INL/CON-23-75158-Revision-0

# Pressurized-Water Reactor Core Design using Multi-Objective Plant Fuel Reload Optimization Platform

April 2024

Changing the World's Energy Future

Junyung Kim, Mohammad Gamal M Mostafa Abdo, Congjian Wang, Yong-Joon Choi, Juan Cristhian Luque Gutierrez, Jason Hou

Idaho National Laboratory

INL/CON-23-75158-Revision-0

# Pressurized-Water Reactor Core Design using Multi-Objective Plant Fuel Reload Optimization Platform

**Junyung Kim, Mohammad Gamal M Mostafa Abdo, Congjian Wang, Yong-Joon Choi, Juan Cristhian Luque Gutierrez, Jason Hou**

**April 2024**

**Idaho National Laboratory**
**Idaho Falls, Idaho 83415**

**http://www.inl.gov**

**Prepared for the**
**U.S. Department of Energy**
**Under DOE Idaho Operations Office**
**Contract DE-AC07-05ID14517**

# Pressurized-Water Reactor Core Design using Multiobjective Plant Fuel Reload Optimization Platform

**Junyung Kim[1], Mohammad G. Abdo[1], Congjian Wang[1], Yong-Joon Choi[1*]**
**Juan Cristhian Luque Gutierrez[2], Jason Hou[2]**

[1]Idaho National Laboratory, Idaho Falls, ID; [2] North Carolina State University, Raleigh, NC;

*[leave space for DOI, which will be inserted by ANS]*

## ABSTRACT

The U.S. Department of Energy Light Water Reactor Sustainability Program Risk-Informed Systems Analysis Pathway Plant Reload Optimization Project aims to develop an integrated, comprehensive framework offering an all-in-one solution for reload evaluations with a special focus on optimizing core design. Optimizing the fuel loading pattern is one of the most important considerations in reducing the amount of new fuel used in the core. Due to thousands of possible core configuration options, finding optimal solutions is an unachievable task for a human. The Plant ReLoad Optimization platform, which supports artificial-intelligence-based reactor core designing, is now fully capable of handling realistic problems. The Plant ReLoad Optimization platform development project aims to build a reactor core design tool that includes reactor safety and fuel performance analyses and uses artificial intelligence to support the optimization of core design solutions. The NSGA-II (Non-dominated Sorting Genetic Algorithm II) optimizer was developed and tested within RAVEN (Risk Analysis and Virtual ENvironment) to handle many constraints by using an augmented objectives methodology. The demonstration was performed with constrained multiobjective optimization of a 17 × 17 pressurized-water reactor core loading patterns to minimize fuel cost and maximize fuel cycle length.

*Keywords*: Fuel reload; multiobjective optimization; genetic algorithm; NSGA-II

## 1. INTRODUCTION

The U.S. Department of Energy Light Water Reactor Sustainability Program Risk-Informed Systems Analysis Pathway Plant ReLoad Optimization (PRLO) project aims to develop an integrated, comprehensive platform offering an all-in-one solution for reactor core reload evaluations with a special focus on optimizing the core design considering feedback from system safety analyses (i.e., thermal hydraulics) and fuel performance. [1] The optimization platform is built on the Risk Analysis and Virtual ENvironment (RAVEN) framework developed by Idaho National Laboratory. [2] RAVEN leverages contemporary artificial intelligence techniques, including the genetic algorithm (GA). The GA approach is an effective technology for optimizing fuel reloads. [3]

RAVEN's utility extends beyond optimization; it can generate input setups for multiple physical simulation codes and carry out postprocessing of simulation outcomes. This capability to integrate multiple codes allows a comprehensive framework that encompasses various physical phenomena. Consequently, RAVEN,

* yong-joon.choi@inl.gov.

acting as the controller of the optimization algorithm, serves as an inclusive and user-friendly PRLO platform that operates independently from specific tools.

The aim of this study is to create a unified and thorough PRLO platform that provides a complete solution for reload assessments, particularly emphasizing fuel optimization in order to minimize the quantity of new fuel and supporting better fuel utilization for a reduced volume of spent fuel. This PRLO platform is an enhanced arrangement for the reactor core, meticulously designed based on critical safety parameters that are essential to fulfill regulatory standards. Figure 1 gives a snapshot of the PRLO platform. The initial core design is given by RAVEN, and PARCS or SIMULATE-3K generates the equilibrium core, which is the required input for RELAP5-3D limiting design-basis accident analyses. Once the core design is found acceptable by RELAP5-3D analyses, fuel performance is assessed by TRANSURANUS for a final confirmation of an acceptable core design. This process is controlled by RAVEN along with an uncertainty analysis performed by RELAP5-3D. The PRLO platform is designed as “plug and play” where individual tools can be replaced, provided the proper interfaces with RAVEN are developed. This report focuses on coupling between RAVEN and SIMULATE-3K (without RELAP5-3D and TRANSURANUS) and presents demonstrations verifying the developed NSGA-II (Non-dominated Sorting Genetic Algorithm II) PRLO platform.

