

Light Water Reactor Sustainability Program

Pressurized-Water Reactor Core Design Demonstration with Genetic Algorithm Based Multi-Objective Plant Fuel Reload Optimization Platform



September 2023

U.S. Department of Energy

Office of Nuclear Energy

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

Pressurized-Water Reactor Core Design Demonstration with Genetic Algorithm Based Multi- Objective Plant Fuel Reload Optimization Platform

**Junyung Kim¹
Mohammad Abdo¹
Yong-Joon Choi¹
Juan Cristhian Luque Gutierrez²
Jason Hou²
Christopher Gosdin³
Jarrett Valeri³**

**¹ Idaho National Laboratory
² North Carolina State University
³ FPoliSolutions, LLC**

September 2023

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

Page intentionally left blank

EXECUTIVE SUMMARY

The U.S. nuclear industry faces a challenge in maintaining required levels of safety while ensuring economic competitiveness to stay in business. Safety remains a key parameter for all aspects of light-water reactor nuclear power plant operations. Safety can become more economical by using a risk-informed ecosystem, such as the one being developed by the Risk-Informed Systems Analysis Pathway under the U.S. Department of Energy Light Water Reactor Sustainability Program. This program promotes a wide range of research and development activities to maximize both the safety and economic efficiency of nuclear power plants through improved scientific understanding, especially given many plants are now considering second license renewals.

The Risk-Informed Systems Analysis Pathway has two main goals: deploy methodologies and technologies that better represent safety margins and cost and safety factors and develop advanced applications that enable cost-effective plant operation.

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.

This report summarizes development and demonstration activities of the Plant Reload Optimization Platform built in the Risk Analysis and Virtual ENvironment, specifically:

- Improvement of multi-objective non-dominated sorting genetic algorithm II to handle large size of objectives and constraints
- Demonstration of pressurized-water reactor core design with the non-dominated sorting genetic algorithm II multi-objective optimization platform
- Single-objective, optimized core design, including system analysis and fuel performance feedback.

Page intentionally left blank

CONTENTS

EXECUTIVE SUMMARY	iii
ACRONYMS.....	ix
1. INTRODUCTION.....	11
2. IMPROVEMENT OF MULTIOBJECTIVE OPTIMIZATION PLATFORM.....	14
2.1 Constraints Handling.....	14
2.2 Demonstration of Large Number Constraints Multiobjective Optimization Problem	15
3. DEMONSTRATION OF CORE DESIGN WITH MULTIOBJECTIVE OPTIMIZATION.....	21
3.1 Problem Statement	21
3.2 Methodology	22
3.2.1 SIMULATE-3K-RAVEN Coupling Interface	23
3.2.2 Workflow for Core Optimization in RAVEN.....	24
3.3 Optimization Results.....	25
3.3.1 NSGA-II Optimization Results.....	25
3.3.2 Sensitivity Study on the Population Size	27
3.3.3 Comparison Between Single-Objective GA and Multiobjective NSGA-II	28
4. OPTIMIZATION WITH MULTIPHYSICS FEEDBACK.....	32
4.1 Multiphysics Coupling.....	32
4.1.1 Core-Physics Data.....	32
4.1.2 Thermal-Hydraulics Data.....	33
4.1.3 TRANSURANUS Output.....	34
4.1.4 RAVEN-TRANSURANUS Coupling Interface.....	34
4.2 Demonstration with Large-Break Loss-of-Coolant Accident Case	37
4.2.1 Problem description	37
4.2.2 Results and discussion	38
5. SUMMARY AND FUTURUE WORKS	41
6. REFERENCES	42

FIGURES

Figure 1. Technology roadmap of PRLO project.	11
Figure 2. High-level flow chart of Light Water Reactor Sustainability Program developed PRLO platform. [1]	13
Figure 3. 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.	15

Figure 4. Exemplary solutions (accepted and eliminated) to the multisum problem with explicit and implicit constraints.....	16
Figure 5. Multisum model for larger number of constraints in PRLO platform.....	16
Figure 6. Implicit and explicit constraint modeling in PRLO platform.....	17
Figure 7. Multisum test results in PRLO platform.....	18
Figure 8. Constraints change over iteration.....	19
Figure 9. Pin map of a fuel assembly with (left) and without (right) Gd burnable poison loading.....	21
Figure 10. A quarter symmetry of a core loading pattern.....	22
Figure 11. Chromosome encoding a one-eight symmetry of PWR loading pattern.....	22
Figure 12. Flowchart of SIMULATE-3K/RAVEN coupling interface.....	24
Figure 13. User input file used by PRLO platform to sample SIMULATE-3K input files.....	25
Figure 14. Search space and feasible region (left) and Pareto frontier (right) for NSGA-II optimization (population size = 50, generations = 50).....	25
Figure 15. Sample Pareto optimal solution for NSGA-II run with population size of 50 and 50 generations (Position #1).....	26
Figure 16. Sample Pareto optimal solution for NSGA-II run with population size of 50 and 50 generations (Position #4).....	26
Figure 17. Sample Pareto optimal solution for NSGA-II run with population size of 50 and 50 generations (Position #11).....	27
Figure 18. Feasible region for population size of 50 (left) and 100 (right).....	28
Figure 19. Search space and feasible region (left) and Pareto frontier (right) from single-objective GA optimization (population size = 50, generations = 50).....	29
Figure 20. Sample Pareto optimal solution for GA run with population size of 50 and 50 generations (Position #1).....	30
Figure 21. Sample Pareto optimal solution for GA run with population size of 50 and 50 generations (Position #5).....	30
Figure 22. Sample Pareto optimal solution for GA run with population size of 50 and 50 generations (Position #6).....	31
Figure 23. Comparison of GA and NSGA-II optimal loading patterns.....	31
Figure 24. Sample interface file for TRANSURANUS from core design.....	33
Figure 25. Example of keywords in TRANSURANUS input file.....	35
Figure 26. Different methods of building time-dependent TRANSURANUS cards.....	36
Figure 27. Schematic diagram of data flow in multiphysics single objective optimization demonstration.....	38
Figure 28. Evolution of EFPD and PCT during multiphysics optimization demonstration.....	39
Figure 29. Evolution of core design criteria in multiphysics optimization demonstration.....	39
Figure 30. Evolution of core design during multiphysics optimization demonstration (1 st 20 th and final, from top to bottom).....	40

TABLES

Table 1. NSGA-II parameters for the multisum problem in PRLO platform.	20
Table 2. A list of optimal solutions of the multisum problem after 60 iterations.	20
Table 3. Fuel assembly inventory for the initial PWR core.	21
Table 4. Performance comparison with respect to population size in NSGA-II.	28
Table 5. Setting of PRLO platform for multiphysics optimization demonstration.	37

Page intentionally left blank

ACRONYMS

ATF	Accident-tolerant fuel
EFPD	effective full power day
FY	fiscal year
GA	genetic algorithm
LBLOCA	large-break loss-of-coolant accident
MOOP	multi-objective optimization problem
NSGA-II	non-dominated sorting genetic algorithm
PCT	peak cladding temperature
PRLO	plant reload optimization
PWR	pressurized-water reactor
RAVEN	Risk Analysis and Virtual ENvironment

Page intentionally left blank

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 analysis (i.e., thermal hydraulics) and fuel performance. [1] The main driving force behind the optimization platform is the Risk Analysis and Virtual ENvironment (RAVEN) developed by the 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 versatility of RAVEN enables seamless integration with other system codes related to core design, fuel performance, and systems thermal hydraulic analysis. This capability to integrate multiple codes allows a comprehensive framework that encompasses various physical phenomena. Consequently, RAVEN, 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 utilization of fuel that results in 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 illustrates the project’s technology roadmap and research strategy spanning multiple fiscal years (FYs), comprising four distinct phases of research and development.

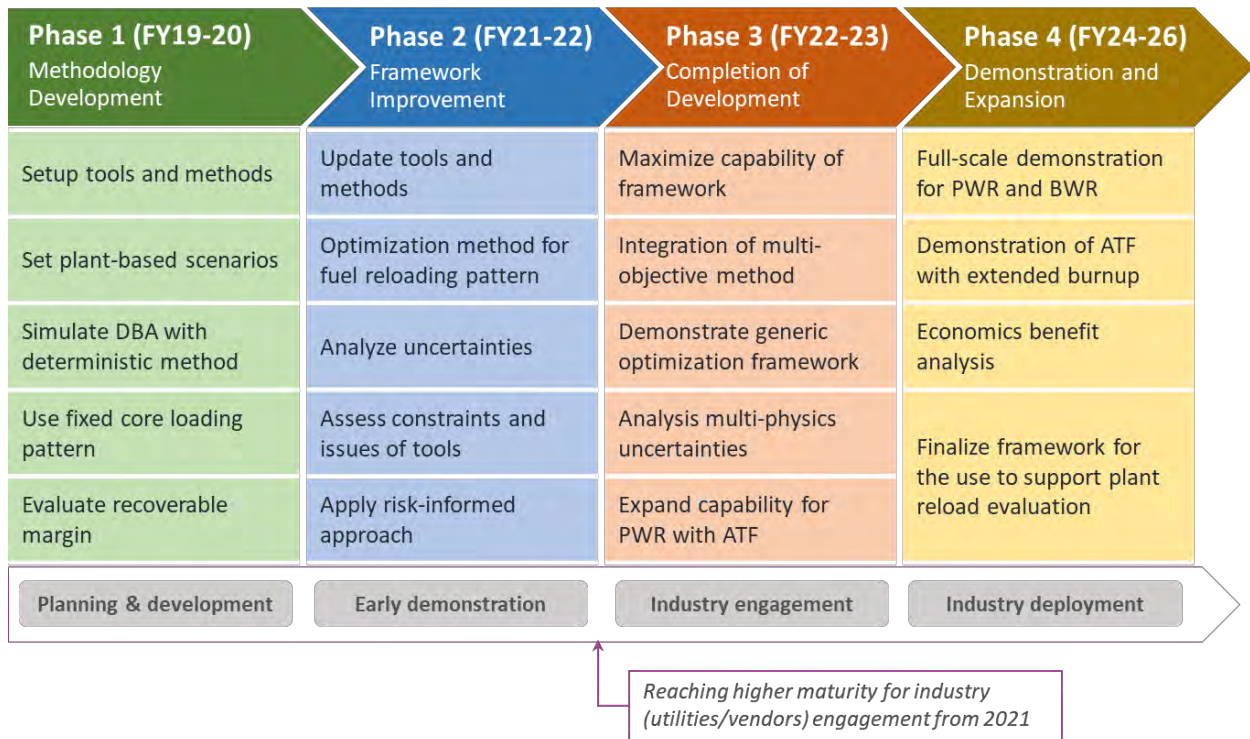


Figure 1. Technology roadmap of PRLO project.

Phase 1—Methodology Development (FY-2019 to FY-2020): Available tools and methods were investigated and tested. Design-basis accident (DBA) scenarios were simulated via traditional deterministic methods by using the Reactor Excursion and Leak Analysis Program (RELAP5) 3D thermal-hydraulics analysis code developed at Idaho National Laboratory. Simulations used fixed core loading and evaluated recoverable margins.

Phase 2—Framework Improvement (FY-2021 to FY-2022): The key activities include:

- The demonstration of the PRLO platform for a generic pressurized-water reactor (PWR) using 10 limiting DBA scenarios
- The development and deployment of the optimization process using GA in RAVEN
- The development of an improved RAVEN infrastructure for the performance of neutronics and thermohydraulic analyses
- The identification of limiting DBA scenarios for the evaluation of the transition from a deterministic to risk-informed approach for fuel analyses.

Phase 3—Demonstration of Benefits (FY-2022 to FY-2023): The project continued the progression through the demonstration phase where the framework was enhanced with additional and extended capabilities to support regulatory-required fuel safety analyses and to allow additional economic benefits from fuel reload optimization. The development and deployment of the multiobjective optimization process using the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) in RAVEN were completed, and the developed NSGA-II optimization PRLO platform was demonstrated with a constrained multiobjective optimization of a PWR core loading pattern.

Phase 4—Improvement and Expansion (FY-2024 to FY-2026): The framework capabilities will be further demonstrated for full-scale analyses performed by nuclear power plants. Equilibrium scenarios will be demonstrated for multiple reactor types (i.e., PWRs, boiling-water reactors) as well as accident-tolerant fuel (ATF) usage with extended burnup. Framework development will be finalized, and a pathway will be established to deploy developed methods and tools to industry.

The major accomplishments in FY-2023 are:

- The development of a multiobjective optimization capability by integrating NSGA-II methodology
- The demonstration of new capabilities by successfully designing an optimized PWR core [4]
- The expansion of capabilities to high-burnup ATF for a PWR core design, including fuel performance analysis for equilibrium fuel cycle and explicitly considered uncertainties [5]
- The initiation of an industrial pilot demonstration project with a utility partner (i.e., Constellation LLC) and computer simulation tool vendor (i.e., Studsvik LLC).

Upon finalizing the research, the following benefits will be obtained from the developed PRLO platform:

- The PRLO platform will significantly simplify the required process of core reload evaluation performed for each fuel reload because it integrates all the required tasks into one seamless automated process
- The PRLO platform will support flexible plant operations with increased or decreased reactor power levels following the fluctuating demand driven by the integration of renewables
- The PRLO platform will be capable of performing all necessary evaluations of ATF, including the optimization of core design.

The PRLO platform was developed to handle multiple objectives and constraints such as fuel cycle length (e.g., an extension from 18 to 24 months), fuel enrichment, burnable poisons, core design limits

(e.g., peaking factors and boron concentration), safety parameters (e.g., peak cladding temperature and departure of nucleate boiling rate). To do this, the Risk-Informed Systems Analysis Pathway GA-based PRLO platform uses the following individual computational tools coupled with RAVEN to provide safety feedback during core designing:

- PARCS and SIMULATE-3K for core design
- RELAP5-3D for system response analysis (i.e., thermal-hydraulic analysis)
- TRANSURANUS for fuel performance analysis.

Figure 2 gives a snapshot of the PRLO platform. 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 DBA 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 choice of the analytical tools herein is for demonstrative purposes. 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 describes coupling between RAVEN, SIMULATE-3K, and TRANSURANUS as well as presents demonstrations verifying the developed NSGA-II PRLO platform.

It is noted that usage of SIMULATE-3K is limited within the agreement with Studsvik LLC.

