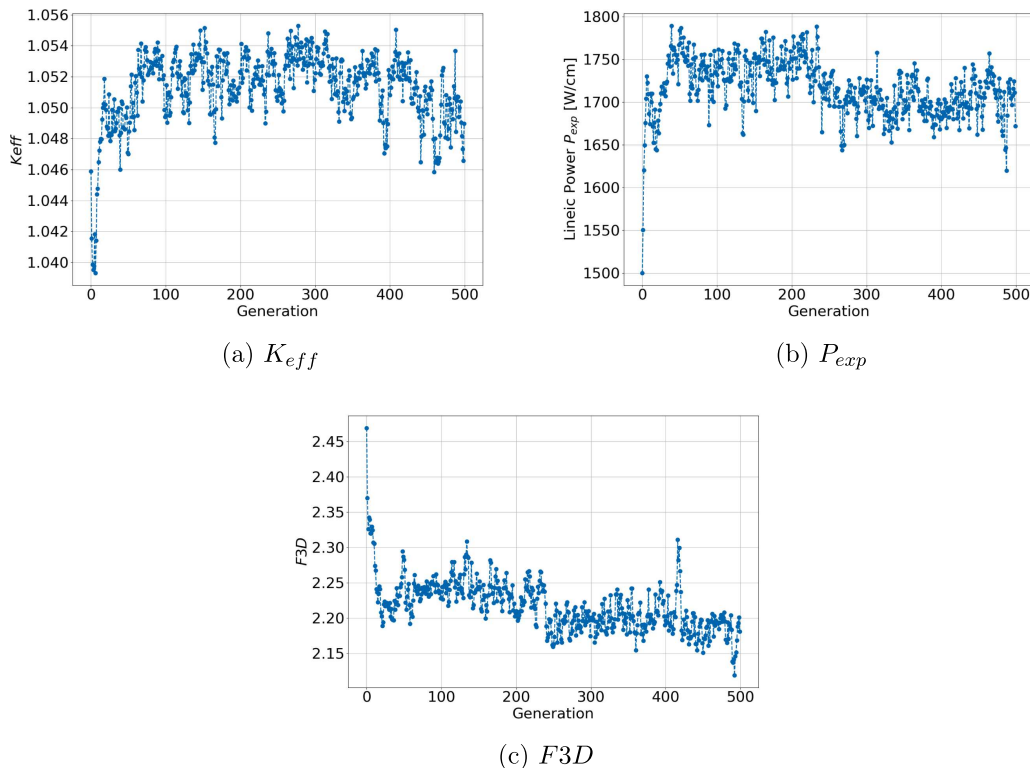
training and validation loss decrease and converge toward zero. It means that the algorithm converges and that the network manages to learn from the training data: the model is not underfitting. The fact that the validation loss is above the training loss means that the network will predict training data better than validation data. Nevertheless, the gap between training and validation loss remains small so the network will be able to efficiently predict new data. Moreover, the gap between training and validation loss remains constant and is not increasing. It means that the model is not overfitting, thus it will predict new data correctly.

The model was evaluated on 25 core configurations randomly selected among the 250 initial calculations. These configurations were not used for the training and validation of the model. The MAPE of  $K_{eff}$  is 0.237%; that of  $P_{exp}$  is 2.02%; and that of  $F3D$  is 1.91%. Therefore, the model is considered a good predictor of the three parameters of interest (Fig. 9).

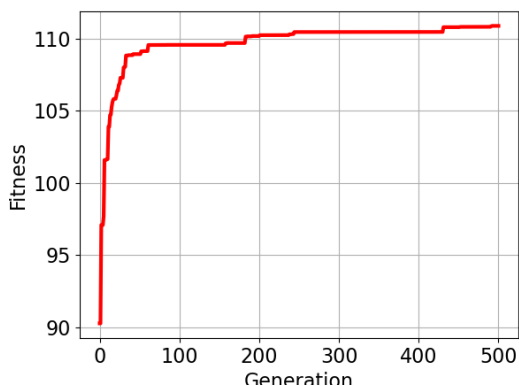
### 4.3 Genetic algorithm results: optimal core configuration

The genetic algorithm explored  $10^4$  new core configurations over 500 generations. For each of these configurations,  $K_{eff}$ ,  $P_{exp}$  and  $F3D$  were predicted with the convolutional neural network. Figure 10 shows the evolution of the average over all the population of the three parameters  $K_{eff}$ ,  $P_{exp}$  and  $F3D$  depending on the generation number. The  $K_{eff}$  parameter has globally an increasing tendency especially during the first 25 generations (Fig. 10a). The  $P_{exp}$  parameter has also globally an increasing tendency especially during the first 50 generations (Fig. 10b). On the other hand, the  $F3D$  parameter has a decreasing tendency especially during the first 20 generations (Fig. 10c). It can be observed that the genetic algorithm achieves the goal of maximizing  $K_{eff}$  and  $P_{exp}$  and minimizing  $F3D$ .