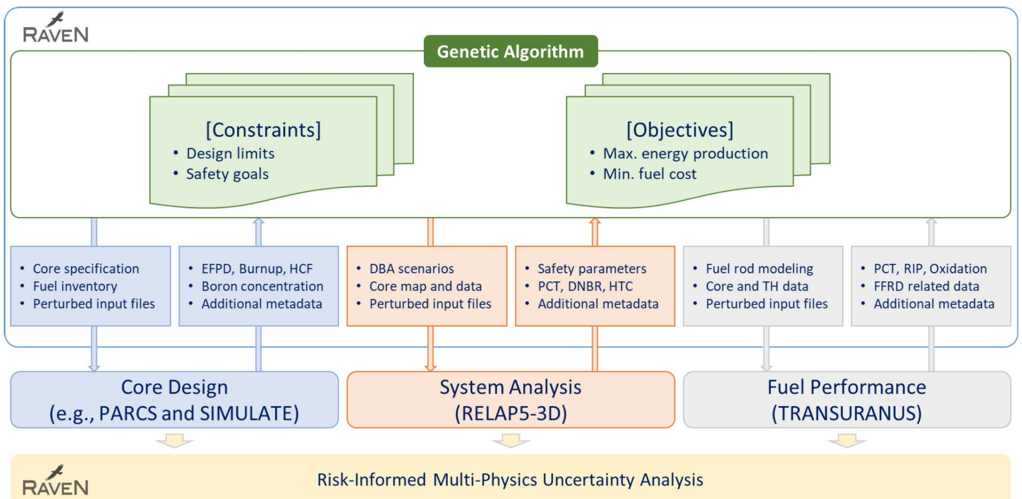



**Figure 1. High-level flow chart of Light Water Reactor Sustainability Program PRLO platform.**

# 2. MULTIOBJECTIVE OPTIMIZATION PLATFORM

## 2.1. Background of Multiobjective Optimization

When a problem involves multiple objectives, it results in a set of optimal solutions known as Pareto-optimal solutions instead of a single optimal solution. Without additional information, the solutions on the Pareto curve (or Pareto front) are assumed to be the optimal solutions, thus Pareto-optimal solutions. Traditional optimization methods, including multicriteria decision-making techniques, recommend transforming the multiobjective optimization problem (MOOP) into a single-objective optimization problem by emphasizing one Pareto-optimal solution during a single simulation. However, for a problem with multiple solutions, this approach needs to be applied multiple times, with each simulation expected to yield a different solution.

A MOOP includes a set of $n$ decision variables, $k$ objective functions, and a set of ($m$ inequality and $p$ equality) constraints. The optimization goal is:

$$\text{Min/Max}\ \boldsymbol{y}(\boldsymbol{x}) = \big(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})\big), k \geq 2 \tag{1}$$

$$\text{Subject to } g_i(\boldsymbol{x}) \leq 0, i = 1, 2, \dots, m \quad (2)$$

$$h_j(\boldsymbol{x}) = 0, i = 1, 2, \dots, p \quad (3)$$

where $\boldsymbol{x} = (x_1, \dots, x_n)$ is an $n$-dimensional decision vector in $\boldsymbol{x} \in \mathbb{R}^n$ ($\mathbb{R}$ is the set of real numbers), $\boldsymbol{y}$ is a $k$-dimensional objective vector in $\mathbb{R}^k$, $f$ defines the mapping function, $g_i$ is the $i^{\text{th}}$ inequality constraint, and $h_j$ is the $j^{\text{th}}$ equality constraint. If the following conditions are satisfied, $\boldsymbol{x}_1$ can be considered as superior to $\boldsymbol{x}_2$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are the two feasible solution vectors of the multiminimization problem:

$$f_j(\boldsymbol{x_1}) \leq f_j(\boldsymbol{x_2}) \text{ for all } j = \{1,2, \dots, k\}, \text{ and } f_j(\boldsymbol{x_1}) < f_j(\boldsymbol{x_2}) \text{ for at least one } j = \{1,2, \dots, k\} \quad (4)$$

where $k$ is the number of objective functions and $f_j(\boldsymbol{x})$ is $j^{\text{th}}$ value of an objective function for decision vector $\boldsymbol{x}$. Here, the vector value $\boldsymbol{x}$ is the Pareto-optimal solution when it is not dominated by any other feasible solutions. The collection of all Pareto-optimal solutions is a Pareto set, and the objective vectors that correspond to the Pareto set are called a Pareto front, as illustrated in Figure 2.

