



Neural Network Acceleration of Genetic Algorithms for the Optimization of a Coupled Fast/Thermal Nuclear Experiment

John Pevey*, Vlad Sobes and Wes. J. Hines

University of Tennessee, Nuclear Engineering Department, Knoxville, TN, United States

Genetic algorithms (GA) are used to optimize the Fast Neutron Source (FNS) core fuel loading to maximize a multiobjective function. The FNS has 150 material locations that can be loaded with one of three different materials resulting in over $3E+71$ combinations. The individual designs are evaluated with computationally intensive calls to MCNP. To speed up the optimization, convolutional neural networks (CNN) are trained as surrogate models and used to produce better performing candidates that will meet the design constraints before they are sent to the costly MCNP evaluations. A major hurdle in training neural networks of all kinds is the availability of robust training data. In this application, we use the data produced by the GA as training data for the surrogate models which combine geometric features of the system to predict the objectives and constraint objectives. Utilizing the surrogate models, the accelerated algorithm produced more viable designs that significantly improved the objective function utilizing the same computational resources.

Keywords: nuclear reactor, optimization, fast neutron source, convolutional neural networks, surrogate model, machine learning

OPEN ACCESS

Edited by:

Xingang Zhao,
Oak Ridge National Laboratory (DOE),
United States

Reviewed by:

Xinyan Wang,
Massachusetts Institute of
Technology, United States
Yinan Cai,
Massachusetts Institute of
Technology, United States

*Correspondence:

John Pevey
jpevey@vols.utk.edu

Specialty section:

This article was submitted to
Nuclear Energy,
a section of the journal
Frontiers in Energy Research

Received: 11 February 2022

Accepted: 28 April 2022

Published: 17 June 2022

Citation:

Pevey J, Sobes V and Hines WJ (2022)
Neural Network Acceleration of
Genetic Algorithms for the Optimization
of a Coupled Fast/Thermal
Nuclear Experiment.
Front. Energy Res. 10:874194.
doi: 10.3389/fenrg.2022.874194

1 INTRODUCTION

The optimization of nuclear problems can be a complex task often with multiple competing objectives and constraints. There is much research into the optimization of various aspects of nuclear reactors such as the initial design (Gougar, et al., 2010), fuel shuffling (Zhao, et al., 1998; Chapot, Da Silva and Schirru 1999), and shielding (Kim and Moon 2010; Tunes, De Oliveira and Schön 2017) optimizations. Due to the non-linear nature of these problems, optimization algorithms such as evolutionary and simulated annealing algorithms are often used. These methods do not guarantee that the optimal solution is found but can, with sufficient computational resources, clever heuristics, and the application of expert knowledge, often find solutions which are near-optimal.

When an optimization is of a function that is prohibitively expensive (such as solving the neutron transport equation), a surrogate model (Sobester, Forrester and Keane 2008) is produced and optimized instead. Surrogate models generally trade the computation expense of the original function for less accurate, but less expensive functions. The surrogate model is optimized instead of the original function. These can be as simple as linear functions or as complex as deep neural networks. In nuclear optimizations, surrogate models have been built to approximate expensive functions such as finite-element structural analysis (Prabhu, et al., 2020), computational

fluid dynamics (Hanna, et al., 2020), and neutron transport (Faria and Pereira, 2003; Hogle 2012; El-Sefy et al., 2021; Sobes, et al., 2021) related objectives.

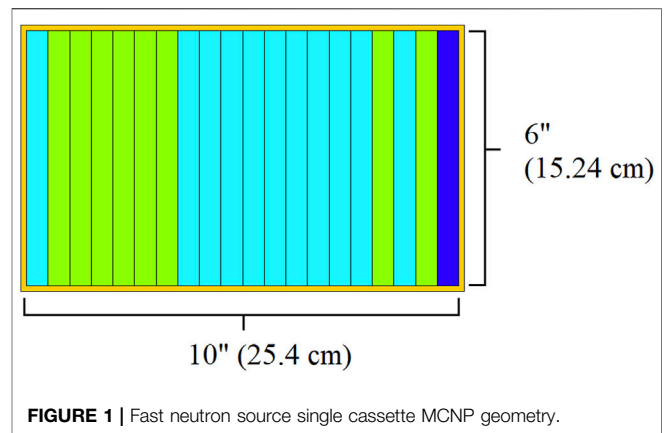
In this paper, an optimization of the Fast Neutron Source at the University of Tennessee was performed using objectives calculated by solving the neutron transport equation. A surrogate model for these objectives is presented, and a genetic algorithm with and without acceleration using that surrogate model are compared. The following sections include overviews of both the Fast Neutron Source and of the neural network architecture used as the surrogate models. Two genetic algorithm optimizations are presented, the first uses the Non-dominated Sorting Algorithm-II and Monte Carlo N-Particle transport code (MCNP) (Goorley, et al., 2012) to solve the objective and constraint functions. The second optimization uses surrogate models with the NSGA-II algorithm, after which the individuals in the Pareto Front are evaluated with MCNP.

1.1 Fast Neutron Source

The Fast Neutron Source (FNS) (Pevey, et al., 2020) will be a platform for sub-critical integral cross section experiments at the University of Tennessee. It will be driven by 2.5 MeV neutrons produced by a deuterium-deuterium (DD) neutron generator and feature a flexible construction which will produce sub-critical benchmark experiments targeting specific nuclear data needs of next generation reactors.