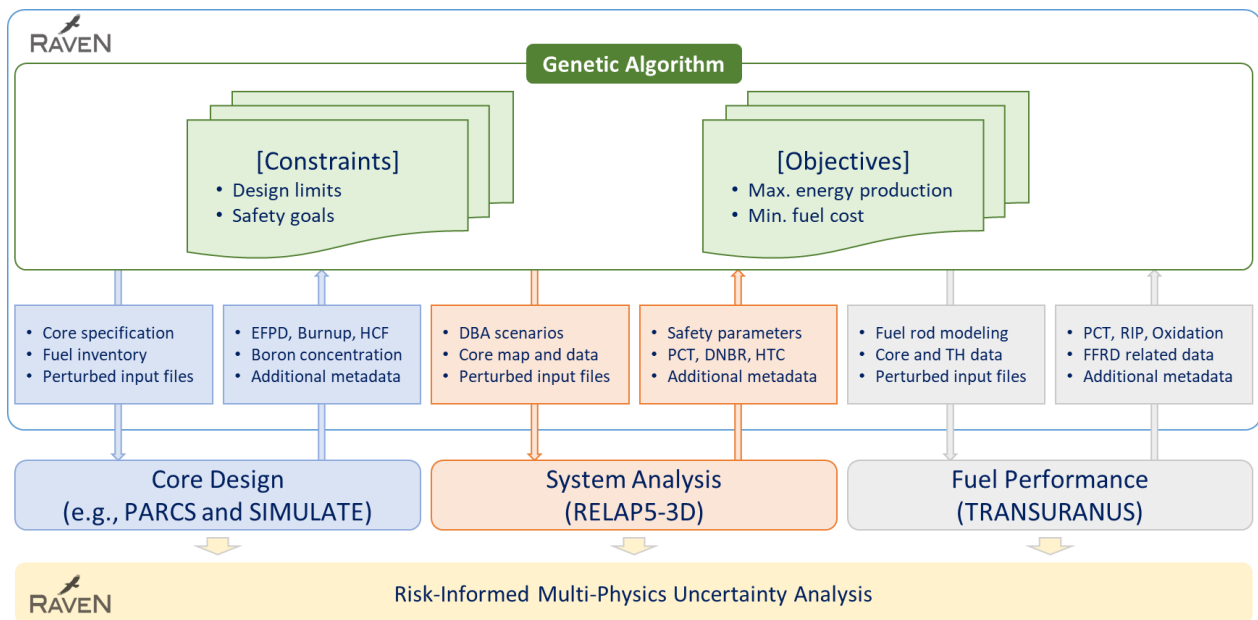


Figure 2. High-level flow chart of Light Water Reactor Sustainability Program developed PRLO platform. [1]

2. IMPROVEMENT OF MULTIOBJECTIVE OPTIMIZATION PLATFORM

2.1 Constraints Handling

Improving the NSGA-II multiobjective PRL0 platform was focused on implementing the larger constraints handling capabilities. Realistic fuel reload problem has larger number of constraints and their violations will be also large. The high degree of violation of constraints increases complexity while solving the multiobjective optimization problem (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:

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

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:

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

where P_i is a penalty weight for each constraint.

The augmented objectives, $f_k^*(x_i)$, is then:

$$f_k^*(x_i) = f_k(x_i) + \Omega_k(x_i)$$

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 fuel cycle length (i.e., EFPD¹) with minimum fuel cost, which is max-min MOOP. By using augmented objective concept, the problem could be transformed into min-min MOOP by adding static penalty in each objective which has minimum degree of violation of the constraints in each original objective. Figure 3 shows example of extracting original objectives from augmented objective. ArtObjOne and ArtObjTwo are the two augmented objectives, which included 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 3), original objectives could be achieved by removing the degree of violation of the constraints (right of Figure 3).

¹ Effective full power day

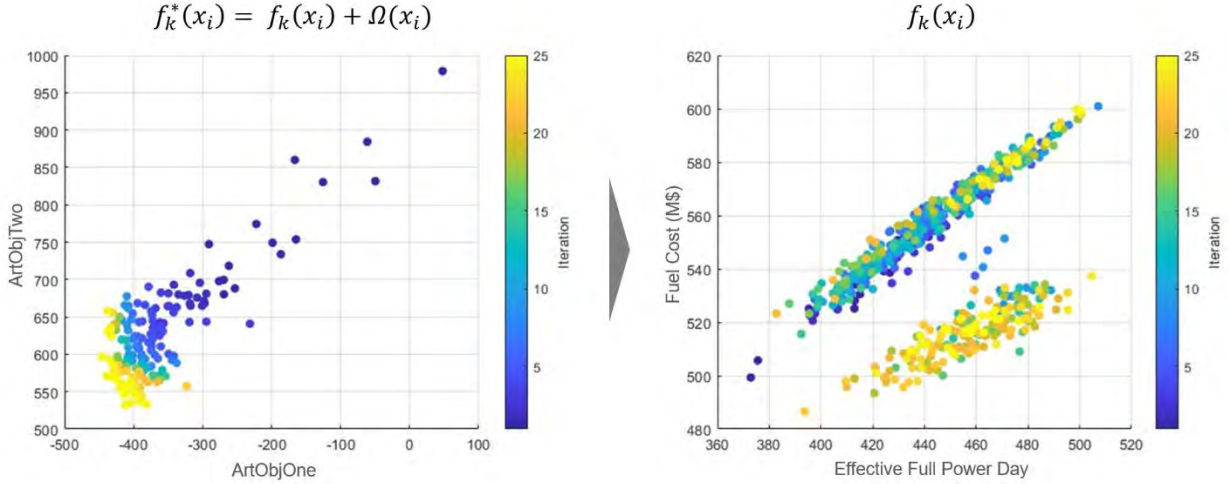


Figure 3. Mapping of real objectives (right) from augmented objectives (left). ArtObjOne and ArtObjTwo imply augmented objectives combining original objectives and degree of violation of the constraints.

2.2 Demonstration of Large Number Constraints Multiobjective Optimization Problem

The multiobjective local sums problem was developed for the larger number of constraints handling demonstration purpose. The demonstration involved a MOOP, the minimization and maximization of multiple local sums with multiple constraints. Two different implicit constraints were set as a minimum of 200 for $f_1(x)$ and at least 16 for $f_2(x)$. The general form of the problem solved in the demonstration can be expressed as:

- Minimize $f_1(\mathbf{x}) = \sum_{i=1}^8 i \times x_i$
- Maximize $f_2(\mathbf{x}) = \sum_{i=1}^2 i \times x_i$
- Minimize $f_3(\mathbf{x}) = \sum_{i=3}^4 i \times x_i$
- Explicit constraint: $x_3 + x_4 < 10$
- Implicit constraint: $f_1(\mathbf{x}) < 200$
- Implicit constraint: $f_2(\mathbf{x}) > 16$

where $x_i \sim \mathcal{U}^d[2, 9]$ is sampled from a discrete uniform distribution. Figure 4 shows examples of accepted and eliminated solutions. For instance, solution $\mathbf{x} = (9, 7, 3, 2, 8, 6, 4, 5)$ is an accepted solution that satisfies all constraints. However, solution $\mathbf{x} = (4, 5, 6, 3, 7, 9, 2, 8)$ violates two implicit constraints, so this solution will be eliminated during the survivor selection process.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
9	7	3	2	8	6	4	5

Objective

$$f_1(x) = 1 \times 9 + 2 \times 7 + 3 \times 3 + 4 \times 2 + 5 \times 8 + 6 \times 6 + 7 \times 4 + 8 \times 5 = 184$$

$$f_2(x) = 1 \times 9 + 2 \times 7 = 23$$

$$f_3(x) = 3 \times 3 + 4 \times 2 = 15$$

Constraints

$$x_3 + x_4 = 5 < 10$$

$$f_1(x) = 184 < 200$$

$$f_2(x) = 23 > 16$$

Chromosome accepted

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
4	5	6	3	7	9	2	8

Objective

$$f_1(x) = 1 \times 4 + 2 \times 5 + 3 \times 6 + 4 \times 3 + 5 \times 7 + 6 \times 9 + 7 \times 2 + 8 \times 8 = 211$$

$$f_2(x) = 1 \times 4 + 2 \times 5 = 14$$

$$f_3(x) = 3 \times 6 + 4 \times 3 = 30$$

Constraints

$$x_3 + x_4 = 9 < 10$$

$$f_1(x) = 211 < 200$$

$$f_2(x) = 14 > 16$$

Constratint Violated!

Chromosome eliminated

Figure 4. Exemplary solutions (accepted and eliminated) to the multisum problem with explicit and implicit constraints.

Figure 5 and Figure 6 show the enhanced multisum model and implicit and explicit constraint models in PRLO platform, respectively. The PRLO platform input deck for the multisum optimization problem is given in Appendix B.

```
def evaluate(Inputs):
    Sum = 0
    LocalSum1 = 0
    LocalSum2 = 0

    for ind,var in enumerate(Inputs.keys()):
        # write the objective function here
        Sum += (ind + 1) * Inputs[var]
        if (ind == 0) or (ind == 1):
            LocalSum1 += (ind + 1) * Inputs[var]
        if (ind == 2) or (ind == 3):
            LocalSum2 += (ind + 1) * Inputs[var]
    return Sum[:], LocalSum1[:], LocalSum2[:]

def run(self,Inputs):
    """
    RAVEN API
    @ In, self, object, RAVEN container
    @ In, Inputs, dict, additional inputs
    @ Out, None
    """
    self.obj1,self.obj2,self.obj3 = evaluate(Inputs)
```

Figure 5. Multisum model for larger number of constraints in PRLO platform.

```

import numpy as np

"""
Evaluates the implicit constraint function at a given point/solution ( $\vec{x}$ )
@ In, Input, object, RAVEN container
@ Out, g(inputs x1,x2,..,output or dependent variable), float, implicit constraint evaluation function
the way the constraint is designed is that
the constraint function has to be  $\geq 0$ ,
so if:
1)  $f(x,y) \geq 0$  then  $g = f$ 
2)  $f(x,y) \geq a$  then  $g = f - a$ 
3)  $f(x,y) \leq b$  then  $g = b - f$ 
4)  $f(x,y) = c$  then  $g = 1e-6 - \text{abs}((f(x,y) - c))$  (equality constraint)
"""

Let's assume that the constraint is:
 $x_3 + x_4 < 8$ 
then g the constraint evaluation function (which has to be  $> 0$ ) is taken to be:
 $g = 8 - (x_3 + x_4)$ 
in this case if  $g(\vec{x}) < 0$  then this x violates the constraint and vice versa
@ In, Input, object, RAVEN container
@ out, g, float, explicit constraint 1 evaluation function
"""

def expConstr1(Input):
    """
    @ In, Input, object, RAVEN container
    @ out, g, float, explicit constraint 3 evaluation function
    """
    g = 10 - Input.x3 - Input.x4
    return g

def impConstr1(Input):
    """
    The implicit constraint involves variables from the output space, for example the objective variable or
    a dependent variable that is not in the optimization search space
    @ In, Input, object, RAVEN container
    @ out, g, float, implicit constraint 3 evaluation function
    """
    g = 200 - Input.obj1
    return g

def impConstr2(Input):
    """
    The implicit constraint involves variables from the output space, for example the objective variable or
    a dependent variable that is not in the optimization search space
    @ In, Input, object, RAVEN container
    @ out, g, float, implicit constraint 3 evaluation function
    """
    g = Input.obj2 - 16
    return g

```

Figure 6. Implicit and explicit constraint modeling in PRLO platform.

Figure 7 and Figure 8 show the multisum test results. In Figure 7, there are three pairs of scatter plots (i.e., Objective 1 with respect to Objective 2, Objective 2 with respect to Objective 3, and Objective 1 with respect to Objective 3) over selected iterations. At Iteration 1, most chromosomes are violating either one of the given constraints. As the number of iterations increases, only chromosomes that comply with given constraints survived. Note that chromosomes in the Objective 1–Objective 2 plane showed a clear tendency toward approaching the top-left corner since the NSGA-II algorithm is trying to minimize Objective 1 at the same time that it maximizes Objective 2. Similarly, chromosomes in the Objective 2–

Objective 3 plane and chromosomes in the Objective 1–Objective 3 plane showed a clear tendency toward approaching the bottom right corner and bottom left corner, respectively.

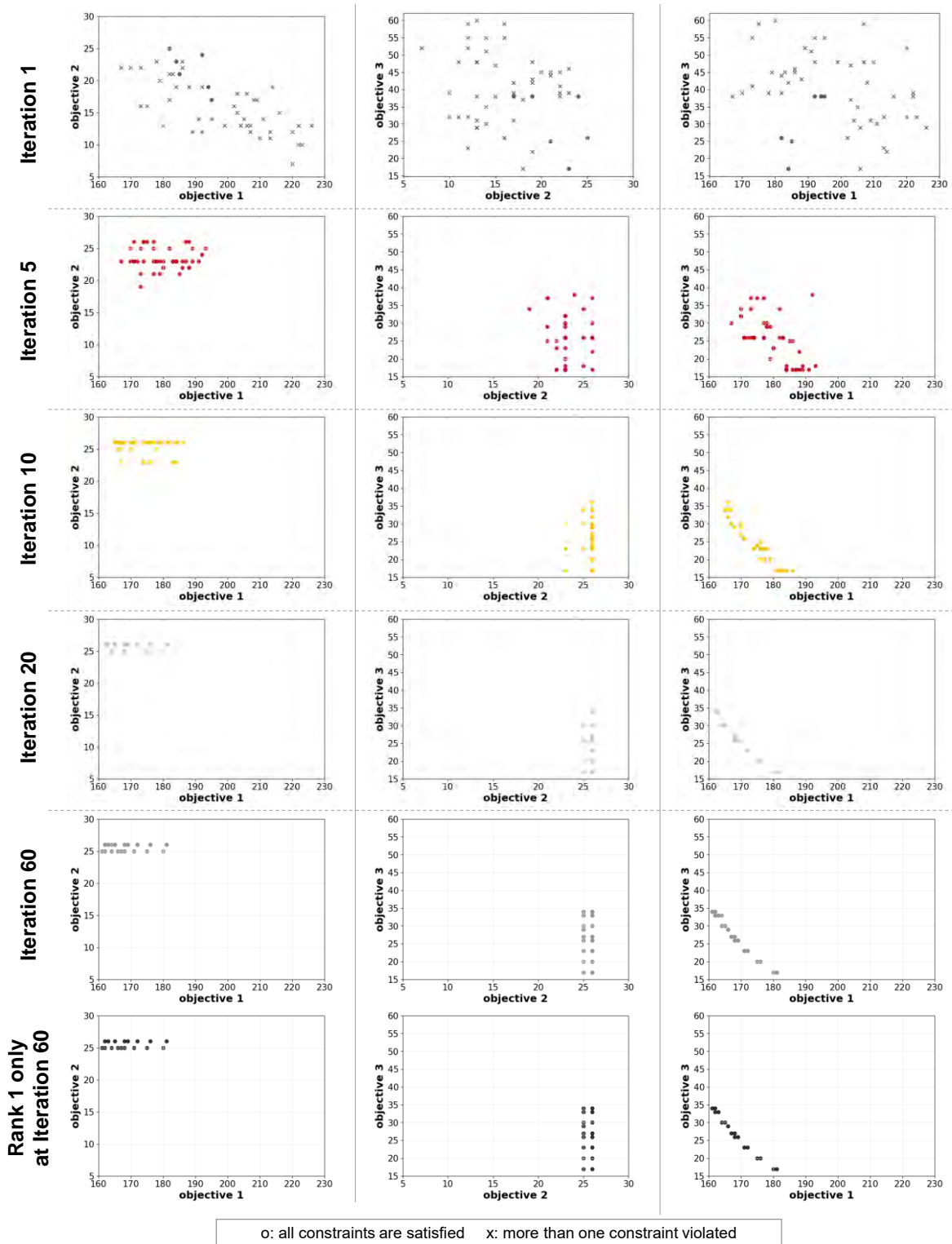
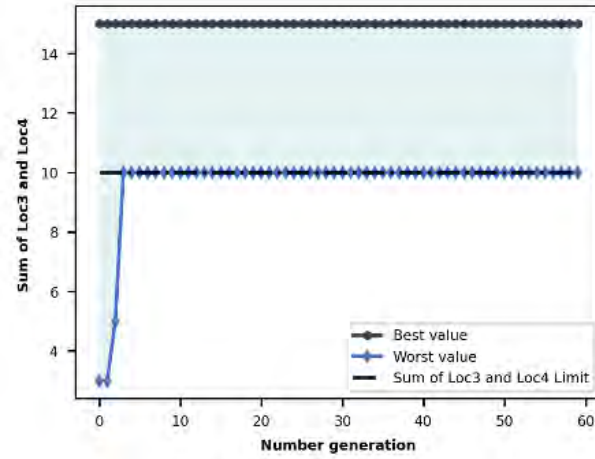
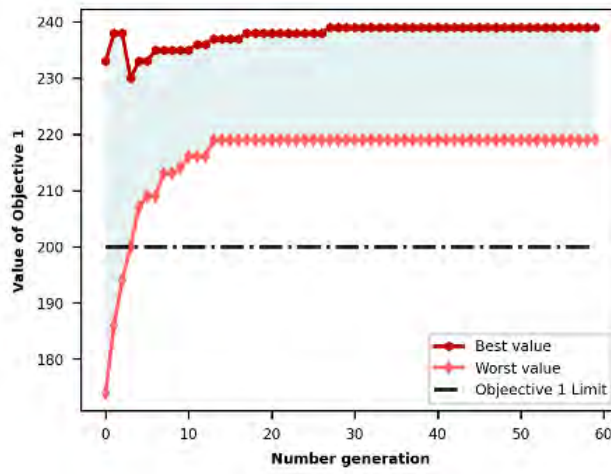