Figure 11 presents the fitness value of the best solution in the population depending on the generation num-



**Fig. 10.** Evolution of the parameters as a function of the generations.

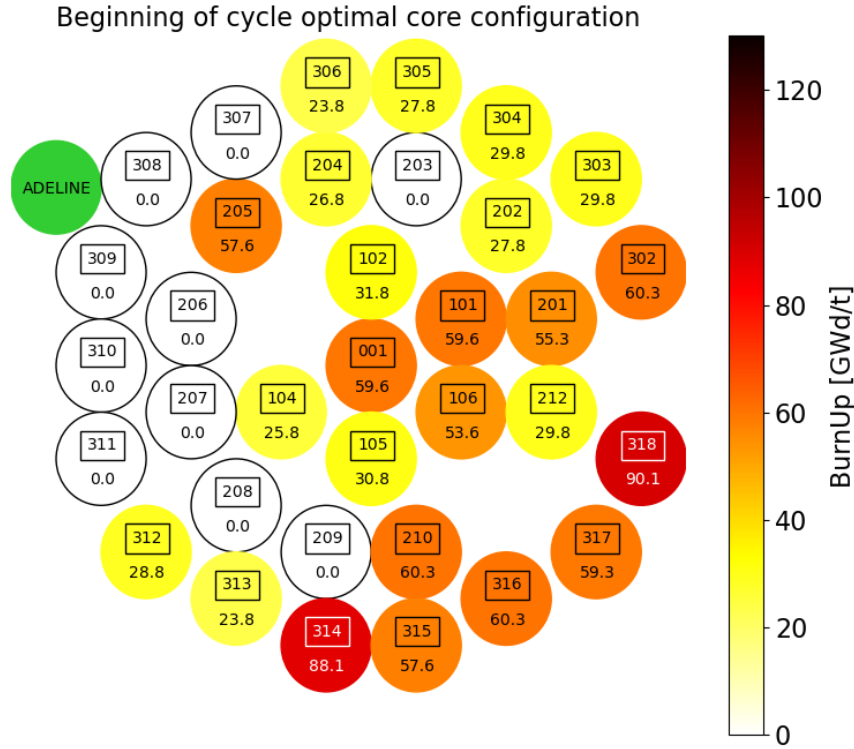


**Fig. 11.** Fitness as a function of the generations.

ber. The fitness value increases with the number of generations. The increase is initially strong for the first 50 generations then weaker for the following generations. This is consistent with the fact that  $K_{eff}$  and  $P_{exp}$  are strongly increasing during the first 50 generations while  $F3D$  is strongly decreasing. The fitness value converges toward a value of 110.9. This figure shows that the genetic algorithm converges toward an optimal core configuration that has the highest fitness value. Since the algorithm has converged and the fitness value is not increasing much anymore, the algorithm is stopped after 500 generations.

Figure 12 shows the optimal core configuration obtained with the genetic algorithm coupled with the convolutional neural network. This configuration is the one that has the highest fitness value in the population at the five hundredth generation. The convolutional neural network predict  $K_{eff} = 1.051$ ,  $P_{exp} = 1739$ , and  $F3D = 2.09$ . The 10 fresh fuel elements, that have a burnup value of zero, are placed in the following locations: 203, 206, 207, 208, 209, 307, 308, 309, 310, and 311. The fresh fuel elements are located at the core periphery, to the left, next to the test device. The moderately depleted fuel elements are located in the core center and in the periphery to the bottom right, on the side opposed to the test device, where depleted fuel elements are also located. Moderately depleted fuel elements are placed on the upper right. This optimal core configuration is consistent with the preliminary neutronics study for which results are commented above. Indeed, fresh fuel elements are placed in front of the test device and depleted ones in the core center on the opposite side in order to maximize  $P_{exp}$ .

