

## Minimizing CRUD deposition through optimization of associated parameters

Brian Andersen<sup>1</sup>, Jason Hou<sup>1</sup>, and Dave Kropaczek<sup>2</sup>

<sup>1</sup>North Carolina State University  
2500 Stinson Drive.  
Raleigh NC,

<sup>2</sup>Oak Ridge National Laboratory  
*One Bethel Valley Road, Bldg. 5700, MS-6003,  
Oak Ridge. TN, 37830,*

Bdander3@ncsu.edu, jason.hou@ncsu.edu, kropaczekdj@ornl.gov

### ABSTRACT

A strong correlation exists between subcooled boiling in assembly subchannels and CRUD deposition. In this work a genetic algorithm is used to optimize a 17 x 17 PWR fuel assembly to have minimized subcooled boiling, minimized peak  $kinf$ , and maximized end of cycle  $kinf$ . Optimization of these parameters act as a surrogate for the optimization of CRUD deposition in fuel assemblies due to their strong correlation. Subcooled boiling, measured by vapor void in a sub channel, and values of  $kinf$ , are calculated using VERA-CS. Due to the high computational cost of VERA-CS, artificial neural networks are used as surrogate models to VERA-CS in order for the optimization to be performed in a timely manner making design work possible. Two neural networks were trained using a training library of 1200 randomly generated assembly designs and a validation library of 100 assembly designs evaluated using VERA-CS. The combination of neural networks and genetic algorithms formed an extremely fast optimization algorithm capable of evaluating designed a set of optimized pin lattices in a matter of minutes. The optimization showed a clear reduction in vapor void in the optimization solutions. This provides a proof of principle that complex phenomena requiring coupled, Multiphysics calculations, such as CRUD deposition, may be optimized.

KEYWORDS: Genetic Algorithm, Machine Learning, CRUD, Optimization

### 1. INTRODUCTION

The deposition of CRUD, with the backronym Chalk River Unidentified deposits, is a serious concern for the operating fleet of light water reactors, particularly as they are pushed towards higher burnups. CRUD deposition causes power shifts within operating reactor cores, known as CRUD induced power shift (CIPS) or the axial offset anomaly (AOA), and increased corrosion of fuel rods, known as CRUD induced local corrosion (CILC) [1]. Therefore, designing assemblies with CRUD deposition in mind is becoming increasingly important as fuel assemblies are pushed towards higher burnups.

High fidelity, Multiphysics simulation tools such as the Virtual Environment for Reactor Analysis Core Simulator (VERA-CS) allows for coupling of codes such as MPACT and CTF to provide three-dimensional pin level detail of parameters correlated to CRUD deposition such as sub channel vapor void.

[2]. In this work, a pressurized water reactor fuel assembly is optimized to have reduced sub-channel vapor void, minimized peak  $k_{inf}$ , and maximized end-of-cycle (EOC)  $k_{inf}$ . Sub-channel vapor void is chosen as an optimization parameter based on the strong correlation between subcooled boiling and CRUD deposition in PWR fuel assemblies. This makes this work a proof of concept for future assembly optimization of CRUD deposition.

Design of fuel assemblies core loading patterns via optimization algorithms is a practice that has been well established. For example, Rogers et al optimized the power peaking factor of a 15 x 15 pressurized water reactor pin lattice at beginning of cycle using adaptive simulated annealing [3]. Optimization algorithms are often extensively used in the optimization of boiling water reactor (BWR) fuel lattices and bundles. Martin-del-Campo et al. used genetic algorithms for optimizing lattice enrichments, gadolinium concentrations, and power peaking factors in BWR lattices [4].

Optimization algorithms often require many solutions, often in the order of thousands, to be evaluated. For this reason, optimization algorithms generally use fast-running, low fidelity analysis codes such as CASMO so that optimization time is kept relatively acceptable [5]. VERA-CS, while extremely powerful in regard to its simulation capabilities, is expensive in terms of wall clock time and computational resources. This makes it unsuited for use in an optimization algorithm. In order to overcome this computational burden, artificial neural networks (ANN's) are used as surrogate models for VERA-CS to maintain the accuracy of the high-fidelity computations from VERA-CS while having a fast-running optimization algorithm [6]. Genetic algorithms are used as the optimization methodology [7].