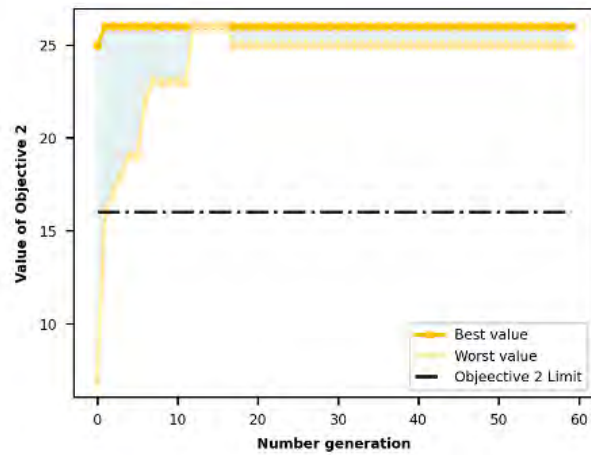
Figure 7. Multisum test results in PRLO platform.



(a) Explicit constrain: $x_3 + x_4 < 10$



(b) Implicit constraint: Objective 1 < 200



(c) Implicit constraint: Objective 2 > 16

Figure 8. Constraints change over iteration.

Table 1 shows parameters used for implementing NSGA-II in the PRLO platform for multisum problem solving. A two points crossover and random mutator gave enough perturbation to the offspring populations. The probabilities of crossover and mutation were set to 0.7 from the maximum value of 1.0. This led to an increased level of diversity in the offspring populations.

Table 1. NSGA-II parameters for the multisum problem in PRLO platform.

Population size	50	Generation	60
Crossover type	Two points crossover	Mutation type	Random mutator
Probability of crossover	0.7	Probability of mutation	0.7

Chromosomes at the Rank 1 frontier at Iteration 60 are the ones that cannot be dominated by other chromosomes, and they are listed in Table 2. In Figure 8, the constraints change over iteration are shown. One can find that the ranges of constraint values of population at an early iteration are wide, which implies that some of chromosomes are violating some constraints. As the number of iterations increases, every chromosome in population complies with constraint limits.

Table 2. A list of optimal solutions of the multisum problem after 60 iterations.

Chromosomes								Objectives			Constraints		
X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	Obj1	Obj2	Obj3	10- (x ₃ +x ₄)	200- Obj1	Obj2- 16
8	9	4	2	7	6	5	3	176	26	20	4	24	10
8	9	3	2	7	6	5	4	181	26	17	5	19	10
9	8	6	4	7	5	3	2	161	25	34	0	39	9
9	8	6	3	7	5	4	2	164	25	30	1	36	9
8	9	6	4	7	5	3	2	162	26	34	0	38	10
9	8	5	2	7	6	4	3	171	25	23	3	29	9
9	8	3	2	7	6	5	4	180	25	17	5	20	9
8	9	5	2	7	6	4	3	172	26	23	3	28	10
9	8	4	2	7	6	5	3	175	25	20	4	25	9
9	8	7	2	6	5	4	3	166	25	29	1	34	9
8	9	6	3	7	5	4	2	165	26	30	1	35	10
8	9	7	3	6	5	4	2	163	26	33	0	37	10
9	8	6	2	7	5	4	3	168	25	26	2	32	9
8	9	6	2	7	5	4	3	169	26	26	2	31	10
8	9	5	3	7	6	4	2	168	26	27	2	32	10
9	8	5	3	7	6	4	2	167	25	27	2	33	9
9	8	7	3	6	5	4	2	162	25	33	0	38	9

3. DEMONSTRATION OF CORE DESIGN WITH MULTIOBJECTIVE OPTIMIZATION

3.1 Problem Statement

A demonstration was performed for the constrained multiobjective optimization of a 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:

- $F_Q < 2.1$
- $F\Delta H < 1.48$
- 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 with five different fuel types is given in Table 3. The fuel cost is calculated by counting how many different fuel assemblies were used for the given loading pattern then multiplying them by their corresponding unitary value as given in Table 3.

Table 3. Fuel assembly 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 (\$)²	0	2.69	3.25	3.25	4.04	4.04

Figure 9 shows the fuel assembly design with and without Gd burnable poison loading. An initial 1/4 core loading pattern with five different fuel types is illustrated in Figure 10.

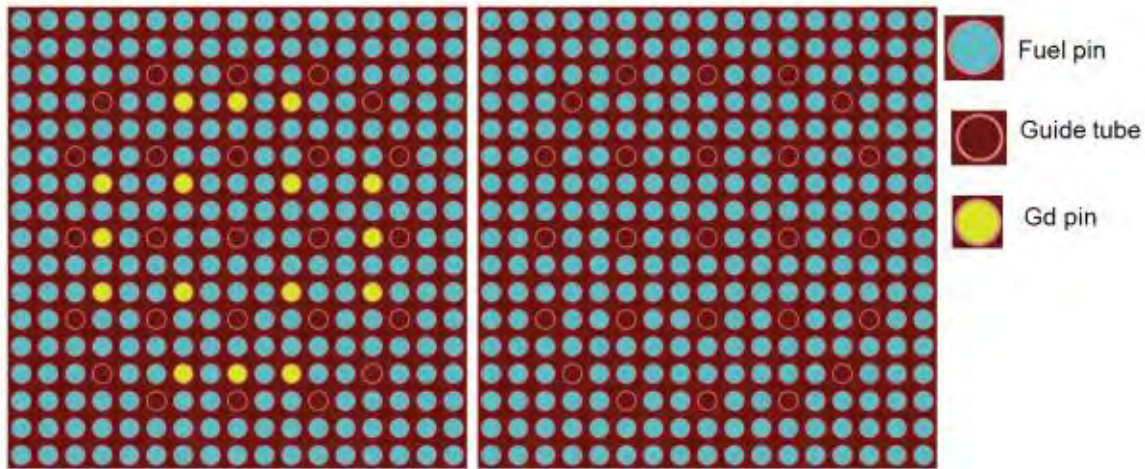


Figure 9. Pin map of a fuel assembly with (left) and without (right) Gd burnable poison loading.

² <https://world-nuclear.org/information-library/economic-aspects/economics-of-nuclear-power.aspx>

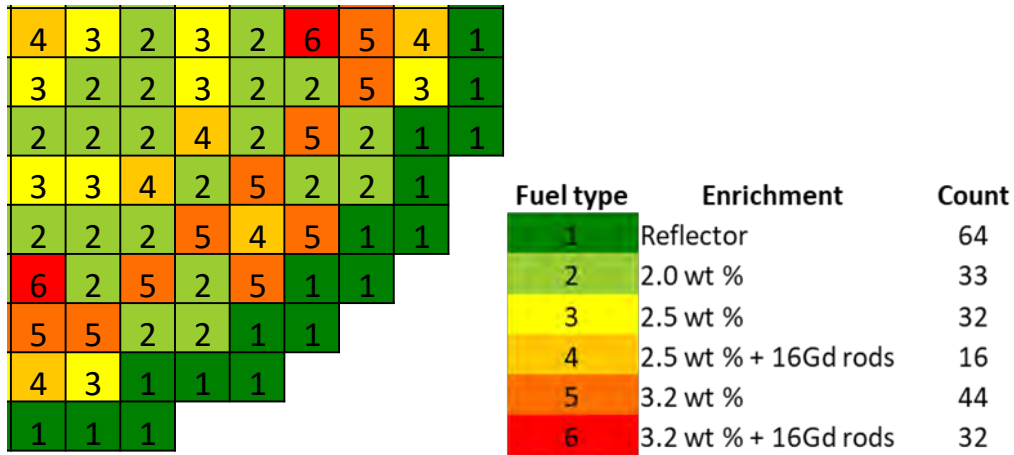


Figure 10. A quarter symmetry of a core loading pattern.

3.2 Methodology

To reduce the complexity of the problem, a one-eighth symmetry for the loading pattern was used and encoded in a chromosome, as shown in Figure 11.

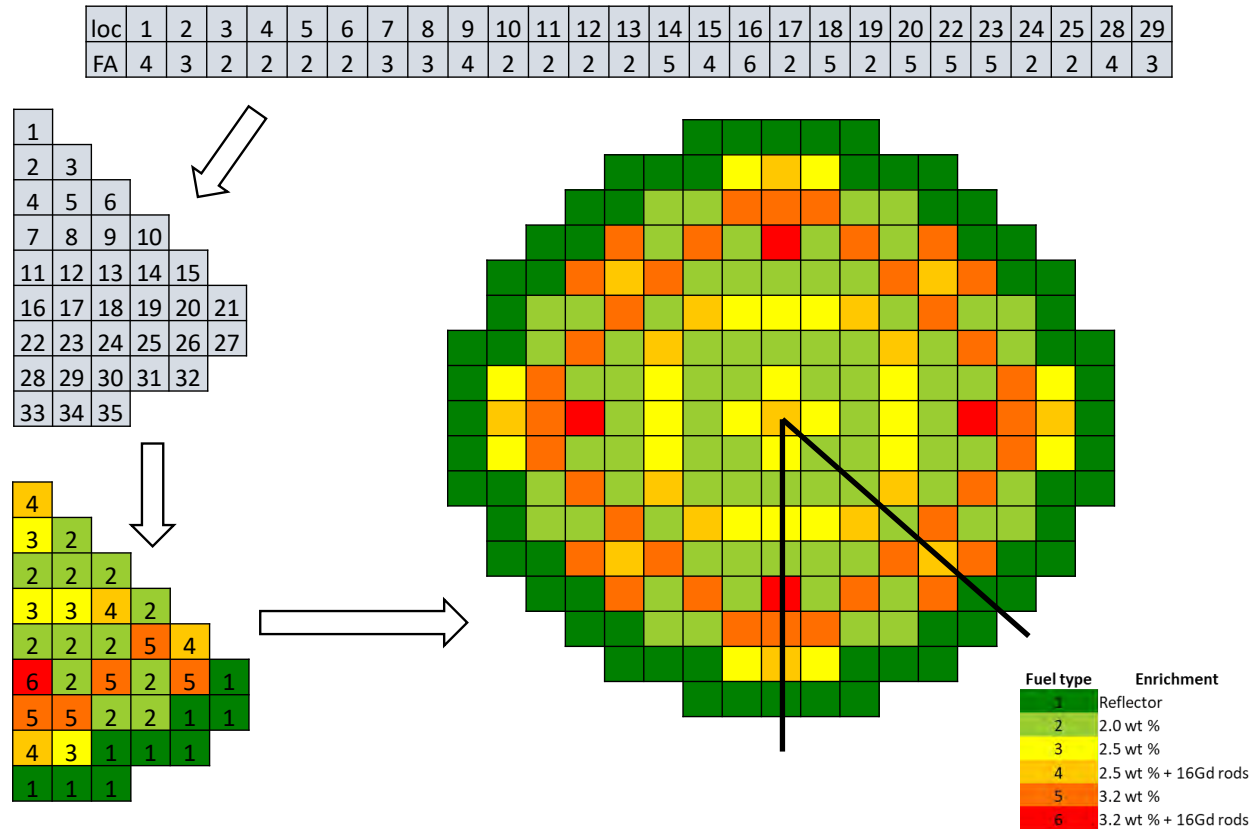


Figure 11. Chromosome encoding a one-eighth symmetry of PWR loading pattern.

A total of 35 positions were set with six different types of assemblies, including the reflectors. Positions 21, 26:27, and 30:35 are fixed with reflectors leaving the rest of 26 chromosomes for the five

different types of fuel assemblies. The loading patterns in a chromosome was encoded to SIMULATE-3K from the PRLO platform input. The cycle length values were taken from the SIMULATE-3K output.

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 the next equations, ArtObjOne is the augmented cycle length objectives and ArtObjTwo is the augmented fuel cost objectives.

$$\text{ArtObjOne} = -\text{Cycle Length} + \sum_{j=1}^{m=3} w_j \max(0, \text{penalty}_j)$$

$$\text{ArtObjTwo} = \text{Fuel Cost} + \sum_{j=1}^{m=3} u_j \max(0, \text{penalty}_j)$$

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.

3.2.1 SIMULATE-3K-RAVEN Coupling Interface

A Python-3 interface was developed for RAVEN to generate inputs of SIMULATE-3K and extract SIMULATE-3K outputs for NSGA-II optimizer execution in PRLO platform. This interface is designed for the 17×17 PWR core design with 157 FAs.

The coupling interface has following files:

- `SIMULATEInterface.py`: Connects and interacts with RAVEN main module
- `SIMULATEData.py`: Collects and extracts data from SIMULATE output file, including:
 - Time-dependent multiplication factor k_{eff}
 - Time-dependent F_q
 - Time-dependent $F\Delta h$
 - Time-dependent critical boron concentration
 - Cycle length determined by the critical boron concentration
 - Time-dependent relative pin power distribution
 - Number of FAs of each type
 - Augmented objectives for cycle length and fuel cost.