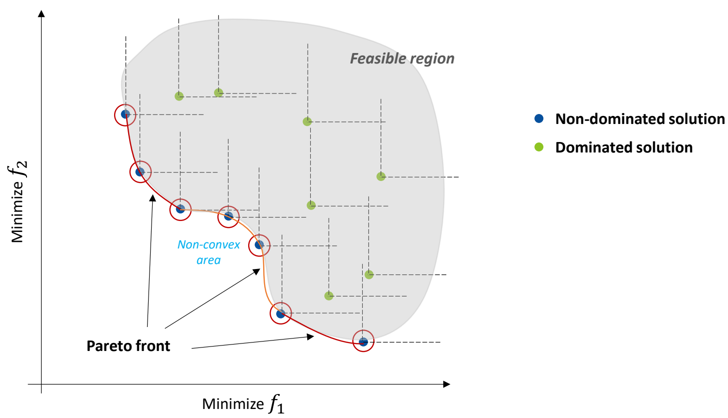



**Figure 2. Pareto dominance.**

Several multiobjective evolutionary algorithms have been proposed with different purposes and applicability. For the plant fuel reload optimization, the NSGA-II was selected for various reasons. Firstly, after testing it on multiple testing problems, NSGA-II showed an advantage in finding a wide range of solutions and converging characteristics compared to the other contemporary multiobjective evolutionary algorithms. [4] NSGA-II, initially proposed by Deb et al. in 2000 [4], is a powerful GA-based method for solving MOOPs and problems with continuous and discrete variables. Furthermore, NSGA-II has shown its efficiency in managing many engineering optimization problems. [5]

## 2.2. Non-Dominated Sorting Genetic Algorithm II

The NSGA-II optimization inherits definitions used in the GA method. For instance, the initial solution set—a population—is made of a chromosome, which is a vector of variables (called genes in NSGA-II).

### 2.2.1. Dominance Depth Method

The dominance depth method sorts nondominated solutions using the Pareto dominance concept. The nondominated sorting procedure commences by allocating the initial population's nondominated members to the first front (or so-called "rank" in NSGA-II). These members are then categorized into the first front and are removed from the initial population. The remaining population members undergo the dominance depth method. The nondominated members of the residual population are then designated the second rank and are added to the second front. This process is reiterated until all population members are grouped into different fronts based on their respective ranks.

### 2.2.2. Elitism

Elitism, also known as the elite preserving strategy, is an essential concept that NSGA-II emphasizes. It conserves a population's elite solutions by directly transferring them to the succeeding generation. Put differently, the nondominated solutions discovered in each generation proceed to the next generations until some solutions dominate them.

### 2.2.3. Crowding Distance

To assess the density of solutions surrounding a specific solution, the crowding distance is computed. It represents the average distance between two solutions on each side of the solution along each objective. When comparing two solutions that have different crowding distances, the one with the greater crowding distance is believed to exist in a less congested area. The $i^{th}$ solution's crowding distance is the average side length of the cuboid, as depicted in Figure 3. If $f_j^i$ is the $j^{th}$ value of an objective function for the $i^{th}$ solution and $f_j^{max}$and $f_j^{min}$ are the maximum and minimum values, respectively, of $j^{th}$ objective function among all the solutions, the crowding distance of $i^{th}$ solution is defined as the average distance of the two nearest solutions on either side, as given in Equation (5):

$$cd(i) = \sum_{i=1}^{k} \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{max} - f_j^{min}} \tag{5}$$

where $k$ is the number of objective functions.

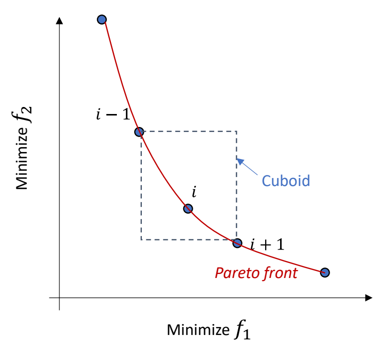



**Figure 3. Cuboid with neighboring solutions for calculating crowding distance.**

### 2.2.4. Survivor Selection of Non-Dominated Sorting Genetic Algorithm II

The population for the next generation was selected using a tournament selection operator, which uses the rank of chromosomes and their crowding distances for selecting ones out of chromosomes for the next generation. The survivor selection process is:

1) Select chromosomes that do not violate any constraints.
2) If both chromosomes have different ranks, the better ranked one is selected for the next generation.
3) If both the chromosomes are of the same ranks, the one with the higher crowding distance is selected for the next generation.