## 2. OPTIMIZATION DESCRIPTION

For this work, a 17 x 17 PWR fuel assembly in octant symmetry was optimized. The chosen optimization objectives are minimizing total vapor void in the assembly over the course of a single cycle, minimization of the peak infinite neutron multiplication factor during the cycle, and maximization of the neutron multiplication factor at end of cycle. Minimizing vapor void in the assembly is representative of reducing subcooled boiling in the assembly. Operating parameters for the assembly are provided in Table I.

**Table I. Assembly Parameters:** Assembly operating parameters used in the optimization.

Operating Power	20.68 MW
Inlet Temperature	565 K
Boron	900 PPM
Pressure	15.51 MPa
Coolant Flow Rate	$3.094 \cdot 10^5$ Kg/hr

### 2.1. Artificial Neural Network Description

ANN's learn by optimizing a set of weights and biases to predict the correct function output based on a given input. The weights and biases are modified through a process known as backpropagation. In backpropagation, the error of the network in predicting the output of a given set of training data is used to determine the change in the weights and biases given a set learning rate [6].

Separate neural networks were utilized for predicting vapor void and kinf values in the assembly designs. The neural networks were developed using the KERAS deep learning library [6]. The standard multilayer perceptron model was used for the ANNs. The number of layers and nodes for the neural networks and training hyperparameters are provided in Table II. These parameters were determined from extensive tuning of the neural network models. This is why hyperparameters such as the learning rate and dropout rate differ extensively between the two models.

**Table II. ANN Architecture:** Parameters used in training the two neural networks used in the optimization.

Parameter	Kinf Neural Network	Vapor Void Neural Network
Number of Input Nodes	762	762
Number Nodes First Hidden Layer	100	700
Number Nodes Second Hidden Layer	100	200
Number of Output Nodes	20	20
Dropout Rate	0.2	0.001
Momentum	0	0.7
Weight Decay	0	$10^{-6}$
Learning Rate	$10^{-3}$	$10^{-5}$

The pin lattice formed the input to the neural network. Pin lattice information was input to the neural network using “one-hot-encoding,” in which a series of ones and zeros are used to represent the specific pin type in a given location in the assembly. The one-hot-encoding is presented in Table II.

**Table III. Encoding:** The one-hot encoding used to represent each of the pin types used in the optimization.

Pin Number	Type	Enrichment (w/o)	Burnable Poison
1	Fuel	3.1	-
2	Guide Tube	-	-
3	Fuel	3.9	-
4	Fuel	4.5	-
5	Fuel	3.1	IFBA
6	Fuel	3.9	IFBA
7	Fuel	4.5	IFBA
8	Fuel	3.1	1% Gad
9	Fuel	3.1	2% Gad
10	Fuel	3.9	1% Gad
11	Fuel	3.9	2% Gad
12	Fuel	4.5	1% Gad
13	Fuel	4.5	2% Gad

A rectilinear activation functions was used for the two hidden layers:

$$F(x) = \max(0, x^+) \quad (1)$$

A linear activation function was used for the final output layer of the neural network:

$$F(x) = \sum x \quad (1)$$

The root-mean-squared (RMS) loss function was used to train the weights and biases of the neural network:

$$F(x, y) = \sqrt{\frac{1}{n} \sum (y - x)^2} \quad (1)$$

Finally, effectiveness of the ANN in matching the training and validation data was calculated using the mean absolute error (MAE):

$$F(x, y) = |y - x| \quad (1)$$

A training library was generated using 1,200 randomly generated pin lattices. A second library for validation of the neural networks was generated using an additional 100 randomly generated lattice cases. VERA-CS version 3.7.1 was used to generate the neural network training libraries on the Henry 2 cluster at North Carolina State University [8].