Eventually, a TRIPOLI-4<sup>®</sup> calculation was made to verify the three parameters  $K_{eff}$ ,  $P_{exp}$  and  $F3D$  of the optimized core configuration. Table 5 shows the results and the absolute percentage errors between the values predicted by the convolutional neural network and TRIPOLI-4<sup>®</sup> calculation. It can be observed that the optimal core configuration has a  $F3D$  below the safety limit and a  $K_{eff}$  that is maximized.



**Fig. 12.** Optimal core configuration at the beginning of the next cycle. The fuel cell location number is framed above while the burnup of the fuel element is specified below.

**Table 5.** Optimal core configuration parameters.

Parameters	CNN	TRIPOLI-4	Absolute percentage error [%]
$K_{\text{eff}}$	1.051	1.047 $\sigma$ [%] = 0.00646	0.395
$P_{\text{exp}}$ [W/cm]	1739	1786 $\sigma$ [%] = 1.02	2.63
$F3D$	2.09	2.34 $\sigma$ [%] = 0.519	10.4

## 5 Conclusion

This study is dedicated to the fuel reloads of the Jules Horowitz Reactor. The goal was to find the optimal core configuration for the next cycle by maximizing the reactivity  $K_{\text{eff}}$ , the linear power  $P_{\text{exp}}$  in a test device, and by minimizing the safety parameter  $F3D$ . The idea is to replace the most burned fuel elements considered at an equilibrium composition with 10 fresh fuel elements. The optimal configuration consists in finding the best location for each fuel element in the core.

A genetic algorithm was used to converge toward the optimal core configuration by selecting the fittest configurations and creating new ones with the two-points

crossover and swap mutation techniques. The higher  $K_{\text{eff}}$  and  $P_{\text{exp}}$  are and the lower  $F3D$  is, the higher the fitness will be. The genetic algorithm parameters were fixed with the help of previous studies and adapted to the current problem. A population size of 20, a mutation rate of 0.05, and a number of 10 mating parents were chosen. After 500 generations, the algorithm converged.

For each core configuration explored by the genetic algorithm, the three neutronics parameters  $K_{\text{eff}}$ ,  $P_{\text{exp}}$  and  $F3D$  were obtained with a convolutional neural network. The network was trained and evaluated from 250 initial TRIPOLI-4<sup>®</sup> calculations. In input, the network takes a burnup map of the core and then predicts the three parameters of interest. The network predictivity is efficient because the associated MAPE uncertainty is 0.237% for  $K_{\text{eff}}$ , 2.02% for  $P_{\text{exp}}$ , and 1.91% for  $F3D$ .

It can be concluded that a genetic algorithm coupled with a convolutional neural network can be efficiently used to find an optimal core configuration where the  $K_{\text{eff}}$  is maximized, where the linear power in the test device is maximized and where the  $F3D$  is below the safety limit. Moreover, the optimal core configuration is obtained in  $\sim 7.5$  minutes considering that the initial 250 TRIPOLI-4<sup>®</sup> core calculations have already been performed and the convolutional neural network has been trained.

TRIPOLI-4<sup>®</sup> core calculations have shown in the preliminary Physics analysis that the control rods position along the cycle has an impact on experimental performances. However, only one control rods extraction sequence was considered in this study. Thus, a perspective of this work could consist in using a genetic algorithm coupled with a convolutional neural network to optimize the extraction sequence of the control rods dedicated to the reactivity loss compensation. The challenge would then be to find an optimal control rods extraction sequence that optimizes the use of fuel fissile content and experimental performances while ensuring that safety limits are not exceeded.

## Glossary

Acronym/Terms	Definition
ADELINÉ	Advanced Device for Experimenting up to Limits Irradiated Nuclear fuel Elements
BOC	Beginning Of Cycle
CEA	French alternative energies and Atomic Energy Commission
CNN	Convolutional Neural Network
CPU	Central Processing Unit
EOC	End Of Cycle
F3D	Power peaking Factor
JHR	Jules Horowitz Reactor
$K_{\text{eff}}$	Effective multiplication factor
GA	Genetic Algorithm
GWd/t	Gigawatt-day/tonne (unit of the burnup)
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
$P_{\text{exp}}$	Linear power in the experimental rod
PyGAD	Genetic Algorithm Python library
PWR	Pressurized Water Reactor
ReLU	Rectified Linear Unit function
TRIPOLI-4 <sup>®</sup>	Continuous-energy radiation transport Monte Carlo code

## Acknowledgments

The authors wish to acknowledge the help of Jérôme Julien for strongly encouraging this work and the JHR project of CEA/Department of Experiments Development for funding.