### 2.2.5. Procedures

The NSGA-II procedure begins with generating an initial population P(t=0) of size N, where t represents the number of iterations. Then a new population Q(t=0) (offspring) is created after performing crossover and mutation operations on the population P(t=0). After that, the population P(t=0) and Q(t=0) are combined to form a new population R(t=0) (which is the size of 2 × N), and the nondominated sorting procedure is performed on R(t=0). Then the population members of R(t=0) are ranked into different fronts according to their nondomination levels.

The next process is to select N members from R(t=0) to create the next population P(t=1). If the size of the first front is greater than or equal to N, only N members are selected from the least crowded region of the first front to form P(t=1). On the contrary, if the size of the first front is less than N, the chromosomes of first front are directly transferred to the next generation, and the remaining members are taken from the least crowded region of the second front and added to P(t=1). If the size of P(t=1) is still less than N, the same procedure is followed for the next consecutive fronts until the size of P(t=1) becomes equal to N. The populations of P(t=2), P(t=3), …, are constructed following same procedure until the stopping criteria are satisfied. The NSGA-II procedure is shown in Figure 4.

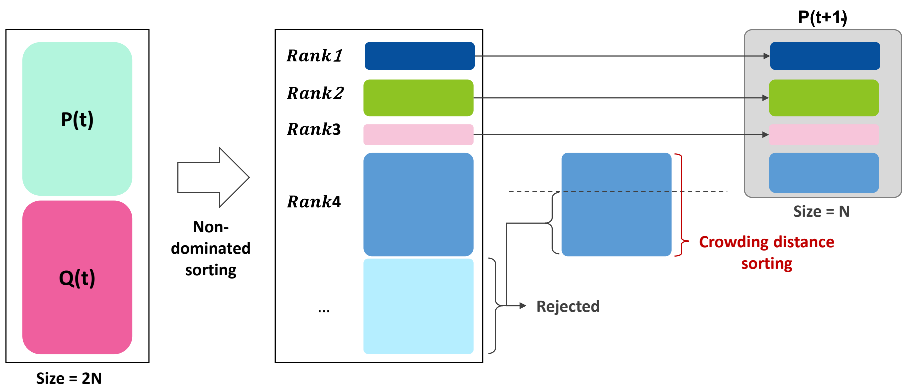



**Figure 4. NSGA-II procedure.**

### 2.2.6. Constraint Handling

In this paper, the enhancement of the multiobjective PRLO platform using NSGA-II concentrated on integrating enhanced capabilities for handling larger constraints. The realistic fuel reload problem has a larger number of constraints and their violations will be also large. The high degree of violation of constraints increases complexity while solving the MOOP. To reduce such complexity, an augmented objectives concept is introduced. [6] This concept includes a static penalty term inside of the objective to reduce the degree of violation of the constraints, which represents that the population will have lower-level fitness in GA methodology. In other words, using the objective that already includes the violation (i.e., static penalty) will ease handling large number of constraints and their violations.

The static penalty term, $\omega_i(x_i)$, could be defined based on the degree of violation of the constraints as:

$$\omega_i(x_i) = \begin{cases} |g(x_i)|, \text{if } g(x_i) < 0 \\ 0, \text{otherwise} \end{cases} \tag{6}$$

where $g(x_i)$ is the degree of violation that is the function of difference between the constraints and actual values. For each constraint, a penalty weight, $\Omega_k(x_i)$, could be given as:

$$\Omega_k(x_i) = \sum_{i=1}^{m} P_i \omega_i(x_i) \tag{7}$$

where $P_i$ is a penalty weight for each constraint. The augmented objectives, $f_k^*(x_i)$, is then:

$$f_k^*(x_i) = f_k(x_i) + \Omega_k(x_i) \tag{8}$$

where $f_k(x_i)$ is original objectives. The original objectives could be separated from augmented objective once optimization is completed.