As previously stated, VERA-CS is a computationally expensive code in terms of wall clock time and processor requirements. 1300 cases were used for the training and validation of the neural network because that was the number of cases that could be run within the allotted time. This was also the decision for running single assembly experiments. The number of processors required to run a full core VERA-CS calculation is more than the allotted processor limit of 128 cores on the Henry 2 cluster.

## 2.2. Genetic Algorithm Description

Genetic algorithms are designed to maximize the fitness of a population of solutions to a given design problem. Solutions to the optimization problem are described through a solution genome. While genomes for the initial population of solutions is generated randomly, the genomes of new solutions, known as children, are created through mutation and crossover of the parent population. In crossover, solutions of similar genomes are “mated” together, and half of the differing genes between the two genomes are randomly swapped between the two solutions to create two new solutions to the optimization problem. In mutation, genes are randomly replaced in a solution genome to generate new solutions to the optimization problem. Values for the population size, number of generations, and mutation rates are provided in Table IV.

The parents of each generation are chosen through a roulette selection method. In roulette, every solution is assigned a probability of reproducing based on its fitness. Random numbers are generated, and if the random number matches the probability of reproducing, the solution is selected to be a parent for the next generation.

**Table IV. Optimization Parameters:** Parameters used in performing the genetic algorithm optimization.

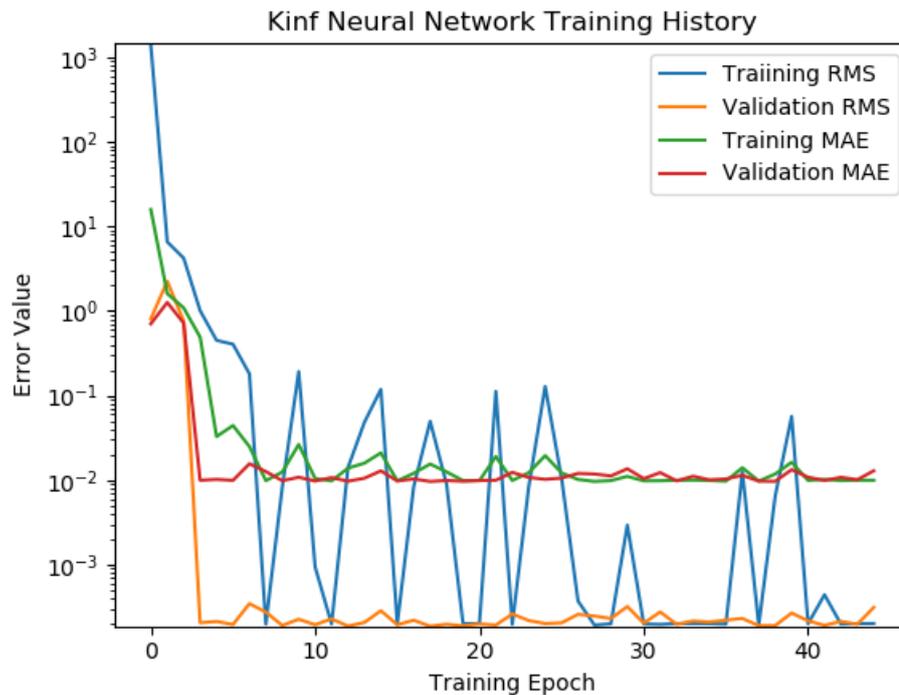
Population Size	160
-----------------	-----

Number of Generations	100
Initial Mutation Rate	25%
Final Mutation Rate	75%

The genetic algorithm implemented for this work is a modification of the Multi-Objective Optimization utilizing Genetic algorithms for Lattice Enhancement (MOOGLE) [8]. The key feature of MOOGLE is the use of a modified Pareto front composed of bins. The motivation for binning the solution space is the thought that in design problems where multiple solutions to the optimization problem are desired, the best combination of solutions may be created utilizing solutions that do not lie directly on the Pareto Front. The desired size of the binned solution space and size of the bins is specified as input to the optimization. In the event that no solutions in the current population of solutions lie on the Pareto Front, fitness is calculated based on the distance to the binned solution space. Once solutions exist inside the binned solution space, fitness is calculated based on the number of bins to the optimal edge of the desired solution space.