## Funding

This work was funded by the JHR project of CEA/Department of Experiments Development.

## Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this article.

## Data availability statement

Data associated with this article cannot be disclosed.

## Author contribution statement

All the authors were involved in the preparation of the manuscript. All the authors have read and approved the final manuscript.

## References

1. D. Iracane, P. Chaix, A. Alamo, Jules Horowitz Reactor: A high performance material testing reactor, *C. R. Phys.* **9**, 445 (2008), <https://doi.org/10.1016/j.crhy.2007.11.003>
2. P.J. Turinsky, G.T. Parks, Advances in nuclear fuel management for light water reactors, in *Advances in Nuclear Science and Technology*, edited by J. Lewins et al. (Springer US, Boston, MA, 1999), pp. 137–165, [https://doi.org/10.1007/0-306-47088-8\\_6](https://doi.org/10.1007/0-306-47088-8_6)
3. J.-P. Haton, E. Haton, M.-Ch. Haton, in *Intelligences artificielles: De la théorie à la pratique, Modèles, Applications et enjeux des IA*, Dunod, 2023, p. 23
4. Z. Li, J. Wang, M. Ding, A review on optimization methods for nuclear reactor fuel reloading analysis, *Nucl. Eng. Des.* **397**, 111950 (2022), <https://doi.org/10.1016/j.nucengdes.2022.111950>, <https://www.sciencedirect.com/science/article/pii/S0029549322003016>
5. P.W. Poon, G.T. Parks, Application of genetic algorithms to in-core nuclear fuel management optimization, in *Proc. Joint Int. Conf. Mathematical Methods and Supercomputing in Nuclear Applications, Karlsruhe* (1993), vol. 1, p. 777
6. J.L.C. Chapot, F. Carvalho Da Silva, R. Schirru, A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem, *Ann. Nucl. Energy* **26**, 641 (1999), [https://doi.org/10.1016/S0306-4549\(98\)00078-4](https://doi.org/10.1016/S0306-4549(98)00078-4), <https://www.sciencedirect.com/science/article/pii/S0306454998000784>
7. B.Q. Do, L.P. Nguyen, Application of a genetic algorithm to the fuel reload optimization for a research reactor, *Appl. Math. Comput.* **187**, 977 (2007), <https://doi.org/10.1016/j.amc.2006.09.024>, <https://www.sciencedirect.com/science/article/pii/S0096300306012343>
8. A.V. Sobolev, A.S. Gazetdinov, D.S. Samokhin, Genetic algorithms for nuclear reactor fuel load and reload optimization problems, *Nucl. Energy Technol.* **3**, 231 (2017), <https://doi.org/10.1016/j.nucet.2017.07.002>, <https://www.sciencedirect.com/science/article/pii/S2452303817300602>
9. E. Israeli, E. Gilad, Novel genetic algorithm for loading pattern optimization based on core physics heuristics, *Ann. Nucl. Energy* **118**, 35 (2018), <https://doi.org/10.1016/j.anucene.2018.03.042>, <https://www.sciencedirect.com/science/article/pii/S0306454918301609>
10. N. Shaukat et al., Multiobjective core reloading pattern optimization of PARR-1 using modified genetic algorithm coupled with Monte Carlo methods, *Sci. Technol. Nucl. Install.* **2021**, 1802492 (2021), <https://doi.org/10.1155/2021/1802492>, <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/1802492>, <https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/1802492>