To account for the degree of violation of constraints, “augmented” variables ArtObjOne and ArtObjTwo are also extracted by the interface.

- `SpecificParser.py`: Generates SIMULATE input from information provided by RAVEN-GA module.

The SIMULATE-3K/RAVEN interface needs the following three input files:

- `Sim-3param.xml`—Fuel assembly definition, cross section information and other specifications for SIMULATE
- `Cms.pwr-all.lib`—Cross section information
- `Input.inp`—Template used by RAVEN to print input files for SIMULATE.

3.2.2 Workflow for 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.

User input file sim3-param.xml was used to specify the characteristics of the PWR, such as the FAs, cross-section libraries, and dimensions. RAVEN checks for any input error, then samples FA location mapping, generating a SIMULATE-3K input file by using SimulateInterface.py. This file was also used to execute SIMULATE-3K. By using SimulateData.py, RAVEN parses specified variables from the SIMULATE-3K output file and was used as part of the NSGA-II optimizer. This whole process will be repeated until the given number of iterations is completed. Figure 12 shows the flowchart of the SIMULATE-3K and RAVEN coupling interface.

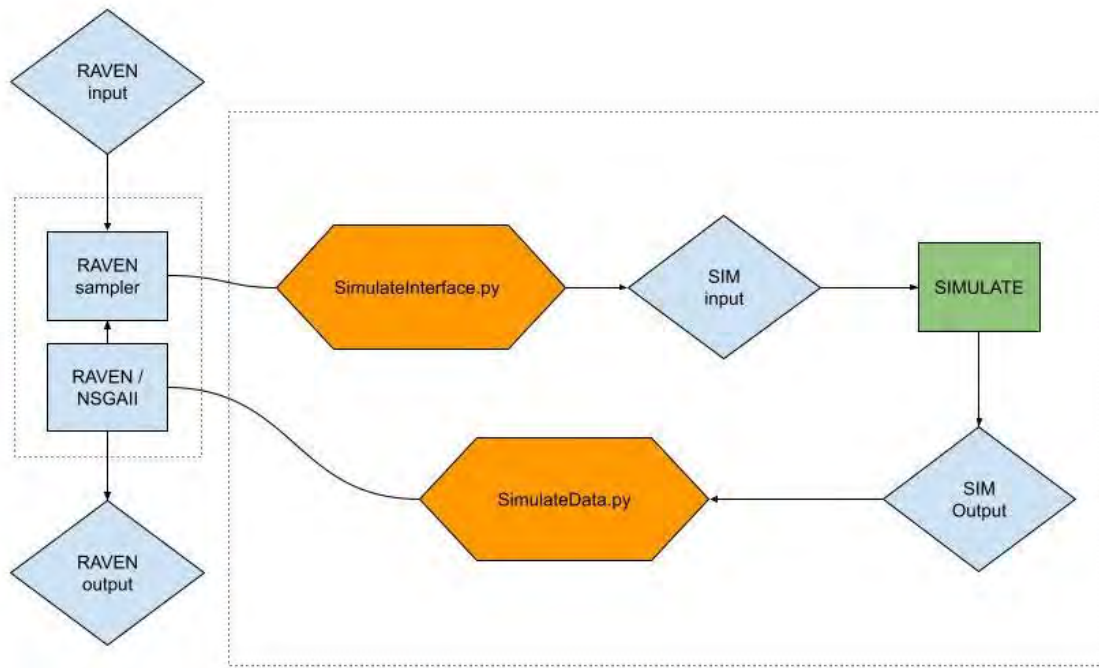


Figure 12. Flowchart of SIMULATE-3K/RAVEN coupling interface.

With the use of the augmented objectives, many simulations could be performed to determine the penalty coefficients inclusive. 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.

Figure 13 shows the PRLO platform input file (e.g., sim3-param.xml) to sample SIMULATE-3K input files.

```

1 <Sim3-input-gen>
2   <pins> 17 </pins>
3   <core_width> 15 </core_width>
4   <load_point> 0.000 </load_point>
5   <depletion> 20 </depletion>
6   <axial_nodes> 25 </axial_nodes>
7   <batch> 0 </batch>
8   <pressure> 2250.0 </pressure>
9   <boron> 900.0 </boron>
10  <power> 100.0 </power>
11  <flow> 100.0 </flow>
12  <inlet_temperature> 550.0 </inlet_temperature>
13  <map_size> quarter </map_size>
14  <symmetry> octant </symmetry>
15  <restart_file> cycle1.res </restart_file>
16  <cs_lib> cms.pwr-all.lib </cs_lib>
17  <number_assemblies> 157 </number_assemblies>
18  <working-dir> SampleSpecificSim3 </working-dir>
19  <reflector> TRUE </reflector>
20  <FA-list>
21    <FA name=' FA1'   FAid = '0'   type = '2' />
22    <FA name=' FA2'   FAid = '1'   type = '3' />
23    <FA name=' FA3'   FAid = '2'   type = '5' />
24    <FA name=' FA4'   FAid = '3'   type = '4' />
25    <FA name=' FA5'   FAid = '4'   type = '6' />
26    <FA name=' REF'   FAid = '5'   type = '1' />
27  </FA-list>
28 </Sim3-input-gen>

```

Figure 13. User input file used by PRLO platform to sample SIMULATE-3K input files.

3.3 Optimization Results

3.3.1 NSGA-II Optimization Results

NSGA-II cycle length and fuel cost optimization were performed for population size of 50 with 50 generations. Figure 14 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.

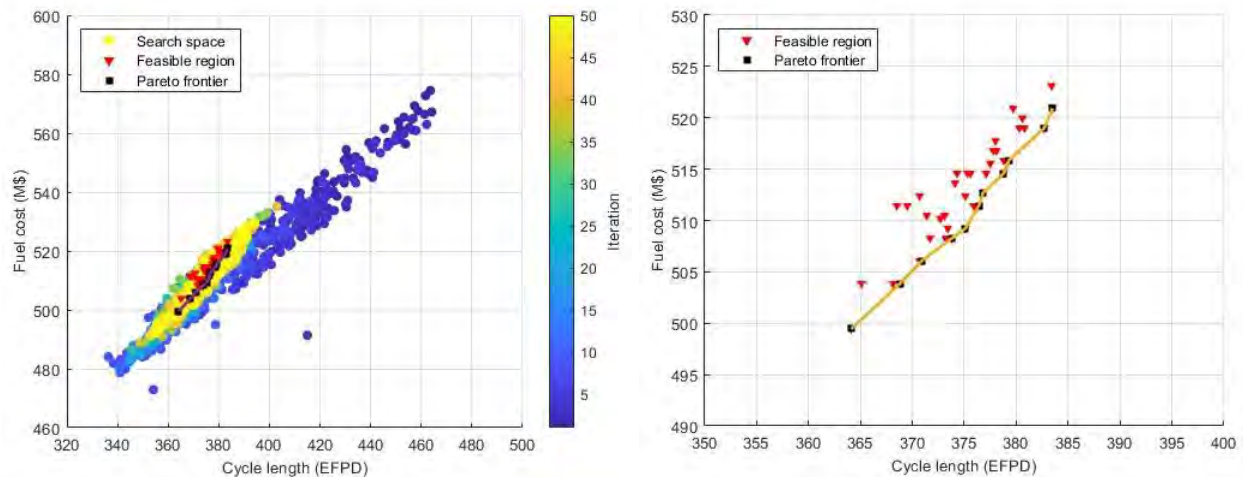


Figure 14. Search space and feasible region (left) and Pareto frontier (right) for NSGA-II optimization (population size = 50, generations = 50).

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 solution. The frontier values are identified as #1 from the lowest to #11 at the highest position. Figure 15 shows the lowest (#1) position of the Pareto frontier. The fuel cost was \$499,450,000 with a fuel cycle length of 364.10 EFPD.

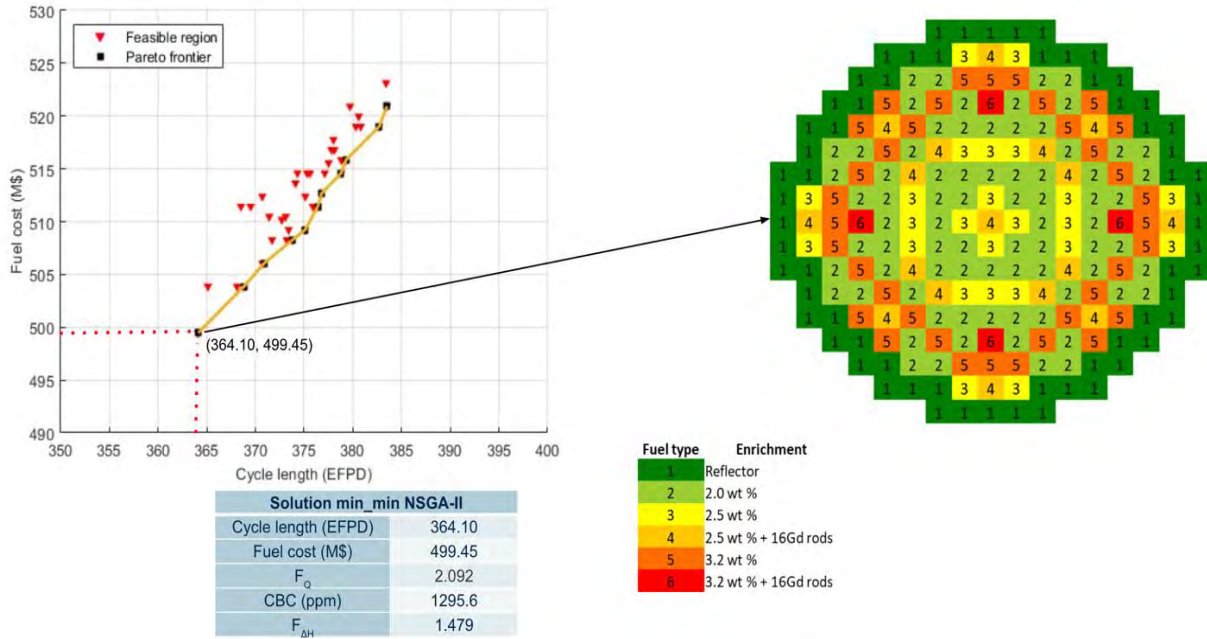


Figure 15. Sample Pareto optimal solution for NSGA-II run with population size of 50 and 50 generations (Position #1).

Figure 16 and Figure 17 show the middle (#4) and highest (#11) positions of the Pareto frontier. The fuel cost was \$508,280,000 and \$520,920,000 and the fuel cycle length of 373.80 EFPD and 383.50 EFPD, respectively.

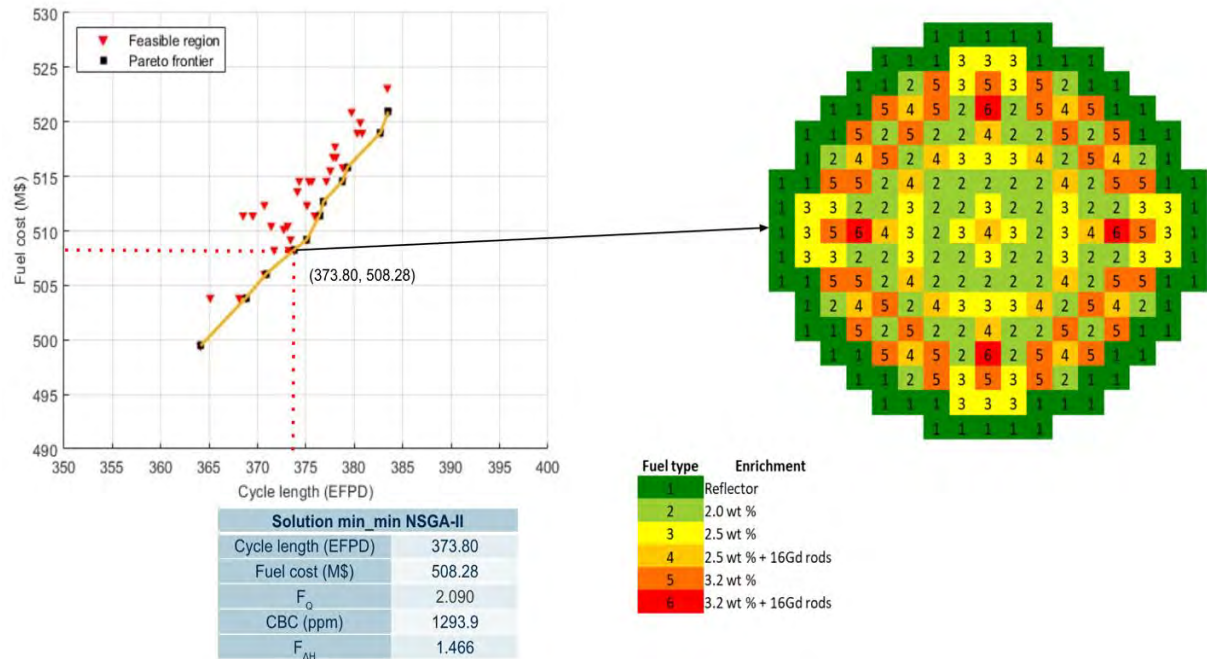


Figure 16. Sample Pareto optimal solution for NSGA-II run with population size of 50 and 50 generations (Position #4).

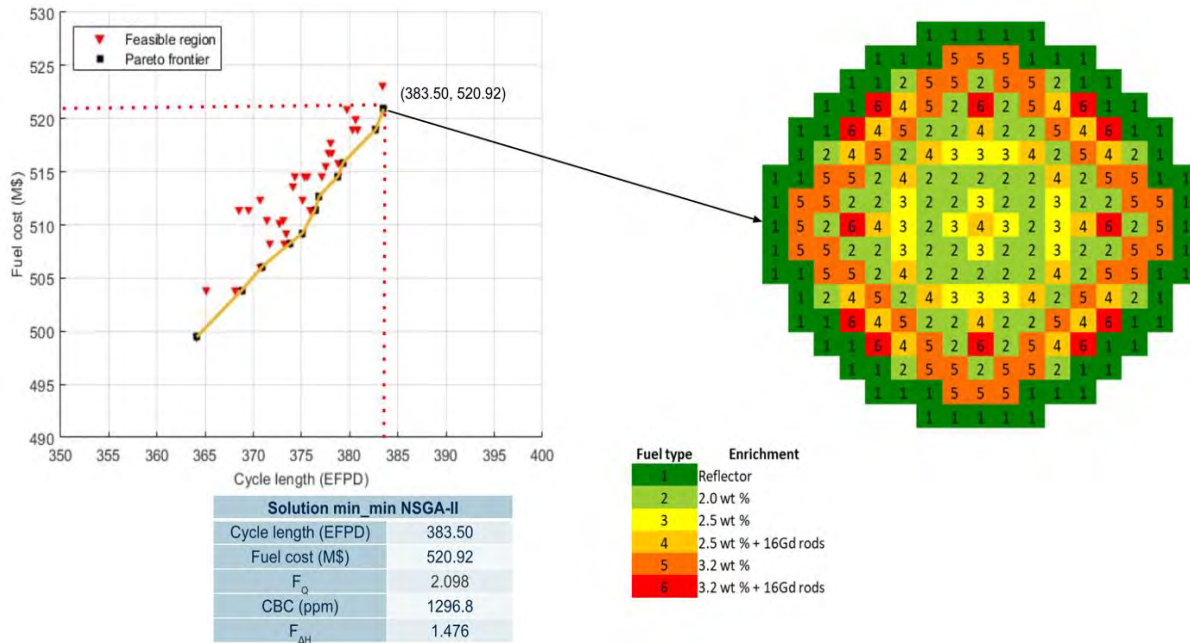


Figure 17. Sample Pareto optimal solution for NSGA-II run with population size of 50 and 50 generations (Position #11).