### 3. RESULTS

#### 3.1. ANN Training Results



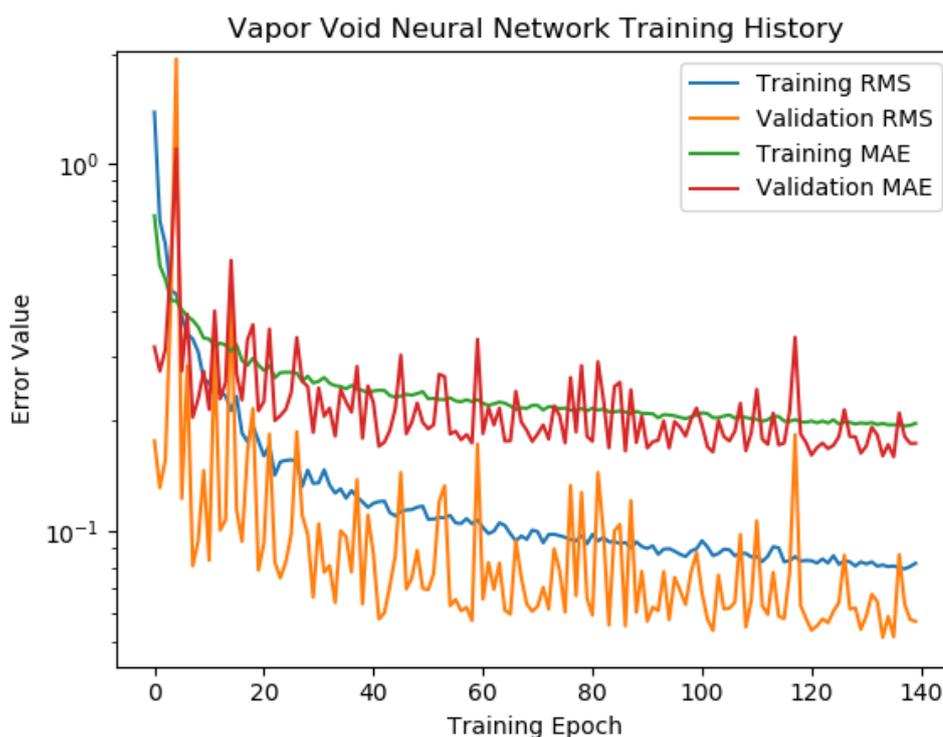
**Figure 1. Training history of the neural network for predicting assembly Kinf values.**

RMS loss sums and MAE for training and validation data over the course of training for the kinf ANN is presented in Figure 1. RMS loss sums and MAE for training and validation data over the course of training for the kinf ANN is provided in Figure 2. Final RMS loss values and MAE for the networks is provided in Table V. The accuracy and prediction capability of the neural networks could likely be

improved through additional training samples. However, the achieved accuracy is sufficient for demonstrating the capabilities of the optimization.