As an example, the problem is defined to find maximum effective full power day with minimum fuel cost, which is max-min MOOP. By using the augmented objective concept, the problem could be transformed into min-min MOOP by adding a static penalty in each objective, which has a minimum degree of violation of the constraints in each original objective. Figure 5 shows an example of extracting original objectives from augmented objective. ArtObjOne and ArtObjTwo are the two augmented objectives, which included the degree of violation of the constraints from two objectives fuel cycle length and fuel cost. Once augmented objectives give a min-min MOOP value (left of Figure 5), original objectives could be achieved by removing the degree of violation of the constraints (right of Figure 5).

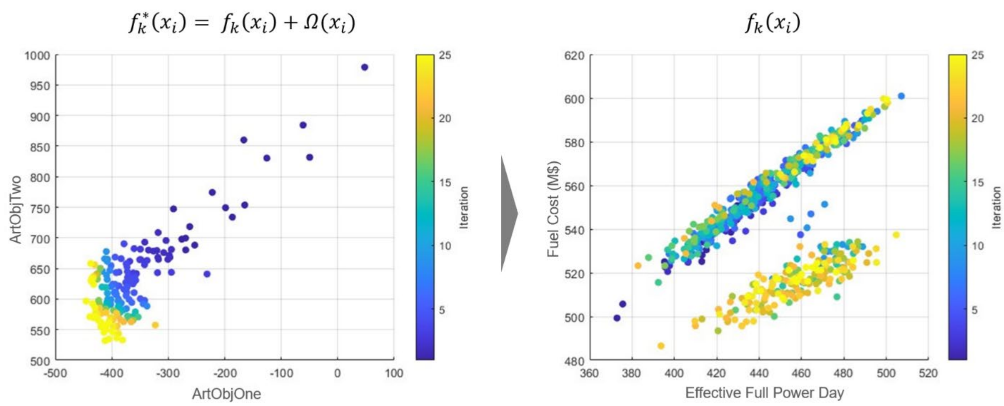



**Figure 5. Mapping of real objectives (right) from augmented objectives (left). ArtObjOne and ArtObjTwo imply augmented objectives combining original objectives and degree of violation of the constraints.**

## 3. DEMONSTRATION OF CORE DESIGN WITH MULTIOBJECTIVE OPTIMIZATION

### 3.1. Problem Statement

A demonstration was performed for the constrained multiobjective optimization of a pressurized-water reactor (PWR) loading pattern. The goal is to minimize the fuel cost and maximize the cycle length, while complying with all actual reactor design constraints. The target constraint values for the design are $F_Q$ (heat flux hot channel factor) < 2.1; $F_{\Delta H}$ (nuclear enthalpy rise hot channel factor) < 1.48; and peak critical boron concentration < 1300 ppm. The reactor used was a generic three-loop Westinghouse PWR with a 17 × 17 core model consisting of 157 fuel assemblies (FAs) with five different fuel types is given in Table I. An initial 1/8 core loading pattern with 35 positions were set with six different types of assemblies, including the reflectors, as illustrated in Figure 6. Positions 21, 26:27, and 30:35 are fixed with reflectors, leaving the remaining 26 positions for the five different types of FAs. These loading patterns were encoded as a chromosome of NSGA-II in a RAVEN input file.

**Table I. FAs inventory for the initial PWR core.**

| Fuel type | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Enrichment (wt%) | Reflector | 2 | 2.5 | 2.5 | 3.2 | 3.2 |
| Burnable poison | | None | None | 16 Gd rods | None | 16 Gd rods |
| Unitary cost ($)[1] | 0 | 2.69 | 3.25 | 3.25 | 4.04 | 4.04 |

[1] https://world-nuclear.org/information-library/economic-aspects/economics-of-nuclear-power.aspx

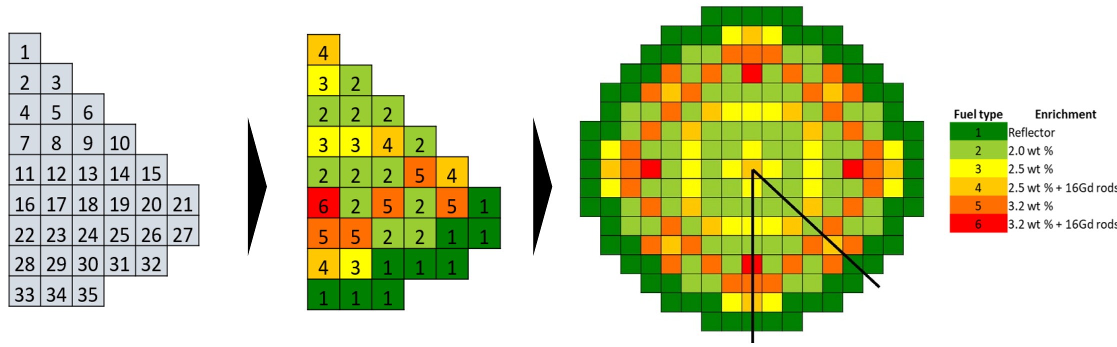



**Figure 6. One-eight symmetry of PWR loading pattern with 35 locational positions and six different types of assemblies.**