The reduction of the uncertainty on next-generation reactor designs is a need for the expected rapid deployment of next generation reactors. Nuclear data uncertainty is propagated to all nuclear-related figures of merit of reactors such as k-eff, void, temperature and power reactivity coefficients and reactivity worth's. In a recent assessment of the nuclear data needed for advanced reactors, it was found that for several next-generation reactors need better resolved nuclear data (Bostelmann, et al., 2021). For example, in a sensitivity analysis of the Advanced Burner Reactor 1000 MWth Reference Concept, it was shown that the uncertainty in important nuclear characteristics is driven by uncertainties in uranium, plutonium, iron and sodium. In this concept, the uncertainty on k-eff, temperature coefficients of reactivity and Na void worth were 0.900%, 8.397% and 13.483%, respectively. These uncertainties require added margin in designs and can lead to less-than-optimal designs to account for these uncertainties.

Systematic integral data assimilation can be done to decrease these uncertainties. In this type of analysis, a suite of known benchmark models is collected, and a sensitivity analysis is completed for each. The known experimental values (and associated biases in the computational models) are then used to further decrease the uncertainty in the relevant quantities of interest. The FNS will be a source of these types of benchmarks which targets the reduction of uncertainties in neutronics calculation due to nuclear data uncertainty. The goal of a given configuration of the FNS then is to maximize the relevance of the experiment to some target advanced reactor concept and to maximize the total flux



produced by the configuration to reduce the required FNS run-times.

The most basic geometry unit of the FNS are six" x six" x 0.5" plates which can be one of three different materials in this study. Up to 20 of these plates are combined into aluminum cassettes (See **Figure 1**). Twenty-five of these cassettes are then combined into a 5 x 5 array called a zone (See **Figure 2**). There are three zones in the FNS, as seen in **Figure 2A**. In each zone, due to rotational symmetry, there are up to six unique cassette patterns (labeled I-N in **Figure 2B**). Note that in the work in this paper only the interior three cassette patterns (I, J, K) in each zone are optimized. The corner cassettes (N) are filled with stainless steel in this optimization and the cassettes labelled L and M are either the target coolants material (Zone A, B) or the thermal moderator (Zone C). The center cassette pattern, labeled I, is a variable sized that can be between 0-30 plates (up to an interior length 15"). The length of this cassette is a function of the number of plates within the cassette, with the experiment volume moving along with the cassette's changing length. The DD neutron source is in Zone C in a fixed location and is modelled as an isotropic 2.5 MeV neutron source. Other features of the FNS in the MCNP model are the stainless-steel reflector (F), the concrete pedestal (H), and the B₄C plates (G) which ensure subcriticality when inserted.

2 MATERIALS AND METHODS

2.1 Non-Dominated Sorting Algorithm-II

Genetic algorithms are a class of optimization algorithms which implement natural selection to optimize what may otherwise be intractable optimization problems. In simplest terms, a genetic algorithm takes an initial generation of individuals which are evaluated with respect to one or more objectives, and a subset of these individuals is selected and then combined to produce a unique individual and/or mutated randomly. How exactly each of the steps is accomplished is part of the art of a well-designed genetic algorithm. Genetic algorithms can also be augmented by using user-defined heuristics in each step to further increase the effectiveness of the algorithm to produce a suite of individuals required by the analyst.

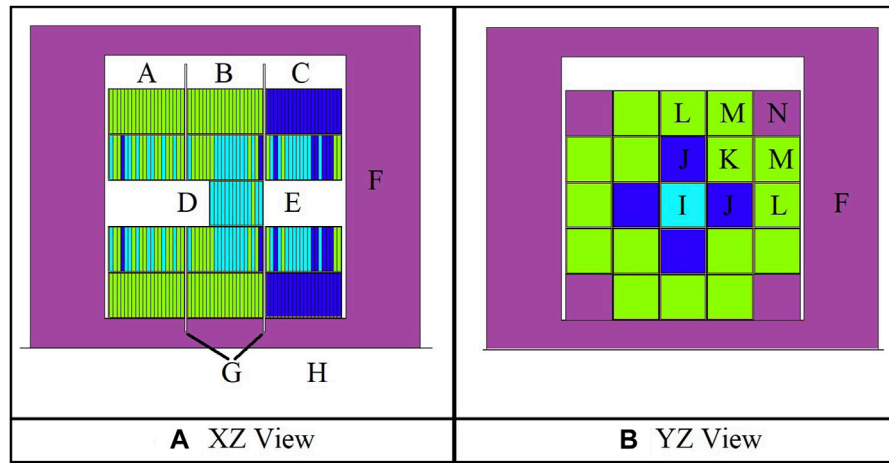


FIGURE 2 | Fast Neutron Source MCNP Geometry (A) XZ and (B) YZ View.

The Non-Dominated Sorting Algorithm-II (Deb et al., 2002) is a well-known variation on the standard genetic algorithm which incorporates the heuristics of elitism, non-dominated sorting and crowding distance to select the parents of the next generation. Elitism is simply the idea that parents are compared to the children of the current generation. This ensures that no progress made during the optimization is lost from one generation to the next. Non-dominated sorting is a heuristic for selecting which individuals are selected as parents of the next generation, which ranks individuals by the number of other individuals which dominate it (i.e.: have a better evaluation of an objective function). If no other individual has at least one objective which is better than a given individual, then that individual is non-dominated and is assigned rank 1. Subsequent individuals are assigned a non-dominated rank based on how many and which individuals dominate it. Rank 2 individuals are only dominated by rank 1 individuals, etc. Parents of the next generation are selected based on their non-dominated rank. If there are more individuals in each rank than there are available slots for parents, then the crowding distance metric is applied. Crowding distance is a heuristic which calculates the volume around each individual in the objective space. Individuals are selected as parents first based on the minimum and maximum for each objective function, and then by which occupy the largest volume in the objective space until the parent population is full. This heuristic is meant to both preserve the maximum and minimum individuals for each objective function and to preferentially select individuals which are in a less populated section of the objective space as parents with the goal of increasing the genetic diversity of the population.