**Table V. Losses:** Final Training Loss Values for the ANNs.

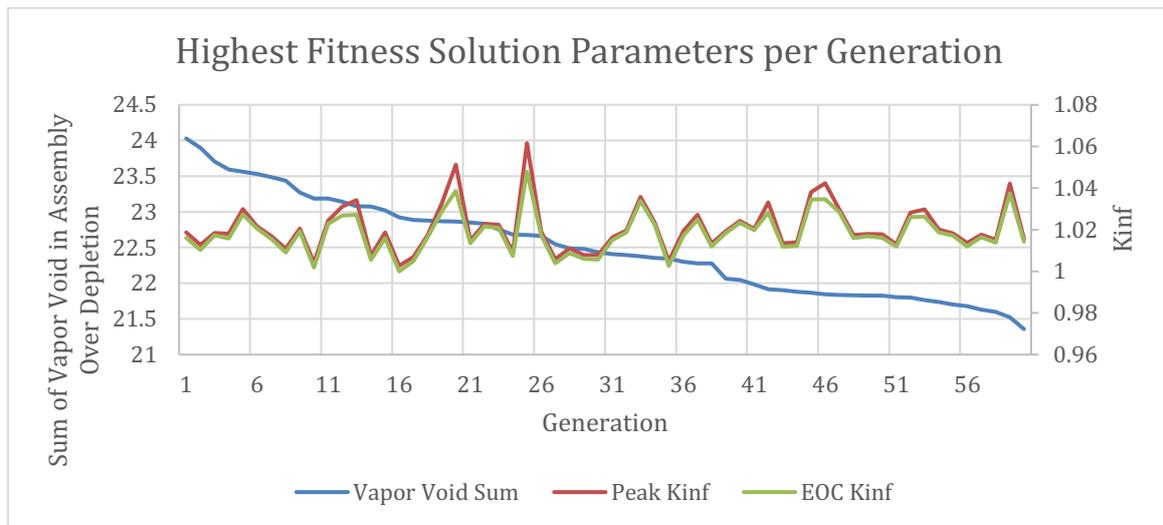
Error	Kinf Network	Vapor void network
Training RMS Loss	$2.018 \cdot 10^{-4}$	$8.196 \cdot 10^{-2}$
Validation RMS Loss	$3.130 \cdot 10^{-4}$	$5.693 \cdot 10^{-2}$
Training MAE	$9.946 \cdot 10^{-3}$	$1.970 \cdot 10^{-1}$
Validation MAE	$1.291 \cdot 10^{-3}$	$1.740 \cdot 10^{-1}$



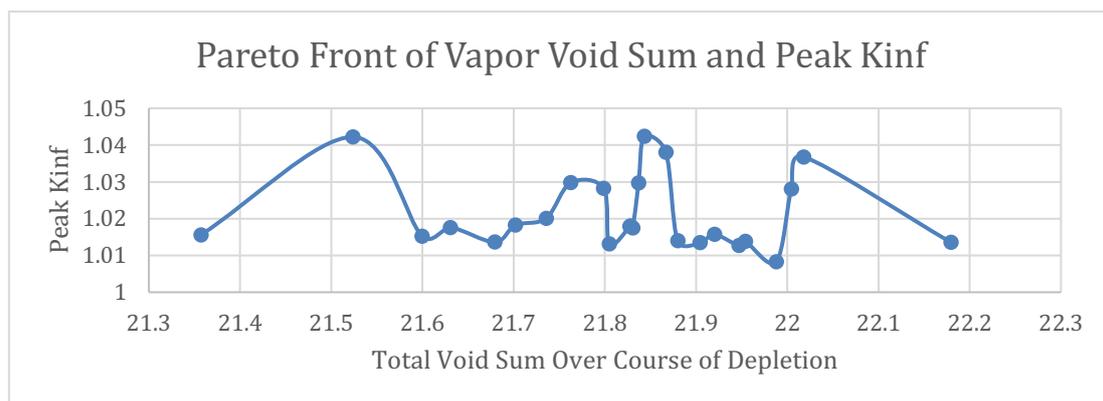
**Figure 2. Training history of the neural network for predicting vapor void sum in assemblies.**

### 3.2. Genetic Algorithm Optimization Results

The genetic algorithm optimization was carried out over 60 generations using a population size of 160 solutions for every generation. The objectives for the optimization were to minimize the total amount of vapor void produced in the assembly and maximum kinf within the assembly over a 20 GWD/MTU depletion, while maximizing the EOC kinf value of the assembly using the pin types listed in Table III as the decision variable. Values for the three optimization objectives for the highest fitness solution at each generation within the optimization are presented in Figure 3. A pareto front, the optimized solutions that are non-dominated by any other solutions, formed from connected solutions in the final optimization population for the parameters vapor void sum and maximum kinf is presented in Figure 4.