The cycle lengths were taken from the SIMULATE-3K output. The fuel cost is calculated by counting how many different FAs were used for the given loading pattern then multiplying them by their corresponding unitary value as given in Table I. The constraints were handled as part of the optimization of two augmented objectives with a static penalty term to account for the degree of violation of the constraints. As shown in Equation (9) and (10), ArtObjOne is the augmented cycle length objectives and ArtObjTwo is the augmented fuel cost objectives.

$$\text{ArtObjOne} = -\text{Cycle Length} + \sum_{j=1}^{m=3} w_j \max\left(0, penalty_j\right) \tag{9}$$

$$\text{ArtObjTwo} = \text{Fuel Cost} + \sum_{j=1}^{m=3} u_j \max\left(0, penalty_j\right) \tag{10}$$

A minimization-minimization framework was used to keep the same expression for calculating the degree of violation in both augmented objectives, thus a negative sign for the cycle length was used.

### 3.2. Workflow for Reactor Core Optimization in RAVEN

The RAVEN xml input file was used to define the number of generations, population size, parent selection, crossover, and mutation operator. RAVEN first checks for any input error, then samples FA location mapping and generating a SIMULATE-3K input file. RAVEN parses specified variables from the SIMULATE-3K output file, and the output file was used as part of the NSGA-II optimizer. This whole process will be repeated until the given number of iterations is completed. Figure 7 shows the flowchart of the SIMULATE-3K and RAVEN coupling interface.

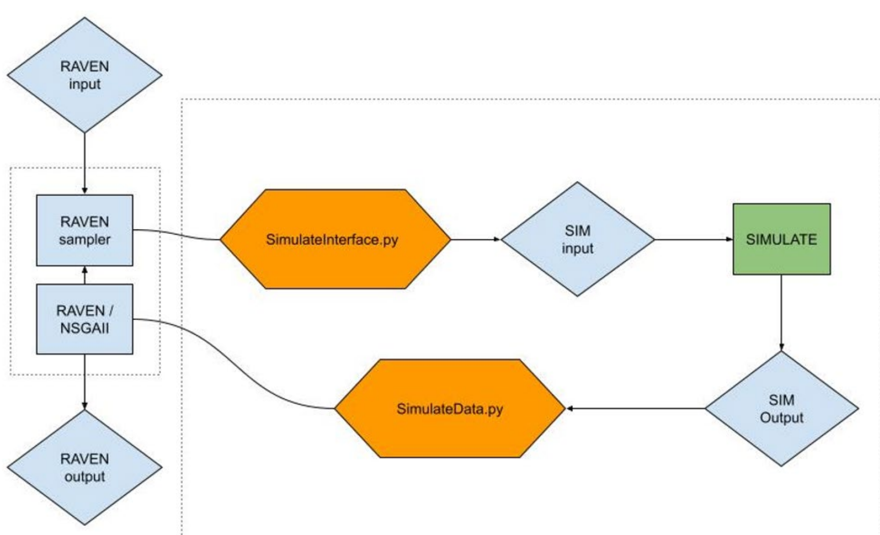


**Figure 7. Flowchart of SIMULATE-3K/RAVEN coupling interface.**

Augmented objective functions with coefficients $u_1 = w_1 = 2{,}800$ includes boron concentration coefficient $j$ = 1. For augmented objective functions $u_2 = w_2 = 700$ and $u_3 = w_3 = 700$ has peaking factor coefficient of $j$ = 2 and $j$ = 3, respectively. It is noted that, in the augmented objectives, the weights assigned for each constraint are intuitively selected based on the evaluation of the user and are heavily dependent on the nature of the problem. For the current demonstration, the boron concentration is tightly related to the objectives, therefore a higher weight was needed. One point crossover and tournament ranking were used in the crossover and selection operators, respectively.