The core loading pattern generated by the NSGA-II optimizer is 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]

3.3.2 Sensitivity Study on the Population Size

A sensitivity study was conducted for 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 18 and Table 4. 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 better optimized core reloading patterns. However, the computational burden would be increased as the population increased. The simulation time was about 6.5 hours for 50 population size and about 16.5 hours for 100 population size. 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.

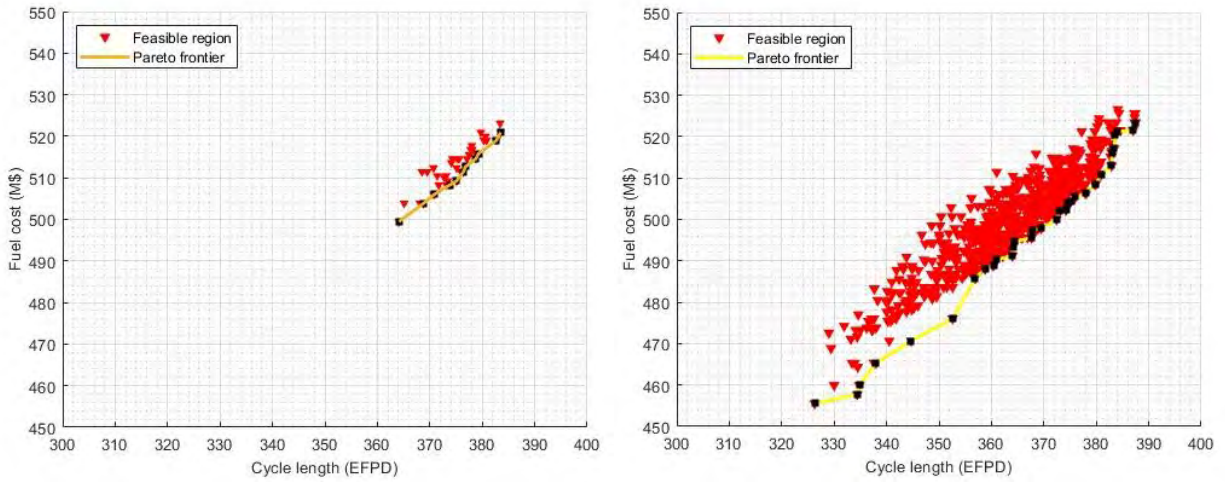


Figure 18. Feasible region for population size of 50 (left) and 100 (right).

Table 4. Performance comparison with respect to population size in NSGA-II.

Population size	50	100
Total solutions generated (\propto 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%
Simulation time	6hr 31m 56s	16hr 39m 46s

3.3.3 Comparison Between Single-Objective GA and Multiobjective NSGA-II

A comparison study was performed between single-objective GA and multiobjective NSGA-II methods to verify the feasibility of the optimizer performance and efficiency. To handle multiple objectives with the single-objective GA method, an artificial objective, \bar{C} , was introduced, which is the function of the fuel cost, C , and cycle length, L :

$$\bar{C} = C/L$$

Minimizing this artificial objective could be achieved through the minimum fuel cost and the maximum fuel cycle length.

Figure 19 shows the feasible region and Pareto frontier from the single-objective GA optimizer. The number of population size and generations were set to 50, which is same as the NSGA-II optimizer test in Section 3.3.1. A total of 2,500 chromosomes encoding a loading pattern were generated in serial, of which 1,154 are unique. The feasible region, which contains chromosomes that comply with all the constraints, is composed of 14 chromosomes, of which six are part of the Pareto frontier, numbered from #1 to #6.

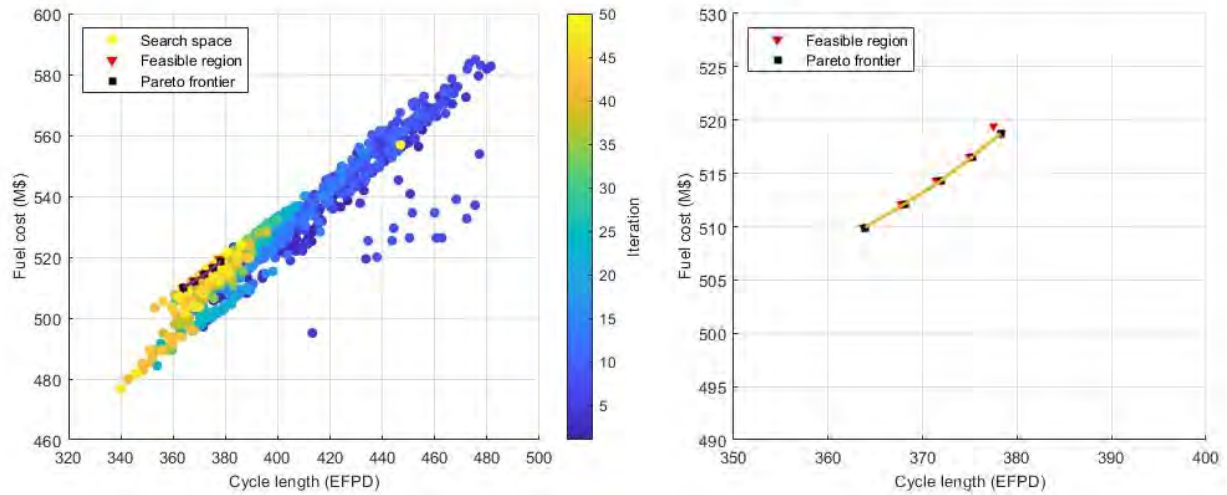


Figure 19. Search space and feasible region (left) and Pareto frontier (right) from single-objective GA optimization (population size = 50, generations = 50).

The feasible region and Pareto frontier was found on the left side of the search space, which gives a shorter fuel cycle length compared to the multiobjective NSGA-II optimizer results shown in Figure 14, which shows the Pareto frontier is located at the right side of the search space. Also, the NSGA-II method produced more optimal solutions at the Pareto frontier (i.e., 11). This result shows the multiobjective NSGA-II optimization method generated more acceptable result for the fuel reload problem.

Figure 20, Figure 21, and Figure 22 show the lowest (#1), upper-middle (#5), and highest (#6) positions of the Pareto frontier from single-objective GA optimization solutions. The fuel cost was \$509,930,000, \$516,550,000, and \$518,760,000 and the fuel cycle length was 364.0 EFPD, 375.2 EFPD, and 378.3 EFPD, respectively. The fuel cost was found to be generally higher than the similar fuel cycle length case result of NSGA-II solutions.

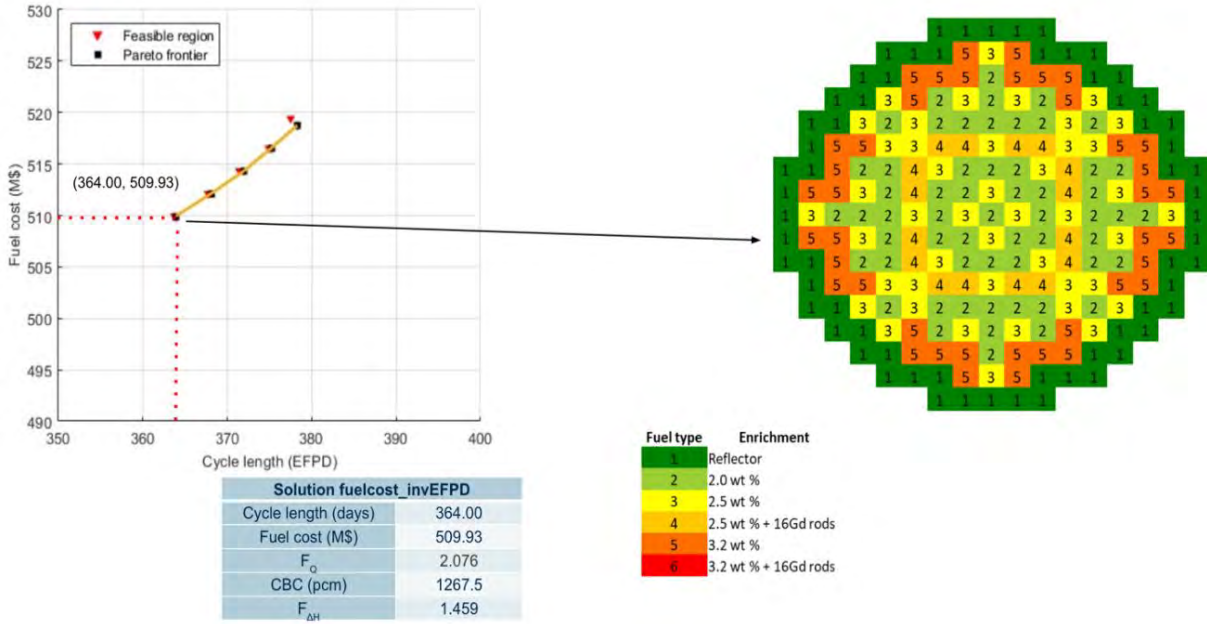


Figure 20. Sample Pareto optimal solution for GA run with population size of 50 and 50 generations (Position #1).

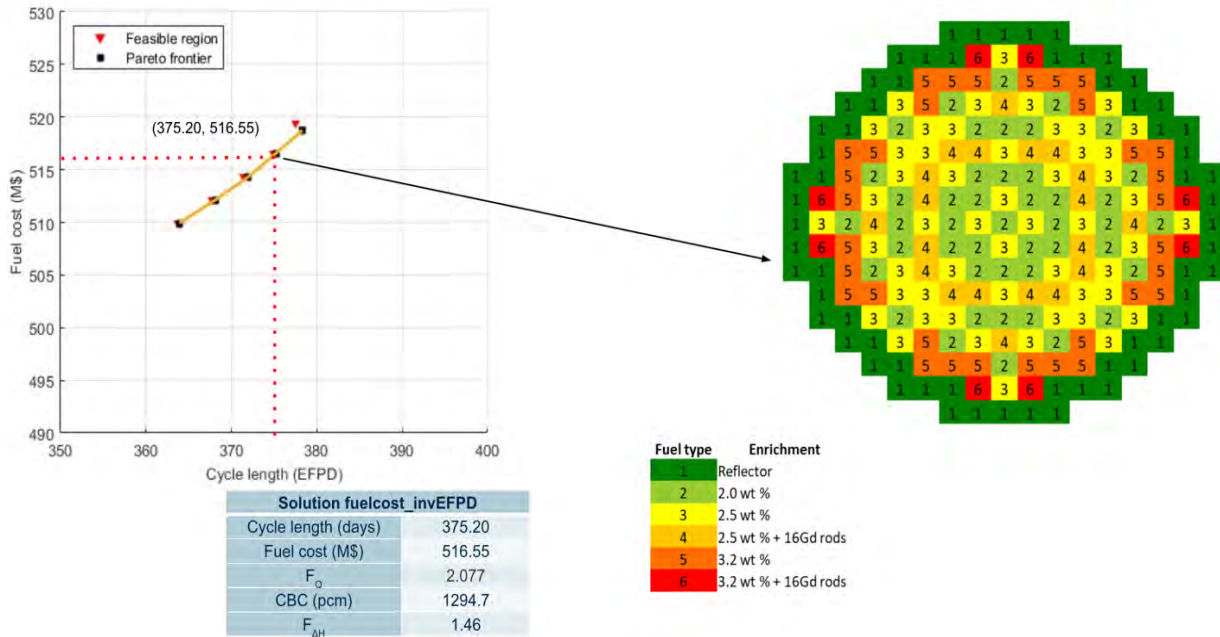


Figure 21. Sample Pareto optimal solution for GA run with population size of 50 and 50 generations (Position #5).

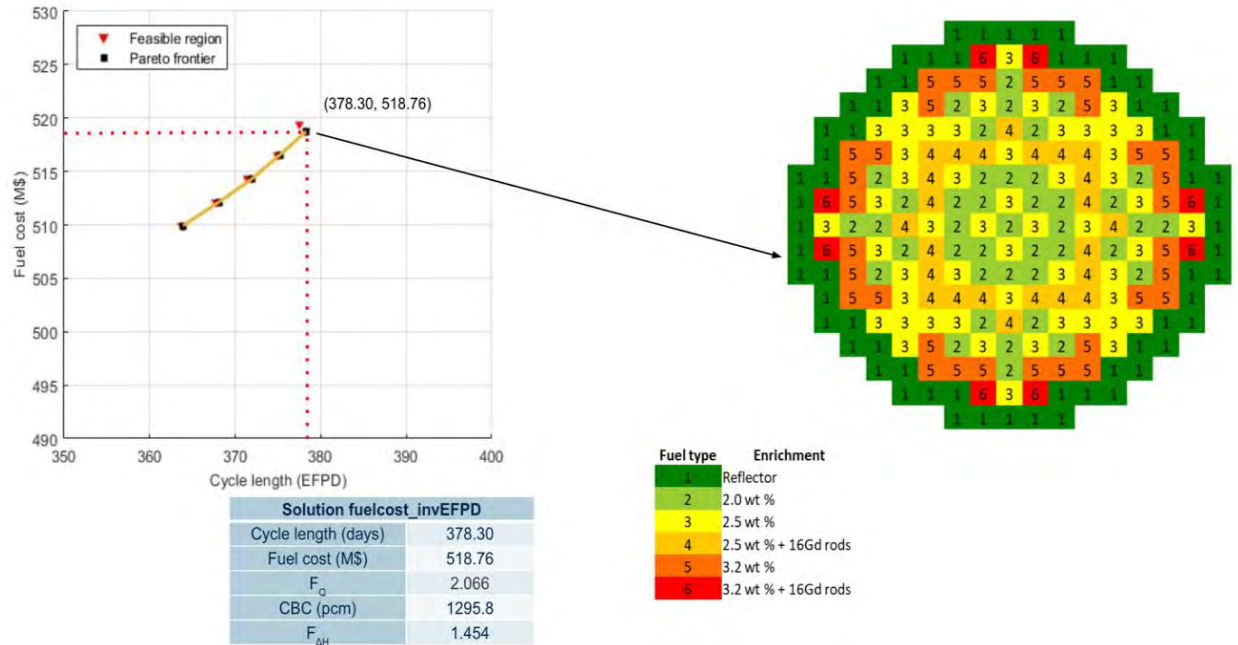


Figure 22. Sample Pareto optimal solution for GA run with population size of 50 and 50 generations (Position #6).