11. E. Chham et al., Fuel reloads optimization for TRIGA research reactor using genetic algorithm coupled with neutronic and thermal-hydraulic codes, *Prog. Nucl. Energy* **133**, 103637 (2021), <https://doi.org/10.1016/j.pnucene.2021.103637>, <https://www.sciencedirect.com/science/article/pii/S0149197021000093>
12. M. Gomez-Fernandez et al., Status of research and development of learning-based approaches in nuclear science and engineering: A review, *Nucl. Eng. Des.* **359**, 110479 (2020), <https://doi.org/10.1016/j.nucengdes.2019.110479>, <https://www.sciencedirect.com/science/article/pii/S0029549319305102>
13. H.G. Kim, S.H. Chang, B.H. Lee, Pressurized water reactor core parameter prediction using an artificial neural network, *Nucl. Sci. Eng.* **113**, 70 (1993), <https://doi.org/10.13182/NSE93-A23994>
14. K. Tirel et al., Coupling reactor design and scenario calculations: A promising method for scenario optimization, *EPJ Nucl. Sci. Technol.* **8**, 7 (2022), <https://doi.org/10.1051/epjn/2022002>
15. B. Leniau et al., A neural network approach for burn-up calculation and its application to the dynamic fuel cycle code CLASS, *Ann. Nucl. Energy* **81**, 125 (2015), <https://doi.org/10.1016/j.anucene.2015.03.035>, <https://www.sciencedirect.com/science/article/pii/S0306454915001693>
16. A. Yamamoto, Application of neural network for loading pattern screening of in-core optimization calculations, *Nucl. Technol.* **144**, 63 (2003), <https://doi.org/10.13182/NT03-A3429>
17. K. Palmi, W. Kubinski, P. Darnowski, Prediction of the evolution of the nuclear reactor core parameters using artificial neural network, *Ann. Nucl. Energy* **211**, 110891 (2025), <https://doi.org/10.1016/j.anucene.2024.110891>, <https://www.sciencedirect.com/science/article/pii/S0306454924005541>
18. H. Jang, H. Lee, Prediction of pressurized water reactor core design parameters using artificial neural network for loading pattern optimization, in *Proceedings of the Transactions of the Korean Nuclear Society Virtual Spring Meeting, Korea* (2020), pp. 9–10
19. C.G. Forrest Shriver, J. Watson, Prediction of neutronics parameters within a two-dimensional reflective PWR assembly using deep learning, *Nucl. Sci. Eng.* **195**, 626 (2021), <https://doi.org/10.1080/00295639.2020.1852021>
20. A. Erdoğan, M. Geçkinli, A PWR reload optimisation code (XCore) using artificial neural networks and genetic algorithms, *Ann. Nucl. Energy* **30**, 35 (2003), [https://doi.org/10.1016/S0306-4549\(02\)00041-5](https://doi.org/10.1016/S0306-4549(02)00041-5), <https://www.sciencedirect.com/science/article/pii/S0306454902000415>
21. C. Wan, K. Lei, Y. Li, Optimization method of fuel-reloading pattern for PWR based on the improved convolutional neural network and genetic algorithm, *Ann. Nucl. Energy* **171**, 109028 (2022), <https://doi.org/10.1016/j.anucene.2022.109028>, <https://www.sciencedirect.com/science/article/pii/S0306454922000639>
22. M.K. Butt et al., Integrating the deep learning and multi-objective genetic algorithm to the reloading pattern optimization of HPR1000 reactor core, *Nucl. Eng. Des.* **428**, 113531 (2024), <https://doi.org/10.1016/j.nucengdes.2024.113531>, <https://www.sciencedirect.com/science/article/pii/S0029549324006319>
23. S. Gaillot et al., Jules Horowitz Reactor development of an experimental loop integrating an optimized irradiation process, in *RRFM-IGORR 2016, RRFM-IGORR 2016, Berlin, Germany* (2016), <https://cea.hal.science/cea-02431790>
24. E. Brun et al., TRIPOLI-4<sup>®</sup>, CEA, EDF and AREVA reference MONTE CARLO code, *Ann. Nucl. Energy* **82**, 151 (2015), Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013, Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms, <https://doi.org/10.1016/j.anucene.2014.07.053>, <https://www.sciencedirect.com/science/article/pii/S0306454914003843>
25. J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (MIT Press, 1992)
26. A.F. Gad, Pygad: An intuitive genetic algorithm python library, in *Multimedia Tools and Applications* (2023), pp. 1–14.
27. Y. LeCun, Y. Bengio et al., Convolutional networks for images, speech, and time series, in *The Handbook of Brain Theory and Neural Networks* (1995), vol. 3361, p. 1995
28. F. Chollet et al., *Keras* (2015), <https://github.com/fchollet/keras>
29. M. Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from [tensorflow.org](https://www.tensorflow.org) (2015), <https://www.tensorflow.org/>
30. A. de Myttenaere et al., Mean absolute percentage error for regression models, *Neurocomputing* **192**, 38 (2016), *Advances in artificial neural networks, machine learning and computational intelligence*, <https://doi.org/10.1016/j.neucom.2015.12.114>, <https://www.sciencedirect.com/science/article/pii/S0925231216003325>

**Cite this article as:** Douchka Dimitrijevic, Guillaume Ritter. Experiments and fuel reloads optimization for the Jules Horowitz Reactor (JHR) using a genetic algorithm combined with a convolutional neural network, *EPJ Nuclear Sci. Technol.* **12**, 3 (2026). <https://doi.org/10.1051/epjn/2025076>