The NSGA-II Algorithm is presented in **Figure 3**. It is adapted from the original paper describing the algorithm, A Fast and Elitist Multiobjective Genetic Algorithm NSGA-II (Deb et al., 2002):

2.2 Convolutional Neural Networks

A convolutional neural network (CNN) is a type of artificial neural network which can approximate a function in which not just the input values are important, but the relative positional information of the inputs is also important. CNNs are used primarily in machine vision tasks (Krizhevsky, Sutskever and Hinton 2012) and language processing tasks (Kalchbrenner, Grefenstette and Blunsom 2014). In machine vision tasks, the magnitude, relative position, and combinations of pixels are important for predicting what the pixels represent. In machine language tasks the relative positions of words are an important aspect to producing accurate translations.

CNN architectures include several layer types such as convolutional layers, pooling layers, non-linearity layers and fully connected layers (Albawi et al., 2017). In the convolutional layer, the namesake of the neural network architecture, is the convolution operation performed on the input to the layer and one or more learned kernels to produce a feature map. The convolution operation is a mathematical function that describes how one function modifies another as it is shifted over it. In practical terms, with a 2D input and kernel, the dot product between both functions is calculated and stored in the feature map. The kernel is then shifted by some number of columns and the dot product is calculated again. A non-linear function, such as a Rectified Linear Unit, is applied to the outputs of the convolutional layers.

The next layer type is the pooling layers. In these layers, a fixed filter is applied to the input to the layer. Commonly used filters are averaging and maximum pooling filters which return the average of a subset of the feature map or the maximum value within a subset. Unlike the convolutional layer, the stride of this operation is generally equal to the width of the filter. Commonly in machine-vision tasks a 2×2 filter is used. A 2×2 max pooling layer would reduce the size of the feature map by producing a new feature map which is composed of the maximum values in each 2×2 grid in the input feature map.

$R_t = P_t \cup Q_t$	Combine parent and offspring population
$F = \text{fast} - \text{non} - \text{dominated} - \text{sort}(R_t)$	$F = (F_1, F_2, I)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $ P_{t+1} + F_i \leq N$:	Until the parent population is filled
Crowding-distance-assignment(F_i)	Calculate crowding-distance in F_i
$P_{i+1} = P_{i+1} \cup F_i$	Include i th nondominated front in the parent population
$i = i + 1$	Check the next front for inclusion
Sort($F_i \prec_n$)	Sort in descending order using \prec_n
$P_{t+1} = P_{t+1} \cup F_i[1: (N - P_{t+1})]$	Choose the first $(N - P_{t+1})$ elements of F_i
$Q_{t+1} = \text{make} - \text{new} - \text{pop}(P_{t+1})$	Use selection, crossover, and mutation to create a new population Q_{t+1}
$t = t + 1$	Increment the generation counter

FIGURE 3 | The Non-Dominated Sorting Genetic Algorithm-II (Deb et al., 2002).

The last layer type commonly used in convolutional neural networks are fully connected layers. These final layers take as input the flattened final feature maps from the previous layers and non-linearly combine them into a prediction. The weights and biases in these layers, along with the kernels of the convolutional layers, are trained by backpropagation algorithm used to train other neural networks.

In nuclear applications, the relative position of materials to each other is important information when predicting nuclear quantities of interest such as k-eff. The k-eff of a given configuration of fissile, moderating, and absorbing materials is a function of where these materials are in 3-D space relative to each other, along with their respective nuclear data. CNNs produce predictions based on combinations of features which incorporate this 3-D data. There are other deep neural network architectures, such as recurrent (Liang and Hu, 2015) and transformer networks (Han et al., 2022), that have similarly been applied to machine vision tasks and therefore may also be able to predict nuclear related figures of merit such as k-eff, representativity, etc.

3 RESULTS AND DISCUSSION

This section discusses the initial optimization of the FNS using the NSGA-II algorithm and the subsequent optimization using the CNN-based surrogate models for the objective and constraint functions.

3.1 Optimization of the FNS by NSGA-II

The target of this FNS optimization is a generic sodium cooled fast reactor spectra. The objectives of this optimization are the

maximization of the neutron flux per source particle in the experiment volume, the maximization of the representativity of the flux spectra in the experiment volume and the maximization of the change in k-eff when placing the target material in the experiment volume. These objectives are used as heuristics in place of the true objective of the FNS, which is to produce configurations which minimize the uncertainty on a target reactor concept propagated from nuclear data. Maximizing the total flux per source particle would mean reducing the total time required to complete a FNS experiment to sufficient statistical certainty. Representativity, or the E similarity coefficient in the SCALE manual (Rearden and Jessee 2018), is the angle between two n-length vectors in n-dimensional space. A value of 0 means that the two vectors are perpendicular to each other, a value of 1.0 would correspond to the two vectors pointing in the same direction and are therefore proportional to each other. The integral k-eff objective seeks to maximize the delta between the FNS experiment and an integral experiment where the entire experiment volume is filled with the material of interest. This last objective approximates a potential use-case of the FNS to perform integral experiment where the reactivity worth of the target moderator in the system is being measured before and after insertion. Maximizing the Δk -eff of that experiment would make the practical matter of measuring the reactivity difference easier.

These objectives were calculated by an MCNP source calculation with a total uncertainty on the experimental volume flux tally converged to <0.005% standard error. In addition, a constraint on k-eff was enforced which required all parents to have a k-eff below 0.95. This constraint was calculated using MCNP and to a standard uncertainty of at least 0.00150 dk-eff. An increasingly strict constraint on

representativity based on the idea of simulated annealing was also enforced. This constraint increased linearly over the first 50 generations of the optimization to $E > 0.95$. This constraint was enforced such that if there were not enough individuals in the parent population which met it, then it would be relaxed until at least 20 individuals met the constraint. Enforcing the constraint on representativity later in the optimization allows the algorithm to explore areas on the design space that would not be allowed by a strict constraint.