Figure 23 shows a comparison of core loading patterns between the single-objective GA and multiobjective NSGA-II optimizers with a similar length of fuel cycle: 364 EFPD and 364.1 EFPD, respectively. The fuel cost was about \$10,000,000 less in the NSGA-II optimizer result: \$509,930,000 and \$499,450,000, respectively. The fuel loading pattern of the NSGA-II result shows a better low-leakage core design, which is the preferable option to actual fuel reloading applications.

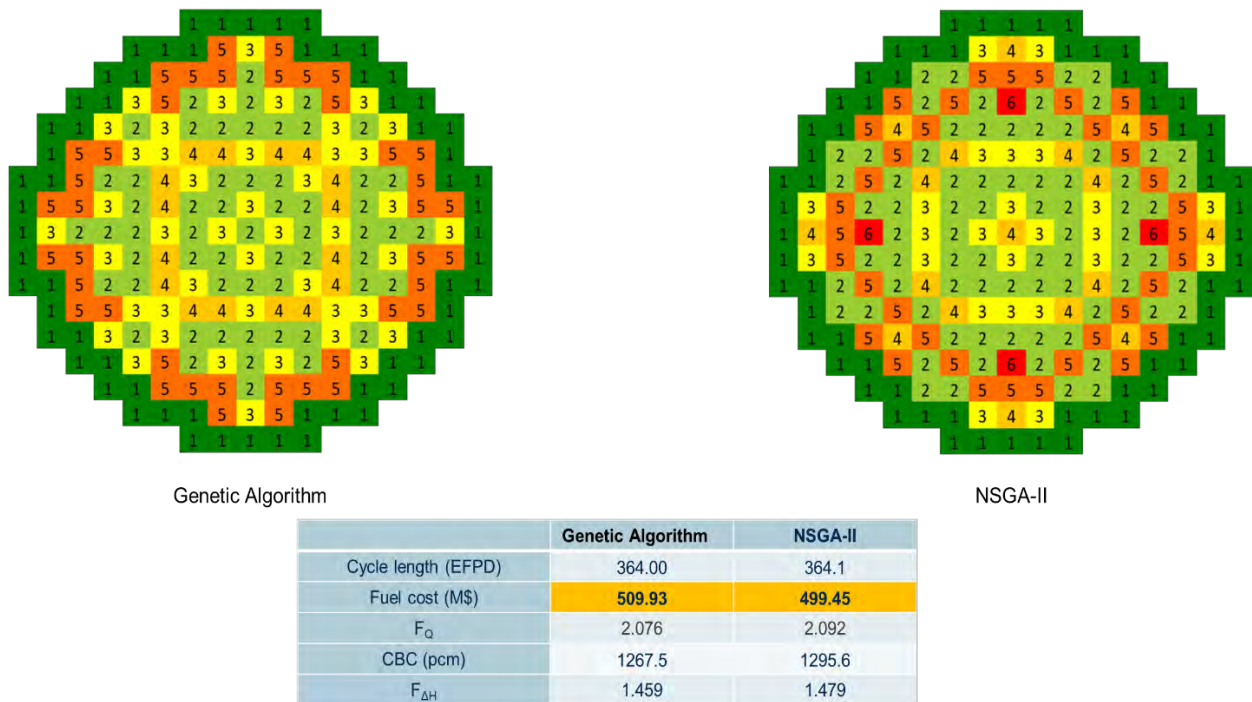


Figure 23. Comparison of GA and NSGA-II optimal loading patterns.

4. OPTIMIZATION WITH MULTIPHYSICS FEEDBACK

4.1 Multiphysics Coupling

The main goal of the PRLO platform is to design an optimized reactor core which considers system safety analysis and fuel performance, thus multiphysics methodology. The computational tools were already coupled through RAVEN. [1] A weak coupling method was used since the tools has different disciplines or multiphysics involved which means input and output data is controlled by RAVEN as shown in Figure 2. In other words, the weak coupling method is suitable for multiphysics problem solving which has different physical time orders between each tool. For instance, core design needs the core physics data throughout the entire cycle duration (generally 18 or 24 months for the core depletion calculation). In the meantime, system safety analysis by RELAP5-3D only needs a certain point of time when accident has occurred. As an example of weak coupling in the PRLO platform, once potential optimized core designs were proposed, RAVEN transfers necessary data to the system code (e.g., RELAP5-3D) for DBA analysis defined in the Nuclear Regulatory Commission's NUREG-0800. [8] Then again RAVEN transfers necessary data to the fuel safety analysis code (e.g., TRANSURANUS) to confirm its performance and detail of failure mechanisms.

Typically, both normal operation, operational transients and anticipated operational occurrences events are considered for the core design. [8] Power excursions may lead to xenon transients, which will translate into a spectrum of power shapes and peaking factors that must be considered in the safety analysis downstream.

4.1.1 Core-Physics Data

This section provides the breakdown of inputs that should come from the core design code. Pin power histories (for every modeled pin) are needed. The breakdown of the TRANSURANUS input file contents is shown in Figure 24 for a sample time step.

- Time-dependent, axially-dependent linear heat rate (W/mm)³
- Time-dependent, axially-dependent fast flux (neutrons/cm²-s)
- Time-dependent steady-state system conditions⁴
 - Coolant pressure
 - Coolant temperature
 - Coolant mass flow rate
 - Thermal-hydraulic diameter of fuel rod channel.

³ A consistent set of axial power shapes between RELAP5-3D and the core physics tools need to be established.

⁴ These data could be entered manually into the RAVEN optimizer input file to be propagated into the TRANSURANUS input.

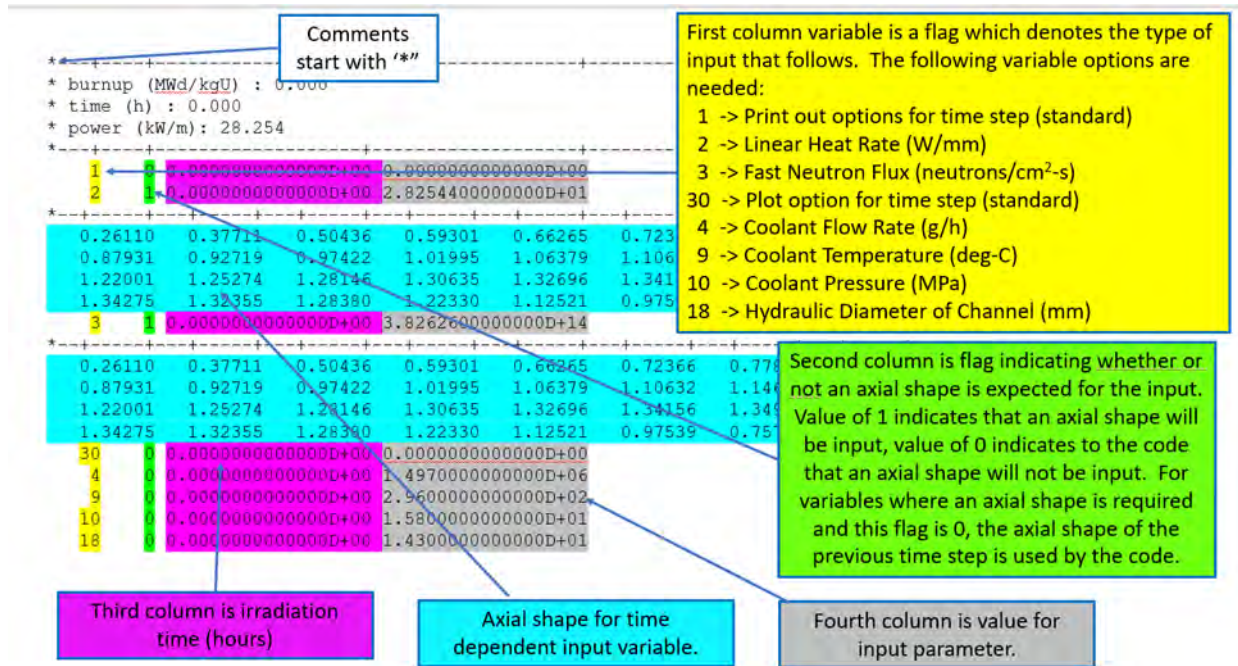


Figure 24. Sample interface file for TRANSURANUS from core design.

4.1.2 Thermal-Hydraulics Data

The RELAP5-3D input needs axial power shape of the reactor core. In general, the power shape is calculated based on the reactor design data from the literature (e.g., FSAR) or from core design computational tools. For PRLO platform a new approach was used for the RELAP5-3D input. This approach obtains the power shape information from the core design tool and distills it into peaking factors, which are then used to define axial power shapes to feed to the RELAP5-3D simulation. The peaking factors may also become constraints managed in the outer optimization algorithm. The two key peaking factors are defined in this section.

4.1.2.1 F_Q Limit

The peaking factor, F_Q , represents the ratio of the maximum local power density to the core average power density. This peaking factor is intended to establish a limit on the linear heat rate experienced in the core, and it is important for many considerations, such as peak cladding temperatures, fuel centerline temperatures, departure of nucleate boiling rate, etc. For safety analysis applications, F_Q is defined as:

$$F_Q = F_Q^N \cdot F_U^N \cdot F_Q^E$$

$$F_Q = \max(P_z(z) \cdot F_{xy}(z)) \cdot F_U^N \cdot F_Q^E$$

where:

$$P_z(z) = \text{Core average axial power distribution}$$

$$F_{xy}(z) = \text{Ratio of the local peak power density to core-average density at } z$$

$$F_U^N = 1.05 \text{ (added 5\% for conservatism)}$$

$$F_Q^E = 1.03 \text{ (engineering heat flux channel factor)}$$

Using this definition, a constraint on F_Q may be derived from a safety limit value of 2.5. Considering conservatisms and engineering heat flux channel factor, the design limit value is 2.31. [9]

4.1.2.2 $F_{\Delta H}$ Limit

The enthalpy rise hot channel factor, $F_{\Delta H}$, represents the ratio of the integral of linear power along the fuel rod with the highest integrated power to the average rod power and is defined as:

$$F_{\Delta H}^N = F_{\Delta H \text{ measured}}^N \cdot F_U^N$$

where:

$$\begin{aligned} F_{\Delta H}^N &= \text{The value used in safety analyses} \\ F_{\Delta H \text{ measured}}^N &= \text{The value obtained from core design} \\ F_U^N &= 1.04 \text{ (typical measurement error)} \end{aligned}$$

Using this definition, a constraint on $F_{\Delta H}$ could be derived from a typical safety limit value of 1.7. The design limit is 1.635 by dividing with typical measurement error. [9]

Taking the peaking factors from the core design outputs, RELAP5-3D simulations could be performed for a certain point of time of operation when accident occurred. Once the system response is calculated with RELAP5-3D, system thermal-hydraulic conditions are extracted from RELAP5-3D results and fed to TRANSURANUS. This was done using a method wherein the time-dependent TRANSURANUS cards are assembled as:

- The core design outputs are used to create the time-dependent TRANSURANUS cards at the intervals specified in the core design outputs up until the time in the cycle at which the transient is determined to occur.
- A new time-dependent TRANSURANUS card is added and interpolated to the exact time in cycle of the transient.
- Time-dependent TRANSURANUS cards are added for the RELAP5-3D time steps. The flux and power values from the last core design-based card are scaled by the power percentage from RELAP5-3D, and the thermal-hydraulic inputs are taken directly from RELAP5-3D.

Detail of above each step is not publicly available.

4.1.3 TRANSURANUS Output

The primary output of a TRANSURANUS calculation provide information about the integrity of the fuel and cladding information such as fuel centerline temperature, fuel rod swelling, rod internal pressure, temperature, cladding oxidation and cladding strain. The information is also important to predict cladding deformation, fuel failure, relocation and dispersal (i.e., FFRD), cladding oxidation and cladding failure by rupture.

During a LOCA, the cladding will further oxidize due to an exothermic reaction between the cladding and steam. The run-away reaction is controlled by regulatory limits imposed by the maximum peak clad temperature (2,200°F of 10 CFR 50.46 acceptance criteria). The transient local oxidation can also be calculated by TRANSURANUS. Note that, in case of cladding burst, double-sided oxidation may occur, and that calculation would also need to account for clad thinning due to the computed strain.

Figures of merit were selected from the TRANSURANUS output to be processed in RAVEN. These include the fuel cladding deformation and assembly rod (i.e., fuel pin) pressure. These figures of merit allow for the RAVEN optimization to factor in safety concerns of ballooning and rod limitations for the refuel design.

4.1.4 RAVEN-TRANSURANUS Coupling Interface

The RAVEN-TRANSURANUS coupling interface was built to allow the PRLO platform to control TRANSURANUS for the following simulations and executions:

- Standalone TRANSURANUS execution with keyword substitution of sampled variables or upstream variables and prebuilt time-dependent cards
- TRANSURANUS execution with keyword substitution downstream of PARCS execution with time-dependent cards built based on the PARCS output
- TRANSURANUS execution with keyword substitution downstream of RELAP5-3D execution with some prebuilt time-dependent cards and time-dependent cards built based on the RELAP5-3D output
- TRANSURANUS execution with keyword substitution downstream of PARCS execution and RELAP5-3D execution with time-dependent cards built based on a combination of PARCS and RELAP5-3D.

In all cases, the output is collected using the TuPlot module, which is a module program included with TRANSURANUS. The interface uses keyword substitution, which requires a RAVEN variable name, a keyword string to search in the deck, and the format of the line that contains the keyword. Note that there are no limitations on the keyword format, the analyst should be careful to use a unique string. This is because TRANSURANUS has some limited input spaces that do not allow every convention to fit as shown in Figure 25.

```

<variable name='rab'>
  <keyword>$rab</keyword>
  <format>8F10.5</format>
</variable>
<variable name='korngnr'>
  <keyword>$korngnr</keyword>
  <format>8F10.5</format>
</variable>
* HEADER: FTELL:      3483 % VALUES: rib, rab, rih, rah,raubl (lschni),rauhl (lschni) % FORMAT: 8F10.5
0.0      $rab      4.72000  5.43000  0.00160  0.0003
* HEADER: FTELL:      3564 % VALUES: aeax6,ikorn % FORMAT: 16I5
0      0
* HEADER: FTELL:      3645 % VALUES: korngnr (1,1,1) % FORMAT: 8F10.5
$korngnr

```

Figure 25. Example of keywords in TRANSURANUS input file.

The different use cases for the TRANSURANUS interface and how the TRANSURANUS time-dependent cards are constructed are shown in Figure 26. The number of pins simulated can be input according to these guidelines. In the cases of standalone TRANSURANUS or RELAP5-3D+TRANSURANUS coupled simulation, there is a single TRANSURANUS simulation (i.e., one fuel pin) executed. In the cases of PARCS+TRANSURANUS or PARCS+RELAP5-3D+TRANSURANUS coupled simulation, the user may specify any number of pin locations in the core (based on assembly row, assembly column, pin row, and pin column). A simulation needs to be performed for each location specified.