## 3.3. Optimization Results

A multiobjective optimization maximizing cycle length and minimizing fuel cost was performed for a population size of 50 with 50 generations. Figure 8 shows the feasible region and Pareto front with optimized solutions. Generally, convergence towards one region is observed as the number of iterations (i.e., generations) increases. A total of 2,500 chromosomes encoding a loading pattern were generated in serial, from which 1,772 are unique. The feasible region, which contains chromosomes that comply with all the constraints, is composed of 40 chromosomes, of which 11 are part of the Pareto frontier that could be the optimized solutions. The frontier values are identified as #1 from the lowest to #11 at the highest position. The core loading patterns generated by the NSGA-II optimizer were very close to a realistic nuclear reactor core shape: lower enrichment fuels in the inner region and higher enriched fuel in the outer region, then another region of lower enriched fuel region, which establishes a low-leakage loading pattern. This pattern is a typical core loading strategy of placing burned FAs in an outer core location to reduce neutron fluence to the reactor core vessel, which extends the vessel lifetime and avoids pressurized thermal shock. [7]

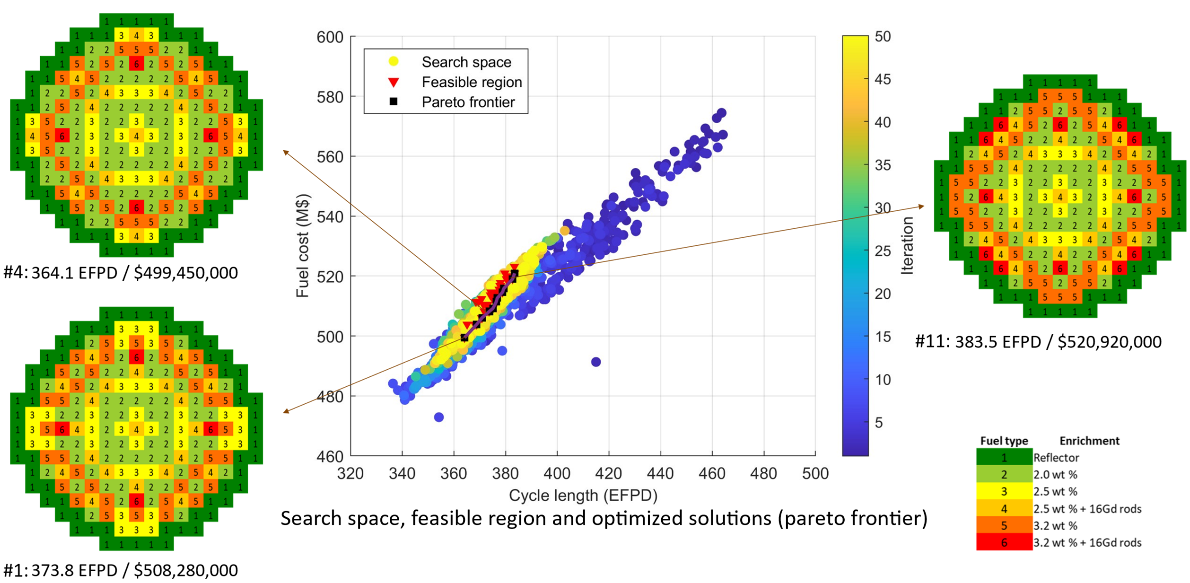
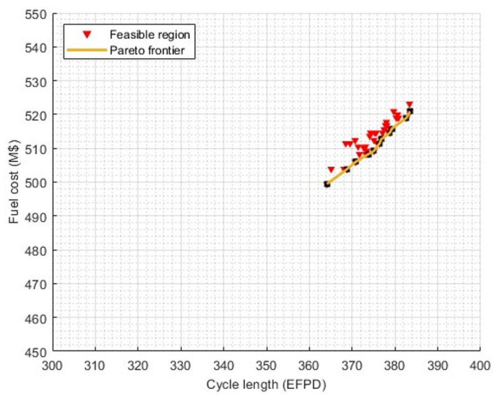



**Figure 8. Search space and feasible region (left) and Pareto frontier (right) for NSGA-II optimization (population size = 50, generations = 50).**

## 3.4. Sensitivity Study on the Population Size

A sensitivity study was conducted with respect to the population size. An increased population size of 100 was compared with a population size of 50. All other parameters remained same in both cases: selection, crossover, mutation types, and number of generations (i.e., 50). The results show a significant increase in the feasibility region and number of Pareto frontiers, as shown in Figure 9 and Table 2. The number of total solutions was increased from 2,500 to 5,000, and the possibility of generating a unique solution was increased from 70.88% to 93.4%. The optimized Pareto frontier solutions were significantly increased from 11 to 77 solutions. It could be concluded that an increased population size could generate more optimized core reloading patterns. However, the computational burden would be increased as the population increased. While the objective (e.g., minimum fuel cost and maximum fuel cycle length) may converge to the user's goal, it is recommended to select a reasonable number of populations to reduce computational burdens.