This optimization used three plate types: 9.75% enriched uranium metal, polyethylene, and sodium metal. The initial optimization of the FNS was run on the NECLUSTER at the University of Tennessee. This NSGA-II algorithm was implemented with the parameters described in **Table 1**. A total of 8,100 potential patterns of the FNS were evaluated in this optimization. Of these, 3,145 individual patterns met the k-eff criteria and were evaluated for the neutron flux-based objectives. The stopping criteria used for this optimization was number of generations, which was selected due to taking approximately 3 days (wall time) to complete. The MCNP calculations of the objective functions required the most computational time.

Figure 4 shows the average k-eff, representativity, total flux, and the integral k-eff value (times 10). The integral value is multiplied by 10 to show more detail in the data. Some features of this figure are that the linearly increasing constraint on representativity is obvious from both the linearly increasing average representativity and the decreasing average total flux over generations 34 to 53. Thereafter, the average total flux of the parents increases slightly but plateaus at generation 61.

This optimization produced a Pareto front of FNS designs which can be seen in **Table 2**. This set of individuals represent the trade-off between the objective functions of representativity, total flux and integral k-eff as found by the optimization algorithm. The representativity of these individuals ranges from 0.9510 to 0.9789. The total flux in the experiment volume ranges from 0.00146 to 0.0041, and the Δk of the integral experiment ranges from -0.00592 to +0.00195. Like previous analysis of a simplified approximation of the FNS, the representativity of the flux in the experiment volume and the total of that flux are negatively correlated (-0.858) while the flux and Δk -eff are positively correlated (0.810). An increase in Δk -eff will increase the total flux in the experimental volume but at the expense of the ability of

the flux spectra to match the primarily fast target spectra as measured by representativity.

3.2 Surrogate Model Optimization of the FNS

The surrogate model optimization of the FNS used the same objectives and constraint as the optimization described in **Section 3.1**, but with the CNN-based surrogate models as solvers for the MCNP k-eff and source calculations required to evaluate the objectives. The architectures of these networks were found by the application of the Keras Tuner Python library (O'Malley, et al., 2019) using the data produced by the NSGA-II optimization. The hyper parameters found by this optimization can be found in **Table 3**. Further discussion of the method of optimizing the surrogate models will be presented in a forthcoming PhD dissertation at the University of Tennessee by the lead author.

The surrogate-based optimization algorithm is as follows:

1. Initialize population of 100 individuals and evaluate for objectives and constraint with MCNP.
2. Train surrogate models if more than 100 individuals have been evaluated.
3. Run NSGA-II algorithm (Deb et al., 2002) with the parameters described in **Table 1**, using the CNN-based surrogate models as the objective and constraint solvers.
4. Evaluate the final Pareto front (80 individuals) from the surrogate-based optimization with MCNP
5. If the total number of generations equals the stopping criteria, exit.
6. Return to Step #2

The data produced by the outer loop of the optimization, where MCNP is used to calculate the objectives and the constraint functions for every individual, was used for the training of the surrogate models used to evaluate the objectives of the inner loop of the optimization. A minimum of 100 valid individuals are required for training to occur, so after the first generation only the k-eff surrogate model was trained. The objective functions were trained after generation 2. The surrogate models were trained for 500 epochs using 90% of the available training data with 10% used for validation. The weights and biases of the epoch which

TABLE 1 | Description of NSGA-II and surrogate NSGA-II hyperparameters.

Description	NSGA-II hyperparameters	Surrogate NSGA-II hyperparameters
Stopping Criteria	100 Generations after initial generation	10 Generations after initial generation
Parent Population	20	80
Child Population	80	1,920
Crossover Rate	50%	50%
Crossover Type	Single point crossover	Single point crossover
Mutation Rate	10%, per plate	10%, per plate
Mutation Type	Single plate material change	Single Plate Material Change
Initial Population	100 randomly created	80 Pareto front individuals selected from all individuals evaluated with MCNP plus 1,920 randomly created
Objective/Constraint Solver	MCNP	CNN surrogate

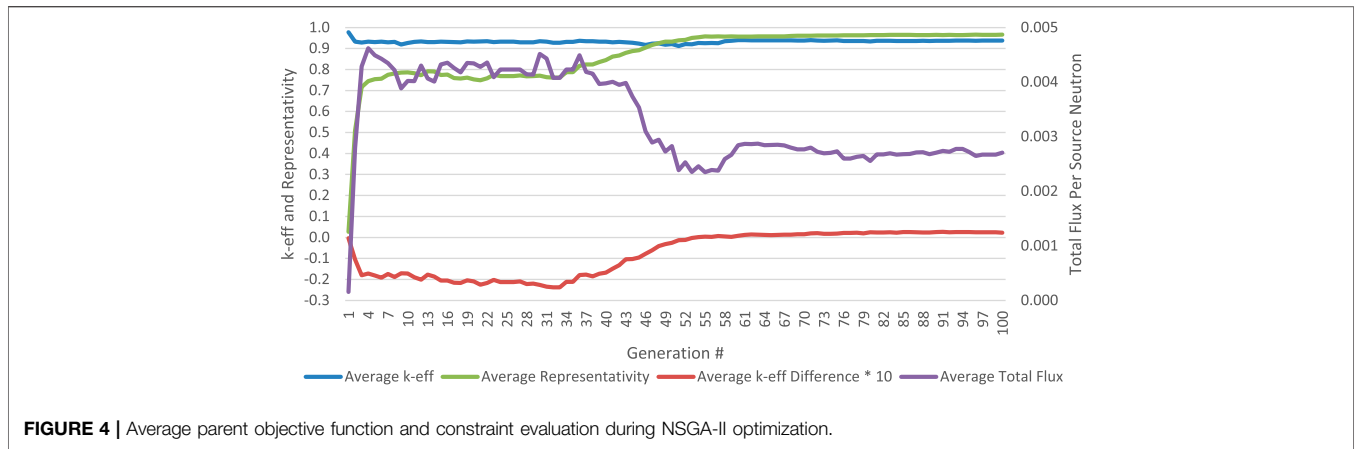


TABLE 2 | Final 20 individuals NSGA-II optimization.