For output collection, the user must supply a TuPlot input file. The outputs are saved back to RAVEN with the following conventions of standalone TRANSURANUS or RELAP5-3D+TRANSURANUS coupled simulation, DNX_CNy_ASz, where:

- DNX stands for diagram number x. This is a naming convention based on the TRANSURANUS Plot, which a TRANSURANUS analyst will be familiar with. Diagram numbers inform the user of information such as “what quantity is being printed?”
- CNY stands for curve number y. This is a naming convention based on TRANSURANUS Plot, which a TRANSURANUS analyst will be familiar with. Curve numbers inform the user of information such as “what is the specific variable being printed?”
- ASZ stands for axial slice z. Axial slices are the TRANSURANUS naming convention for the axial levels in the simulation.

For PARCS+TRANSURANUS or PARCS+RELAP5-3D+TRANSURANUS coupled simulations, ARa_ACb_PRC_PCD_DNx_CNy_Asz are defined as:

- DNX_CNy_Asz is described above
- ARa stands for assembly row a
- ACb stands for assembly column b
- PRc stands for pin row c
- PCd stands for pin column d

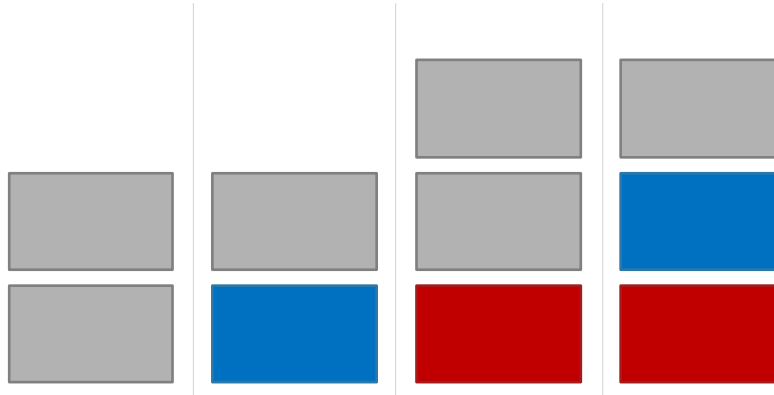


Figure 26. Different methods of building time-dependent TRANSURANUS cards.

The interface coding is briefly described below. The RAVEN interface consists of one Python class titled TRANSURANUS. The class is comprised of multiple methods:

- Method `__init__`: This is a required constructor method, which leverages the base class constructor.
- Method `_readMoreXML`: This allows the portion of the xml input with specialized class and initialize attributes based on inputs.
- Method `parseParcs`: Obtains values from the PARCS outputs for input to TRANSURANUS, including time steps, linear heat rates, fast neutron fluxes, flow rate, and coolant temperatures.
- Method `parseRelap`: Obtains values from the PARCS outputs for input to TRANSURANUS, including thermal-hydraulic time steps, power fractions, flow rate, coolant temperatures, and coolant pressures.
- Method `subValues`: Creates the perturbed TRANSURANUS input file excluding the time-dependent cards. Note that this method includes logic to handle various fixed format statements.
- Method `addTdParcsCards`: Adds time-dependent PARCS cards to the rod-specific TRANSURANUS file, as described earlier in this section.
- Method `addTdRelapCards`: Adds time-dependent RELAP5-3D cards to the rod-specific TRANSURANUS file, as described earlier in this section. Note that this includes logic to interpolate the existing PARCS cards to the exact time in the cycle that the RELAP5-3D simulation is determined to start at.
- Method `readOutputs`: Reads the TuPlot outputs to create the output data to send back to RAVEN.

- Method `createNewInput`: Calls other methods and does miscellaneous processing to create an input file in a subdirectory for each rod being simulated.
- Method `generateCommand`: Creates and executes a script to change into a rod-specific directory, execute TRANSURANUS and TuPlot, then exits the directory. Note that this method was used because RAVEN does not have the capability to embed RAVEN-running-RAVEN more than two processes deep. The existing setup already used RAVEN-running-RAVEN, and thus the separating running of TRANSURANUS for each rod was handled this way.
- Method `finalizeCodeOutput`: Collects the outputs from each individual rod run and saves into one common CSV file with labeled assembly and rod coordinates.

4.2 Demonstration with Large-Break Loss-of-Coolant Accident Case

4.2.1 Problem description

A demonstration was performed with a LBLOCA to verify the correct workflow between PARCS, RELAP5-3D, and TRANSURANUS with a single objective (i.e., maximize fuel cycle length, effective full power day, EFPD) PRLO platform. The Zion NPP RELAP5-3D LBLOCA model was used. [8]

The reactor core has 157 fuel assemblies of 17×17 fuel configurations as shown in Figure 9. Each fuel assembly has five different fuel types as given in Table 3. The objective is to find the maximum fuel cycle length (i.e., EFPD). A single constraint was set as follows:

- Peak cladding temperature (PCT) < 1,800 F

Other reactor design parameters (e.g., $F_Q < 2.1$, $F_{\Delta H} < 1.48$ and Boron concentration < 1300 ppm) were not set as problem constraints because this test is designed for the verification purpose of the multiphysics PRLO platform.

The setting of PRLO platform is shown in Table 5. Description of the option values are given in the reference. [8] Monte Carlo sampling method was used to select initial 5 random chromosomes (i.e., 5 different core configurations). Each population has 5 chromosomes, and 10 populations were given (i.e., total of 50 different core configurations) for each generation. Optimization ends when 10th generation was obtained.

Table 5. Setting of PRLO platform for multiphysics optimization demonstration.

Parameters	Value
Single objective	Maximum EFPD
Single constraint	PCT < 1,800 F
Number of chromosomes	5
Number of populations	10
Number of generations	10
Parent selection method	Roulette
Crossover operation method	One point
Mutation operation method	Random

Figure 27 shows data flow in demonstration of multiphysics single objective optimization. Each step of the optimization process is given as follows:

- PRLO platform generates initial shuffled core and PARCS produces power distribution data and EFPD.
- PRLO platform generates RELAP5-3D input with peak power location from PARCS power distribution data and run to check PCT during LBLOCA is under 1,800F.
- If above PCT condition is met, PRLO platform generates TRANSURANUS input from RELAP5-3D (e.g., PCT) and PARCS (e.g., power distribution) to simulate cladding deformation.
- Simulation ends when 10th generation is calculated.

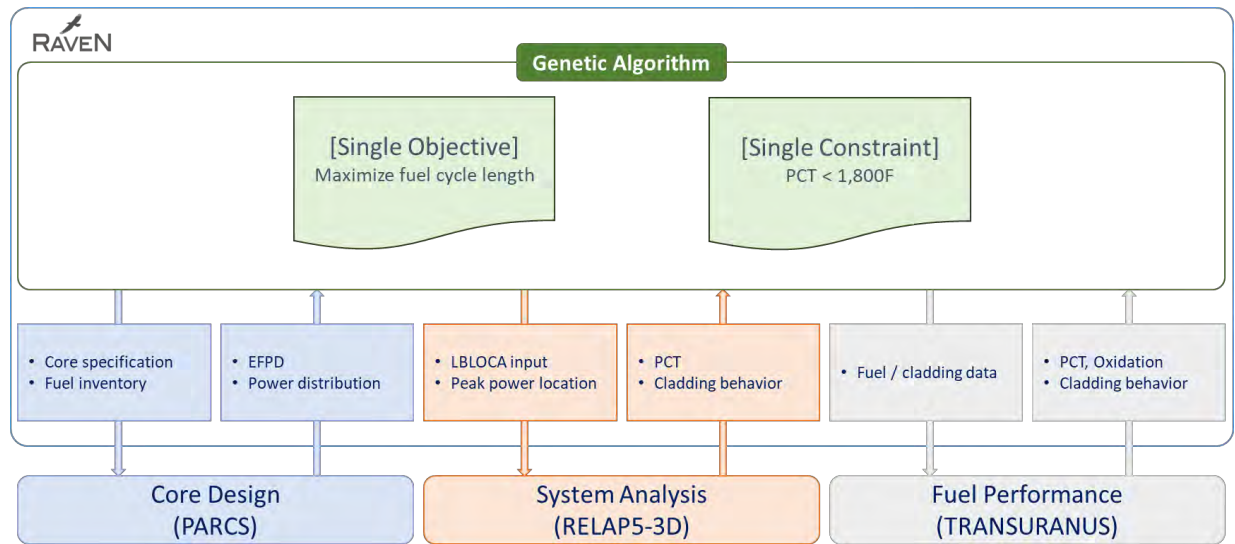


Figure 27. Schematic diagram of data flow in multiphysics single objective optimization demonstration.

4.2.2 Results and discussion

Figure 28 shows evolution of EFPD during multiphysics optimization demonstration which shows value of 542.8585 at the end of iteration. During the simulation, PCT was converged into given constraint (e.g., PCT < 1,800 F) after 22nd iteration. Other design criteria such as Fq, FΔH and boron concentration, as shown in Figure 29, were found higher than design limitations since these criteria were not set as the constraints in this demonstration case. Figure 30 shows the core configuration, PCT and cladding deformation value at 1st, 22nd and the final iteration which is optimized core during this multiphysics optimization demonstration. From the PCT graph, the maximum value was found lower than constraint. At the final iteration, it is clear to observe that PCT was found lower than given constraint.

It is noted that the results shown in Figure 28, Figure 29 and Figure 30 does not correctly represent realistic behavior of PWR since this demonstration aims to verify multiphysics PRLO platform is correctly working. Actual reactor problem will be performed in upcoming years.

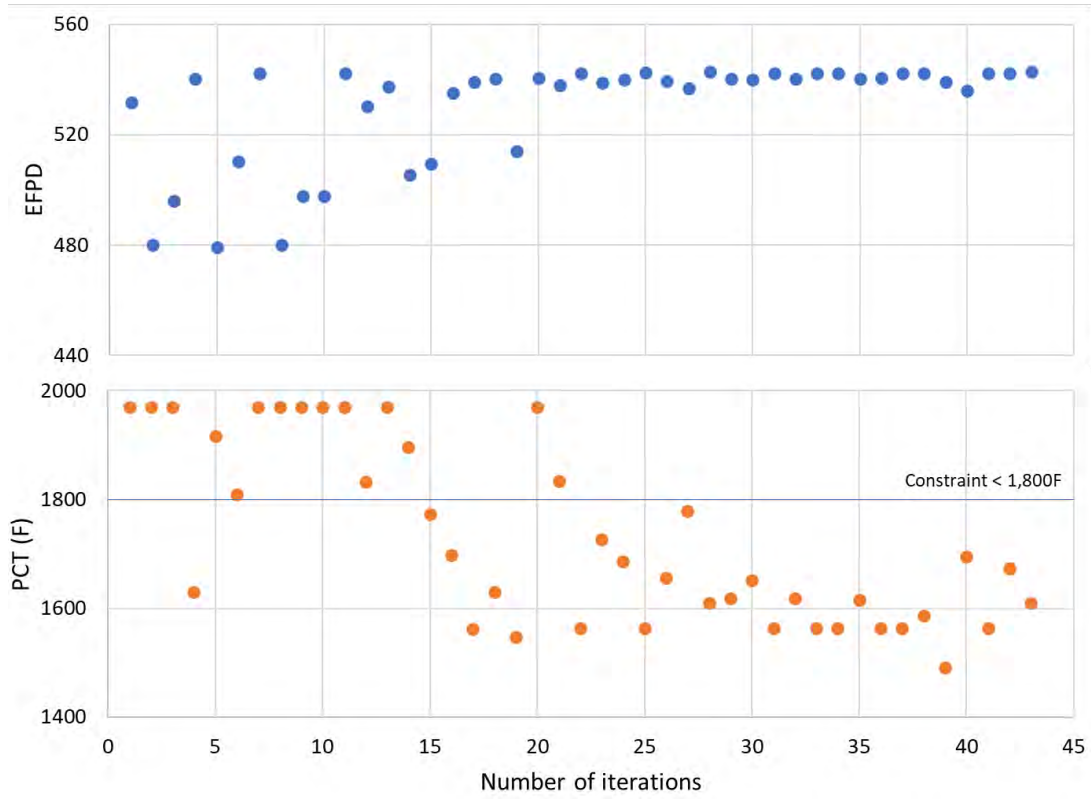


Figure 28. Evolution of EFPD and PCT during multiphysics optimization demonstration.

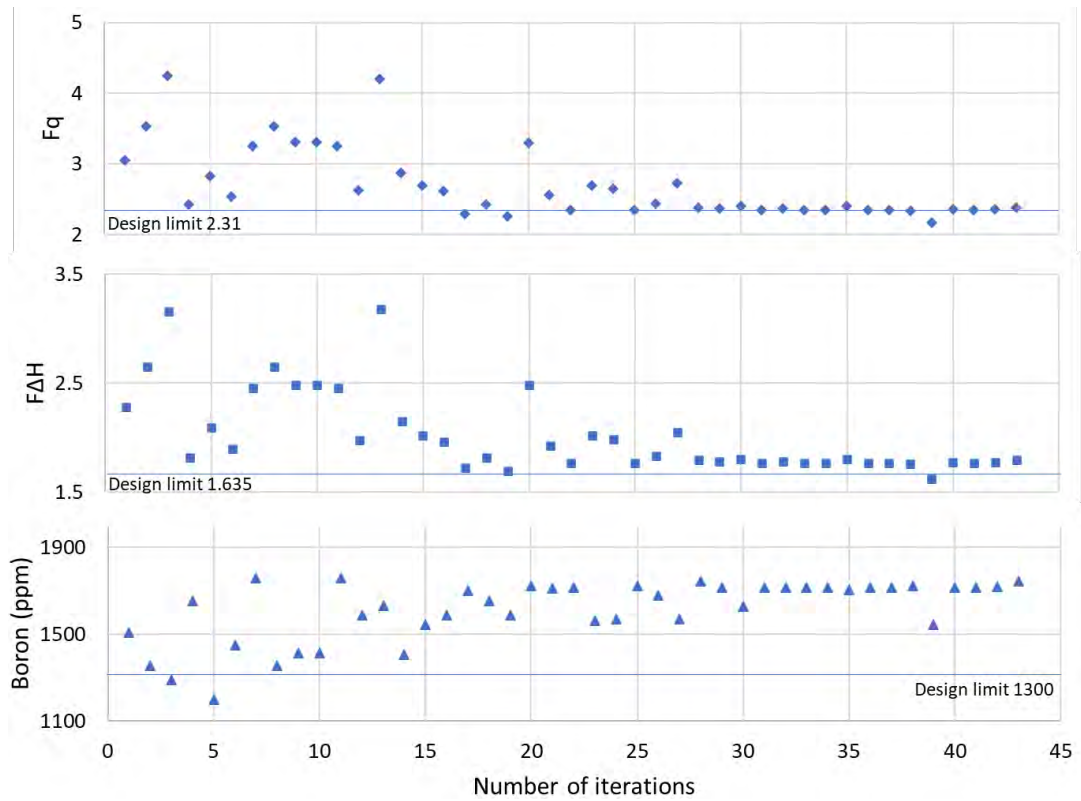


Figure 29. Evolution of core design criteria in multiphysics optimization demonstration.