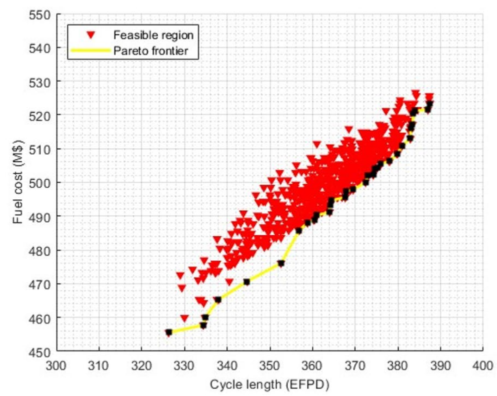




**Figure 9. Feasible region for population size of 50 (left) and 100 (right).**

**Table 2. Performance comparison with respect to population size in NSGA-II.**

| Population size | 50 | 100 |
|---|---|---|
| **Total solutions generated (∝ runtime if serial)** | 2,500 | 5,000 |
| **Unique solutions generated** | 1,772 / 2,500 = 70.88% | 4,670 / 5,000 = 93.4% |
| **Feasibility regions share** | 40 / 2,500 = 1.6% | 903 / 4,670 = 19.33% |
| **Pareto front share** | 11 / 1,772 = 0.62% | 77 / 4,670 = 1.65% |

# 4. CONCLUSIONS

In this paper, we presented the PRLO platform, designed for artificial-intelligence-driven reactor core design to tackle real-world challenges. The objective of the PRLO Platform is to create a comprehensive reactor core design tool incorporating safety and fuel performance analyses, leveraging artificial intelligence for optimizing core design solutions. The NSGA-II optimizer was developed and tested within the RAVEN framework, employing an augmented objectives methodology to handle numerous constraints. A demonstration was conducted, showcasing constrained multiobjective optimization for a 17 × 17 PWR core loading pattern with the goal of minimizing fuel cost and maximizing fuel cycle length. The SIMULATE-3K to RAVEN coupling interface was built and tested for NSGA-II optimizer and actual reactor design parameters were applied as constraints, and an augmented objectives method was used. Optimization with population and generation sizes of 50 provided reasonable results, including a low-leakage core configuration, which is preferable for a realistic core loading pattern. From the sensitivity study on the population size, a larger (i.e., 100) population case generated significant improvements in potential optimal solutions. For future works, a full-scale demonstration of a PWR core design to minimize the volume of new fuel, including core and system safety analysis considerations will be conducted, and additional capabilities of the multiobjective optimization methodology by applying adaptive weighting and searching algorithms and advanced termination criteria will be developed in the RAVEN framework.

## REFERENCES


[1] Y.-J. Choi, M. G. Abdo, G. Palamone, S. Heagy, C. Frepoli, K. Ogujiuba, N. Rollins, G. Deplipei and J. Hou, "Development and Demonstration of a Risk-Informed Approach to the Regulatory Required Fuel Reload Safety Analysis," Idaho National Laboratory (INL), Idaho Falls, ID, USA, 2022.

[2] C. Rabiti, A. Alfonsi, J. Cogliati, D. Mandelli, C. Wang, P. W. Talbot, M. G. Abdo, D. J. McDowell, R. K. Yoshiura, R. Kinoshita, S. Sen and D. P. Maljovec, "RAVEN User Manual (INL/EXT-15-34123, Revision 8)," Idaho National Laboratory (INL), Idaho Falls, ID, USA, 2023.

[3] B. D. Andersen, "A Machine-Learning Based Approach to Minimize Crud Induced Effects in Pressurized Water Reactors," North Carolina State University, Raleigh, NC, USA, 2021.

[4] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 20, pp. 182-197, 2002.

[5] P. Wang, K. Ye, X. Hao and J. Wang, "Combining multi-objective genetic algorithm and neural network dynamically for the complex optimization problems in physics," *Scientific Reports,* vol. 13, no. 1, p. 880, 2023.

[6] G. T. Park, "Multiobjective pressurized water reactor reload core design by nondominated genetic algorithm search," *Nuclear Science and Engineering,* vol. 124, no. 1, pp. 178-187, 1996.

[7] Y.-J. Kim, T. Downer and A. Sesonske, "Optimization of Coe Reload Design for Low-Leakage Fuel Management in Pressurized Water Reactors," *Nuclear Science and Engineering,,* vol. 96, pp. 85-101, 1987.