**Figure 3. Parameters for highest fitness solution for each generation in the optimization.**



**Figure 4. Pareto front vapor void sum versus peak kinf in final solution space.**

Figure 3 shows that vapor void is significantly reduced over the course of the optimization. In contrast, the values of maximum and EOC kinf vary within a 6% range between 1.00 and 1.06. This indicates that the optimization prioritizes vapor void over kinf. As somewhat expected, the values of peak and EOC kinf are closely aligned due to the correlated nature of the two parameters.

The pareto front indicates that there is no direct correlation between peak kinf values and the vapor void sum over the course of the optimization. This is drawn from the fact that as the vapor void sum increases over the front there is no clear trend in peak kinf values.

#### 4. CONCLUSIONS

High fidelity Multiphysics codes such as VERA-CS cannot be used for design work due to the high computational cost involved. The results of this paper indicate that machine learning methods such as ANNs may be used to successfully as a surrogate model for reproducing these high-fidelity Multiphysics calculations. These trained neural networks may then be used in conjunction with optimization methodologies such as genetic algorithms to produce optimized fuel designs.

In this work, the above process was used to minimize vapor void in an assembly over the course of a 20 GWD/MTU depletion, while also minimizing assembly peak  $k_{inf}$  and maximizing EOC  $k_{inf}$ . Vapor void was chosen as the optimization parameter due to the strong correlation between subcooled boiling and CRUD deposition. The ability to successfully minimize vapor void in this work serves as proof-of-concept that this work could be reproduced using a code that calculates CRUD deposition in fuel assemblies, making it possible to successfully design a fuel assembly in such a way as to reduce CRUD deposition within the assembly.

This work forms a humble starting point, but there is much further work that needs to be done in order to fully apply this methodology to reducing CRUD deposition in PWRs. This work itself would be further improved by increasing the number of samples in the training library to improve the reliability and versatility of the neural network. Developing a neural network for true CRUD prediction on an assembly level will require just as many or more samples than were used in this work and will have an increased computational cost due to the extra calculations. Once a working assembly model has been developed, a full core CRUD surrogate model would be developed. These two models, in conjunction with optimization algorithms could then be used to significantly reduce CRUD deposition in PWRs.

## REFERENCES

1. V. Petrov, B. Kendrick, D. Walter, A. Manera, J. Secker, "Prediction of CRUD deposition on PWR fuel using a state-of-the-art CFD-based multi-physics computational tool," *Nuclear Engineering and Design*, **299**, pp. 95-104 (2016).
2. J.A. Turner, K. Clarno, M. Sieger, R. Bartlett, B. Collins, R. Pawlowski, R. Schmidt, and R. Summers, "The Virtual Environment for Reactor Applications (VERA): Design and Architecture," *Journal of Computational Physics*, **326**, pp. 544-568 (2016).
3. T. Rogers, J. Ragusa, S. Schultz, and R. St. Clair, "Optimization of PWR fuel assembly radial enrichment and burnable poison location based on adaptive simulated annealing," *Nuclear Engineering and Design*, **239**, pp. 1019-1029 (2009).
4. C. Martin-del-Campo, J. Francois, R. Carmona, and I. Oropeza, "Optimization of BWR fuel lattice enrichment and gadolima distribution using genetic algorithms and knowledge," *Annals of Nuclear Energy*, **34**, pp. 248-253 (2007).
5. J. Rhodes, K. Smith, and D. Lee, "CASMO-5 Development and Applications," *Proceedings of ANS Topical Meeting on Reactor Physics, (PHYSOR-2006)*, Vancouver, British Columbia, September 10-14, 2006 (2006).
6. F. Chollet, "Deep Learning with Python," Chapter 4, Manning Publications Co., Shelter Island, NY, (2018).
7. D. Goldberg "Genetic Algorithms in search, Optimization, and Machine Learning," pp 1-87, Pearson, (1989).
8. "Compute Resources," <https://projects.ncsu.edu/hpc/About/ComputeResources.php> (2019).
9. B. Andersen "Development and Assessment of Multi-Objective Optimization Utilizing Genetic Algorithms for Nuclear Fuel Assembly Design," Master's Thesis, North Carolina State University, 2018.