Individual #	k-eff	Representativity	Total flux per source particle	Delta k-eff
1	0.94695	0.97893	0.002218	0.00002
2	0.93427	0.97825	0.002068	0.00144
3	0.91207	0.97750	0.001461	0.00191
4	0.91730	0.97548	0.001545	0.00405
5	0.94904	0.97460	0.002740	0.00178
6	0.91365	0.97293	0.001757	0.00468
7	0.93771	0.97225	0.002202	-0.00195
8	0.93805	0.96988	0.002228	0.00393
9	0.93879	0.96869	0.002518	0.00296
10	0.94841	0.96764	0.002966	0.00374
11	0.94323	0.96562	0.002991	0.00086
12	0.92581	0.96504	0.002105	0.00592
13	0.94596	0.96129	0.003270	-0.00027
14	0.92903	0.96045	0.002396	0.00471
15	0.94987	0.95876	0.003172	0.00266
16	0.94908	0.95574	0.003736	0.00122
17	0.94713	0.95333	0.003174	0.00268
18	0.94721	0.95227	0.003905	0.00150
19	0.94859	0.95187	0.004131	-0.00151
20	0.94371	0.95095	0.003571	0.00450

minimized the validation error were used as the surrogate model for that generation. The models were initialized with random weights at the beginning of the optimization.

The ability of the surrogate models to predict their respective values can be quantified by calculating the mean squared error (MSE) of the predictions to the true values as defined by,

$$Mean\ Squared\ Error = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2.$$

Where, x_i and \hat{x}_i are the true and predicted values.

The MSE of each of the surrogate models initially are relatively large and decrease as the number of training examples increases. The surrogate model predicting k-eff is the first to stabilize at a value of approximately 1.0E-01 at generation 10. During the optimization the training data for this model increases by 80 in each step. If an individual does not have a k-eff below 0.95, then the three objective functions are not calculated for it. **Figure 5** shows the MSE of the prediction of the objective and constraints versus the true, MCNP-calculated, values for all individuals produced in each generation. The MSE of the surrogate models of the objective functions plateau at approximately

TABLE 3 | FNS CNN surrogate model hyperparameters.

Hyper parameter	k-eff	Total flux	Representativity	Integral K-Eff
1st Conv. Width	32	40	56	16
2nd Conv. Width	64	64	16	32
3rd Conv. Width	64	40	48	40
4th Conv. Width	40	24	16	24
5th Conv. Width	N/A	N/A	N/A	40
6th Conv. Width	N/A	N/A	N/A	16
Kernel Size	4	4	3	3
Dense Layer Width	32	256	32	96
Max Pool Size	9	4	10	7
# of Hidden Conv. Layers	3	3	3	5
Dropout Percentage	0.5	0.2	0.5	0.4
Learning Rate	0.0208839	0.0160466	0.0317394	0.0018139

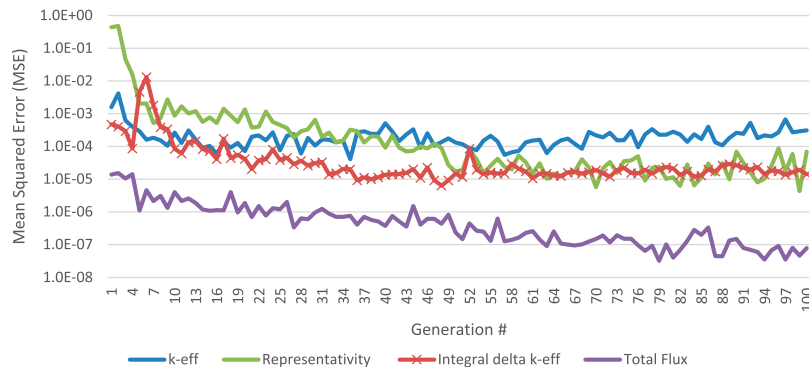


FIGURE 5 | MSE of surrogate models during optimization.

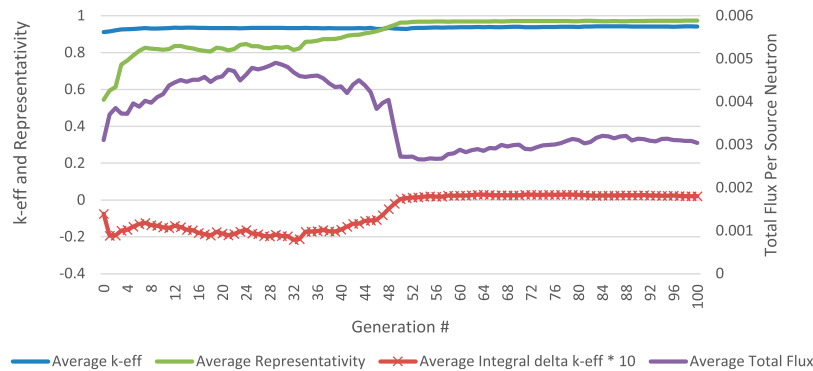


FIGURE 6 | Average objective and constraint evaluation during surrogate NSGA-II optimization for all parents in each generation.

generation 36 for the experimental k-eff value surrogate, generation 65 for the representativity surrogate and generation 80 for the total flux surrogate.