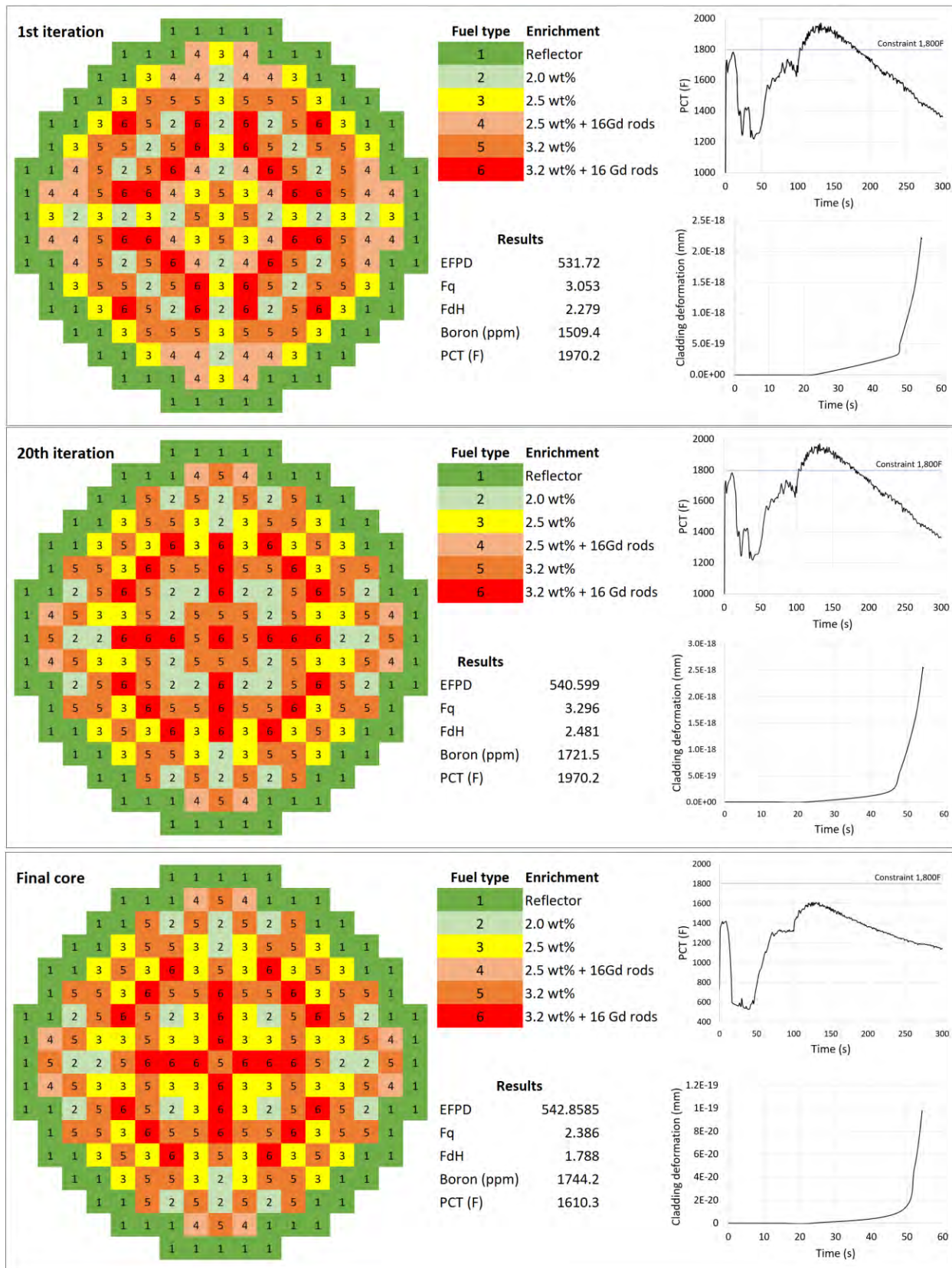


Figure 30. Evolution of core design during multiphysics optimization demonstration (1st 20th and final, from top to bottom).

5. SUMMARY AND FUTURUE WORKS

The NSGA-II optimizer was improved to handle a large number of constraints by using an augmented objectives methodology and was tested within the PRLO platform. The PRLO platform is now fully capable of handling realistic problems of artificial-intelligence-based reactor core designing.

The demonstration was performed with constrained multiobjective optimization of a 17×17 PWR core loading pattern to minimize fuel cost and maximize fuel cycle length. Five different fuel enrichments were initially given, including Gd burnable poisons. The SIMULATE-3K to RAVEN coupling interface was built and tested for NSGA-II optimizer. 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. Compared with the single-objective GA optimizer with an artificial objective function, the multiobjective NSGA-II optimizer shows better results in both the generation of optimized solutions and realistic reloading pattern.

The multiphysics (i.e., core design, thermal-hydraulics and fuel performance) PRLO platform was demonstrated for the verification purpose. RAVEN was modified and tested to control input and output data flow from RELAP5-3D and TRANSURANUS in PRLO platform. The demonstration was only performed with a single-objective GA optimizer with single constraint to verify the feedback effect is correctly applying. Further research will be continued for full-scale PWR DBA scenarios with a constrained multiobjective NSGA-II PRLO platform.

Future activities planned for FY-2024 are:

- Conduct a full-scale demonstration of a PWR core design to minimize the volume of new fuel, including core and system safety analysis considerations.
- Develop additional capabilities of the multiobjective optimization methodology by applying adaptive algorithms, termination criteria, and constraints; these improvements are necessary to integrate thermal-hydraulics and fuel performance inputs into the core design process.

6. REFERENCES

- [1] Choi, Y-J., M. Abdo, G. Palamone, S. Heagy, C. Frepoli, K. Ogujiuba, N. Rollins, N., et al. 2022. “Development and Demonstration of a Risk-Informed Approach to the Regulatory Required Fuel Reload Safety Analysis.” INL/RPT-22-68628, Idaho National Laboratory. <https://doi.org/10.2172/1885790>.
- [2] Alfonsi, A., C. Rabiti, D. Mandelli, J. J. Cogliati, C. Wang, D. P. Maljovec, P. W. Talbot, et al. 2021. “RAVEN Theory Manual.” INL/EXT-16-38178, Rev. 4, Idaho National Laboratory. <https://doi.org/10.2172/1260312>.
- [3] Andersen, B. D., “A Machine-Learning Based Approach to Minimize Crud Induced Effects in Pressurized Water Reactors.” Dissertation, North Carolina State University, 2021. <https://repository.lib.ncsu.edu/handle/1840.20/38656>.
- [4] Kim, J., M. Abdo, C. Wang, and Y.-J. Choi. 2023. “Development of Genetic Algorithm Based Multi-Objective Plant Reload Optimization Platform.” INL/RPT-23-71667, Idaho National Laboratory.
- [5] Choi, Y.-J., M. Abdo, C. Wang, J. Valeri, C. Frepoli, K. Nguyen, and J. Hou. 2022. “Development of Plant Reload Optimization Platform Capabilities for Core Design and Fuel Performance Analysis.” INL/RPT-22-70382, Idaho National Laboratory.
- [6] Park, G. T. 1996. “Multiobjective pressurized water reactor reload core design by nondominated genetic algorithm search,” *Nuclear Science and Engineering*, vol. 124, no. 1, pp. 178-187, 1996. <https://doi.org/10.13182/NSE96-A24233>.
- [7] Kim, Y-J., Downer T., and Sesonske, A. 1987. “Optimization of Coe Reload Design for Low-Leakage Fuel Management in Pressurized Water Reactors”, *Nuclear Science and Engineering*, vol. 96, pp. 85-101. <https://doi.org/10.13182/NSE87-A16369>.
- [8] Choi, Y-J., M. Abdo, D. Mandelli, A. Epiney, J. Valeri, C. Gosdin, C. Frepoli, et al. 2021. “Demonstration of the Plant Fuel Reload Process Optimization for an Operating PWR.” INL/EXT-21-64549, Idaho National Laboratory, 2021.
- [9] U.S. NRC, “Westinghouse Technology Systems Manual, Section 2.2, Power Distribution Limits”, <https://www.nrc.gov/docs/ML1122/ML11223A208.pdf>.

Page intentionally left blank

Page intentionally left blank

Appendix A

RAVEN Input for Multisum Optimization Problem

```

<?xml version="1.0" ?>
<Simulation verbosity="debug" profile="jobs">
  <TestInfo>
    <name>\raven\tests\framework\Optimizers\NSGAI
I\discrete\constrained</name>
    <author>Junyung Kim, Mohammad Abdo </author>
    <created>2022-12-21</created>
    <classesTested/>
    <description>NSGA-II min-min test
</description>
  </TestInfo>

  <RunInfo>
    <WorkingDir>Multi_MinwoReplacement_Complex_23
0807_pop50_60</WorkingDir>
    <Sequence>optimize,print</Sequence>
    <batchSize>4</batchSize>
  </RunInfo>

  <Steps>
    <MultiRun name="optimize" re-seeding="2286">
      <Input class="DataObjects"
type="PointSet">placeholder</Input>
      <Model class="Models"
type="ExternalModel">myLocalSum</Model>
      <Optimizer class="Optimizers"
type="GeneticAlgorithm">GAopt</Optimizer>
      <SolutionExport class="DataObjects"
type="PointSet">opt_export</SolutionExport>
      <Output class="DataObjects"
type="PointSet">optOut</Output>
      <Output class="OutStreams"
type="Print">opt_export</Output>
      </MultiRun>
      <IOStep name="print">
        <Input class="DataObjects"
type="PointSet">opt_export</Input>
        <Input class="DataObjects"
type="PointSet">optOut</Input>
        <Output class="OutStreams"
type="Print">opt_export</Output>
        <Output class="OutStreams"
type="Print">optOut</Output>
      </IOStep>
    </Steps>

    <Models>
      <ExternalModel
ModuleToLoad="..\myLocalSum_multi.py"
name="myLocalSum" subType="">
        <variables>x1,x2,x3,x4,x5,x6,x7,x8,obj1,,obj
j2,obj3</variables>
      </ExternalModel>
    </Models>

    <Functions>
      <External file="..\myConstraints.py"
name="expConstr3">
        <variables>x1,x2,x3,x4,x5,x6,x7,x8</variabl
es>
      </External>

      <External file="..\myConstraints.py"
name="impConstr4">
        <variables>x1,x2,x3,x4,x5,x6,x7,x8,,obj2</v
ariables>
      </External>
      <External file="..\myConstraints.py"
name="impConstr5">
        <variables>x1,x2,x3,x4,x5,x6,x7,x8,obj1</va
riables>
      </External>
    </Functions>

    <Distributions>
      <UniformDiscrete name='woRep_dist'>
        <lowerBound>2</lowerBound>
        <upperBound>9</upperBound>
        <strategy>withoutReplacement</strategy>
      </UniformDiscrete>
    </Distributions>

    <Optimizers>
      <GeneticAlgorithm name="GAopt">
        <samplerInit>
          <limit>60</limit>
          <initialSeed>42</initialSeed>
          <writeSteps>every</writeSteps>
          <type>min, max, min</type>
        </samplerInit>

        <GParams>
          <populationSize>50</populationSize>
          <parentSelection>tournamentSelection</par
entSelection>
          <reproduction>
            <crossover type="twoPointsCrossover">
              <crossoverProb>0.7</crossoverProb>
            </crossover>
            <mutation type="randomMutator">
              <mutationProb>0.7</mutationProb>
            </mutation>
          </reproduction>
          <fitness type="rank_crowding">
            </fitness>
          <survivorSelection>rankNcrowdingBased</su
rvivorSelection>
        </GParams>
        <convergence>
          <AHDp>0.0</AHDp>
        </convergence>
        <variable name="x1">
          <distribution>woRep_dist</distribution>
        </variable>
        <variable name="x2">
          <distribution>woRep_dist</distribution>
        </variable>
        <variable name="x3">
          <distribution>woRep_dist</distribution>
        </variable>
        <variable name="x4">
          <distribution>woRep_dist</distribution>
        </variable>
        <variable name="x5">

```

```

    <distribution>woRep_dist</distribution>
  </variable>
  <variable name="x6">
    <distribution>woRep_dist</distribution>
  </variable>
  <variable name="x7">
    <distribution>woRep_dist</distribution>
  </variable>
  <variable name="x8">
    <distribution>woRep_dist</distribution>
  </variable>
  <objective>obj1, obj2, obj3 </objective>
  <TargetEvaluation class="DataObjects"
type="PointSet">optOut</TargetEvaluation>
  <Sampler class="Samplers"
type="MonteCarlo">MC_samp</Sampler>

  <Constraint class='Functions'
type='External'>expConstr3</Constraint>
  <ImplicitConstraint class='Functions'
type='External'>impConstr4</ImplicitConstraint>
  <ImplicitConstraint class='Functions'
type='External'>impConstr5</ImplicitConstraint>

</GeneticAlgorithm>
</Optimizers>

<Samplers>
  <MonteCarlo name="MC_samp">
    <samplerInit>
      <limit>50</limit>
      <initialSeed>050877</initialSeed>
    </samplerInit>
    <variable name="x1">
      <distribution>woRep_dist</distribution>
    </variable>
    <variable name="x2">
      <distribution>woRep_dist</distribution>
    </variable>
    <variable name="x3">
      <distribution>woRep_dist</distribution>
    </variable>
    <variable name="x4">
      <distribution>woRep_dist</distribution>
    </variable>
    <variable name="x5">
      <distribution>woRep_dist</distribution>
    </variable>
    <variable name="x6">
      <distribution>woRep_dist</distribution>
    </variable>
    <variable name="x7">
      <distribution>woRep_dist</distribution>
    </variable>
    <variable name="x8">
      <distribution>woRep_dist</distribution>
    </variable>
  </MonteCarlo>
</Samplers>

<DataObjects>
  <PointSet name="placeholder"/>
  <PointSet name="optOut">
    <Input>x1,x2,x3,x4,x5,x6,x7,x8</Input>
    <Output>obj1,,obj2,obj3</Output>
  </PointSet>
  <PointSet name="opt_export">
    <Input>trajID</Input>    <Output>x1,x2,x3
,x4,x5,x6,x7,x8,obj1,,obj2,obj3,age,batchId,rank,

```

```

CD,ConstraintEvaluation_expConstr3,
ConstraintEvaluation_impConstr4,ConstraintEvaluat
ion_impConstr5,fitness,accepted </Output>
  </PointSet>
</DataObjects>

<OutStreams>
  <Print name="optOut">
    <type>csv</type>
    <source>optOut</source>
  </Print>
  <Print name="opt_export">
    <type>csv</type>
    <source>opt_export</source>
    <clusterLabel>trajID</clusterLabel>
  </Print>
</OutStreams>
</Simulate>

```