During this optimization, a total of 8,100 individuals were evaluated with MCNP and 2,120,000 individuals were evaluated with the surrogate models. Of the 8,100 patterns evaluated with MCNP a total of 5,868 individuals met the k-eff constraint and therefore evaluated with a MCNP source calculation. The average objective function and k-eff of all the individuals in the parent population is plotted in **Figure 6**. Like the NSGA-II algorithm before, the constraint on representativity that is maximally applied at generation 50 has a visible effect on the parent population.

The objective and constraint evaluations of the Pareto front individuals are provided in **Table 4**. The representativity of these individuals ranged between 0.95075 and 0.99502, the total flux per source neutron between 0.00077 and 0.00512 and the maximum and minimum integral Δk -eff values were +0.00717 and -0.00490, respectively. Like in the NSGA-II calculation, representativity and total flux per source neutron had a strong negative correlation (-0.95096).

3.3 Comparison of the Optimizations

The surrogate-based NSGA-II optimization of the FNS produced a Pareto front of potential FNS designs which outperformed those produced by the standard NSGA-II algorithm. **Table 5** presents the average, maximum and minimum of the objectives and constraint functions of the final Pareto front of each calculation. The surrogate-based optimization produced individuals with higher representativity, total flux per source particle and both a larger positive and negative integral Δk -eff. The final Pareto front of both calculations plotted by representativity vs. total flux per source particle and integral experimental k-eff vs. representativity is presented in **Figure 7** and **Figure 8**. These figures show that the surrogate-model based optimization produced a suite of individuals which better optimize the objectives.

A total of 8,100 potential FNS patterns were evaluated in during the both the NSGA-II and NSGA-II surrogate optimizations. In the standard NSGA-II calculation a total of 3,145 potential patterns met the k-eff constraint of 0.95. During the NSGA-II with surrogate model, a total of 5,845 potential patterns met the k-eff constraint. This is an increase of over 85% more viable FNS configurations evaluated in the surrogate accelerated calculation.

TABLE 4 | Final 100 individuals CNN-surrogate NSGA-II optimization.

Ind. #	k-eff	Rep	Total flux	Int Δk-eff	Ind. #	k-eff	Rep	Total flux	Int Δk-eff
1	0.86656	0.99502	0.00085	-0.00058	41	0.94480	0.97322	0.00303	0.00381
2	0.84565	0.99483	0.00077	-0.00128	42	0.94430	0.97188	0.00280	0.00645
3	0.90486	0.99476	0.00110	0.00037	43	0.94883	0.97118	0.00352	0.00157
4	0.90453	0.99462	0.00106	0.00148	44	0.94627	0.97090	0.00307	0.00400
5	0.88307	0.99409	0.00092	0.00303	45	0.94724	0.97068	0.00339	0.00256
6	0.94573	0.99409	0.00176	0.00003	46	0.94332	0.97064	0.00317	0.00430
7	0.94834	0.99308	0.00192	0.00116	47	0.94785	0.96974	0.00344	0.00199
8	0.94824	0.99297	0.00185	-0.00179	48	0.94906	0.96917	0.00357	0.00157
9	0.94705	0.99280	0.00187	-0.00385	49	0.94334	0.96841	0.00322	0.00327
10	0.94807	0.99187	0.00201	0.00203	50	0.94213	0.96836	0.00324	0.00393
11	0.94601	0.99114	0.00197	0.00304	51	0.94366	0.96778	0.00309	0.00459
12	0.94664	0.99079	0.00208	0.00229	52	0.94655	0.96706	0.00338	0.00465
13	0.94800	0.98978	0.00200	-0.00348	53	0.93790	0.96665	0.00303	0.00529
14	0.94374	0.98886	0.00172	-0.00448	54	0.94861	0.96638	0.00346	0.00184
15	0.94591	0.98876	0.00230	0.00208	55	0.94789	0.96625	0.00344	0.00261
16	0.88478	0.98852	0.00110	0.00502	56	0.94633	0.96557	0.00350	0.00167
17	0.94551	0.98746	0.00239	0.00278	57	0.94892	0.96512	0.00356	0.00307
18	0.93909	0.98695	0.00181	0.00415	58	0.94965	0.96450	0.00413	0.00157
19	0.94767	0.98596	0.00225	-0.00348	59	0.94463	0.96399	0.00357	0.00343
20	0.94870	0.98494	0.00251	0.00296	60	0.94700	0.96272	0.00419	0.00030
21	0.94806	0.98472	0.00252	-0.00003	61	0.94698	0.96081	0.00430	-0.00065
22	0.93854	0.98468	0.00188	-0.00374	62	0.94770	0.96068	0.00435	0.00118
23	0.94867	0.98407	0.00253	0.00192	63	0.94942	0.96004	0.00430	0.00121
24	0.94735	0.98385	0.00240	0.00301	64	0.94336	0.95989	0.00432	0.00135
25	0.92275	0.98374	0.00172	0.00576	65	0.94886	0.95958	0.00456	0.00107
26	0.94248	0.98346	0.00207	-0.00490	66	0.94951	0.95917	0.00458	0.00181
27	0.94835	0.98342	0.00237	0.00348	67	0.94756	0.95866	0.00452	0.00330
28	0.94765	0.98300	0.00266	0.00090	68	0.94824	0.95809	0.00479	0.00130
29	0.94594	0.98173	0.00233	0.00524	69	0.91854	0.95791	0.00279	0.00717
30	0.94690	0.98116	0.00266	0.00408	70	0.94885	0.95771	0.00459	0.00290
31	0.94897	0.97944	0.00282	0.00201	71	0.94605	0.95724	0.00434	0.00338
32	0.94878	0.97897	0.00267	0.00249	72	0.94886	0.95590	0.00433	0.00344
33	0.94429	0.97866	0.00280	0.00213	73	0.94709	0.95555	0.00473	0.00371
34	0.94766	0.97851	0.00283	0.00327	74	0.94565	0.95527	0.00476	0.00243
35	0.94557	0.97816	0.00282	0.00388	75	0.94688	0.95517	0.00448	0.00508
36	0.94740	0.97655	0.00288	-0.00150	76	0.94799	0.95497	0.00486	0.00351
37	0.94984	0.97601	0.00315	-0.00178	77	0.94526	0.95436	0.00458	0.00443
38	0.94662	0.97463	0.00286	0.00258	78	0.94195	0.95367	0.00453	0.00575
39	0.94811	0.97421	0.00304	0.00197	79	0.94922	0.95197	0.00512	0.00101
40	0.94567	0.97381	0.00266	0.00411	80	0.94733	0.95075	0.00472	0.00513

TABLE 5 | Average, Minimum and Maximum Constraint and Objective Functions of Final Pareto Front of NSGA-II and NSGA-II Surrogate Calculations. Mean values are italicized with the minimum and maximum values given in brackets below.

	NSGA-II	Surrogate NSGA-II	% Difference
k-eff	<i>0.93829, 0.91207, 0.94987</i>	<i>0.94102, 0.84565, 0.94984</i>	0.003% [-7.3%, 0.3%]
Representativity	<i>0.96557, 0.95095, 0.97893</i>	<i>0.99502, 0.95075, 0.97402</i>	1.6% [-0.02%, 0.9%]
Total Flux Per Source Particle	<i>0.00271, 0.00146, 0.00413</i>	<i>0.00304, 0.00077, 0.00512</i>	24% [-47%, 12%]
Integral k-eff	<i>0.00592, -0.00195, 0.00224</i>	<i>0.00209, -0.00490, 0.00717</i>	-6.7% [151%, 21%]

The trade-off of this increase in both calculational efficiency and in the more optimized Pareto Front is computational time for both the increased number of MCNP calculations and for the training and utilizing of the surrogate models. The increased number of MCNP calculations is a by-product of more effectively producing potential FNS designs and could be resolved by decreasing the number of the parent population. In the first time step the training of the surrogate models

required an average of 2.3 min. By the final time step this increased to a maximum of 33 min. This increase in time requirement could be offset by capping the total number of examples used for surrogate model training or by reducing the size and complexity of the surrogate models themselves. Once trained, the surrogate models required on average a total of approximately 7.5 s to evaluate the 1,920 unique children in each interior GA step.

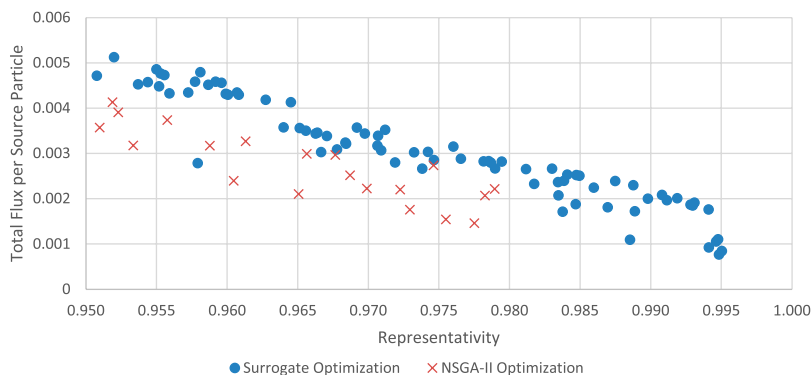


FIGURE 7 | Comparison of Pareto Front Individuals from Both Optimizations, Total Flux vs. Representativity.

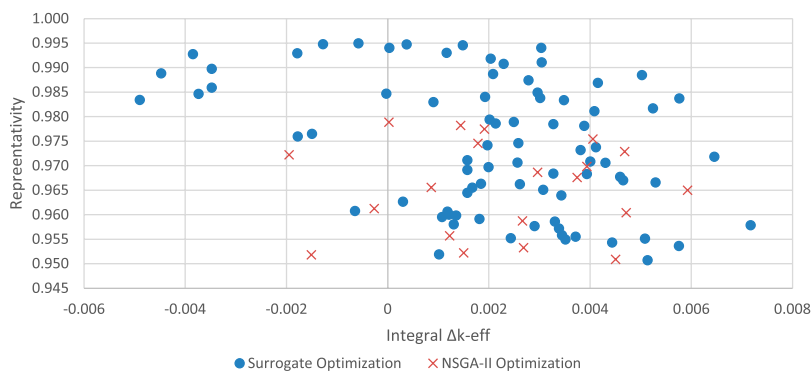


FIGURE 8 | Comparison of Pareto Front Individuals from Both Optimizations, Representativity vs. Integral Experiment Δk -eff.

4 CONCLUSION

The work in this paper presents the benefit of accelerating a genetic algorithm used for a multi-objective optimization of a nuclear experiment using convolutional neural network surrogate models comparing to a standard benchmark genetic algorithm. These surrogate models are trained in-line during the genetic algorithm and allow the evaluation of an increased number of potential designs, which leads to an increase in all objective functions. The architectures for the k-eff, representativity, neutron flux and integral k-eff experiment surrogate models are presented along with the methodology for producing them. Future work includes further expanding the use of the surrogate models for other useful objectives relevant for selecting FNS designs and producing more surrogates for other FNS designs targeting uncertainties in next-generation reactor designs.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

JP: Student research lead and lead author. VS: Committee member and primary research advisor. WH: Graduate committee chair and research lead.

FUNDING

This work is funded by the University of Tennessee, Knoxville Department of Nuclear Engineering.

ACKNOWLEDGMENTS

This is a short text to acknowledge the contributions of specific colleagues, institutions, or agencies that aided the efforts of the authors.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fenrg.2022.874194/full#supplementary-material>

REFERENCES

- Albawi, S., Tareq, A. M., and Al-Zawi, S. (2017). "Understanding of a Convolutional Neural Network," in 2017 International Conference on Engineering and Technology (ICET), 1–6. doi:10.1109/icengtechnol.2017.8308186
- Bostelmann, R., Ilas, G., Celik, C., Holcomb, A. M., and Wieselquist, W. (2021). *Nuclear Data Assessment for Advanced Reactors*. Oak Ridge, TN (United States): Tech. Rep., Oak Ridge National Lab. doi:10.2172/1840202
- Chapot, J. L. C., Carvalho Da Silva, F., and Schirru, R. (1999). A New Approach to the Use of Genetic Algorithms to Solve the Pressurized Water Reactor's Fuel Management Optimization Problem. *Ann. Nucl. Energy* 26, 641–655. doi:10.1016/s0306-4549(98)00078-4
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 182–197. doi:10.1109/4235.996017
- El-Sefy, M., Yosri, A., El-Dakhkhni, W., Nagasaki, S., and Wiebe, L. (2021). Artificial Neural Network for Predicting Nuclear Power Plant Dynamic Behaviors. *Nucl. Eng. Technol.* 53, 3275–3285. doi:10.1016/j.net.2021.05.003
- Faria, E. F., and Pereira, C. (2003). Nuclear Fuel Loading Pattern Optimisation Using a Neural Network. *Ann. Nucl. Energy* 30, 603–613. doi:10.1016/s0306-4549(02)00092-0
- Goorley, T., James, M., Booth, T., Brown, F., Cox, L. J., Durkee, J., et al. (2012). Initial MCNP6 Release Overview. *Nucl. Technol.* 180, 298–315. doi:10.13182/nt11-135
- Gougar, H. D., Ougouag, A. M., Terry, W. K., and Ivanov, K. N. (2010). Automated Design and Optimization of Pebble-Bed Reactor Cores. *Nucl. Sci. Eng.* 165, 245–269. doi:10.13182/nse08-89
- Han, K., Yunhe, W., Hanting, C., Xinghao, C., Jianyuan, G., Zhenhua, L., et al. (2022). "A Survey on Vision Transformer," in IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE).
- Hanna, B. N., Dinh, N. T., Youngblood, R. W., and Bolotnov, I. A. (2020). Machine-Learning Based Error Prediction Approach for Coarse-Grid Computational Fluid Dynamics (CG-CFD). *Prog. Nucl. Energy* 118, 103140. doi:10.1016/j.pnucene.2019.103140
- Hogle, S. (2012). *Optimization of Transcurium Isotope Production in the High Flux Isotope Reactor*. PhD Thesis. University of Tennessee.
- Kalchbrenner, Nal., Grefenstette, Edward., and Blunsom, Phil. (2014). "A Convolutional Neural Network for Modelling Sentences," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Baltimore, Maryland: Association for Computational Linguistics), 655–665. doi:10.3115/v1/p14-1062
- Kim, B. S., and Moon, J. H. (2010). Use of a Genetic Algorithm in the Search for a Near-Optimal Shielding Design. *Ann. Nucl. Energy* 37, 120–129. doi:10.1016/j.anucene.2009.11.014
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* 25. doi:10.1145/3065386
- Liang, M., and Xiaolin, H. (2015). "Recurrent Convolutional Neural Network for Object Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3367–3375.
- O'Malley, T., Elie, B., Long, J., Francios, C., Haifeng, J., and Luca, I. (2019). *Keras Tuner*. <https://github.com/keras-team/keras-tuner>.
- Pevey, J., Chvála, O., Davis, S., Sobes, V., and Hines, J. W. (2020). Genetic Algorithm Design of a Coupled Fast and Thermal Subcritical Assembly. *Nucl. Technol.* 206, 609–619. doi:10.1080/00295450.2019.1664198
- Prabhu, S. R., Pandey, M. D., Christodoulou, N., and Leitch, B. W. (2020). A Surrogate Model for the 3D Prediction of In-Service Deformation in CANDU Fuel Channels. *Nucl. Eng. Des.* 369, 110871. doi:10.1016/j.nucengdes.2020.110871
- Rearden, B. T., and Jessee, M. A. (2018). *SCALE Code System*. Oak Ridge, TN (United States): Tech. Rep., Oak Ridge National Lab. doi:10.2172/1426571
- Sobes, V., Hiscox, B., Popov, E., Archibald, R., Hauck, C., Betzler, B., et al. (2021). AI-Based Design of a Nuclear Reactor Core. *Sci. Rep.* 11, 19646–19649. doi:10.1038/s41598-021-98037-1
- Sobester, A., Forrester, A., and Keane, A. (2008). *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons.
- Tunes, M. A., De Oliveira, C. R. E., and Schön, C. G. (2017). Multi-Objective Optimization of a Compact Pressurized Water Nuclear Reactor Computational Model for Biological Shielding Design Using Innovative Materials. *Nucl. Eng. Des.* 313, 20–28. doi:10.1016/j.nucengdes.2016.11.009
- Zhao, J., Knight, B., Nissan, E., and Soper, A. (1998). FUELGEN: A Genetic Algorithm-Based System for Fuel Loading Pattern Design in Nuclear Power Reactors. *Expert Syst. Appl.* 14, 461–470. doi:10.1016/s0957-4174(98)00005-0

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Pevey, Sobes and Hines. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.