Optimization of Multi-reactor Multi-cycle Core Design Using SIMULATE and Dakota for a Shared Fuel Pool Power Plant Configuration

Benjamin Lilley

Advisor: Dr. Todd S. Palmer

Ph.D. Research Proposal

Oregon State University

School of Nuclear Science and Engineering

Oregon State University

Contents

Introduction

In the following proposal, I give an overview of core design, small modular reactors (SMRs), optimization, and some core physics concepts. I discuss some of the existing literature for core design optimization. I propose a methodology for optimizing the multi-reactor multi-cycle problem, summarize some preliminary results, and lay out a plan for achieving the framework to solve this new problem. The main objective of the dissertation is to use automation and optimization to learn more about the design of small modular reactor (SMR) cores over multiple cycles and the potential benefits of a shared fuel pool. This understanding may enable more affordable carbon-free energy and increased fuel efficiency in SMRs.

Background

In a nuclear power plant, a *core* is a region where fission reactions occur in fuel generating large amounts of heat. The basic product of *core design* is a *cycle*. A cycle is an extended period of time, usually one to four years, in which the core designer ensures that the reactor can operate at full power while meeting safety limits. In between cycles, an *outage* occurs in which the core is maintained at zero power while refueling and maintenance activities take place. Fuel in the core without enough remaining usable energy is removed to a *fuel pool*, a designated area for fuel assemblies to be stored. The used fuel stays there for several years to facilitate decay heat removal as neutron-rich fission products decay towards stability. During the outage, staff perform maintenance and inspections on many other systems and new fuel is inserted into the core. Many cycles occur over the life of the power plant, each one a chance for core design improvements.

About two thirds of commercial nuclear reactors in the United States are Pressurized Water Reactors (PWRs). Water coolant, kept in liquid phase by high pressure, is used to transfer core heat to a steam turbine that generates electricity. The core is made up of many fuel assemblies, arranged in a cylindrical shape. Each fuel assembly is several meters in height and around 20 centimeters in depth and width.

A common PWR fuel assembly configuration is 17 rods by 17 rods held a fixed distance apart by spacer grids (see Figure 1). It contains 264 fuel rods, 24 guide tubes, and one instrumentation

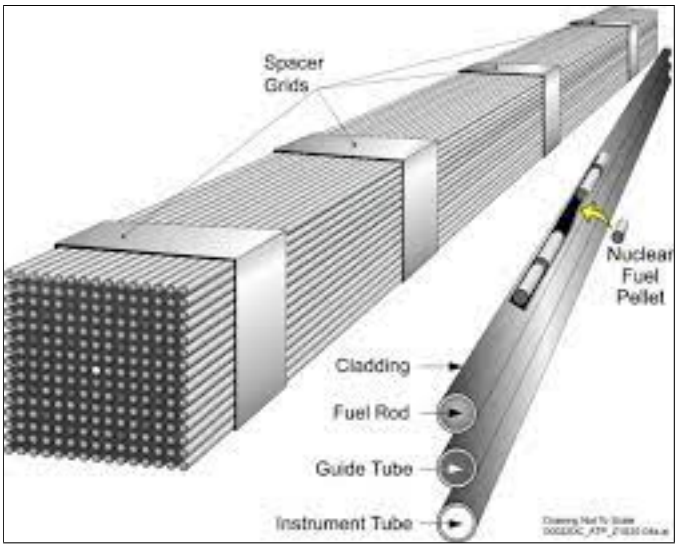

Figure 1: A typical PWR fuel assembly, rod, and pellet configuration. (Image: United States Department of Energy)

tube. The guide tubes allow for the intermittent insertion of control rods which are made of neutron absorbing material and can be used to control the neutron population and rate of nuclear reactions in the core. Each of the 264 fuel rods, also called fuel pins, is made of a zirconium-based metal tube with a stack of fuel pellets inside. Each fuel pellet contains enriched uranium oxide (UO_2). Enrichment is the mass ratio of the U-235 isotope to total uranium, expressed in weight percent. To further control local neutron generation, a neutron absorber may also coat the surface of or be mixed into the fuel pellet. These absorbers are called *burnable* because their impact on neutron population is short-lived (usually less than one cycle). Fuel assemblies may have different enrichment levels in each pin and each pin may vary axially in enrichment or burnable absorber content.

Another key concept in core design is the *loading pattern*. By replacing only a fraction of the fuel each cycle, more energy can be generated by the same fuel assembly before its retirement [1]. By replacing a decreasing fraction of the core each cycle, fuel costs can be reduced as fuel is used more efficiently, but the number of outages increases, which leads to increasing costs. Most commercial reactors have adopted loading strategies that replace a third to a quarter of the fuel assemblies each cycle, increasing energy output per assembly about 50 to 60% over the single long-cycle strategy.

Because this strategy has been adopted, the core is made of a mixture of fresh fuel and fuel that has been in the core for one or more cycles, depleting the fissile nuclides and building

up fission products. The loading pattern describes where these different fuel assemblies are located in the core. For example, fresh fuel can be placed around the periphery of the core in the "out-in" loading pattern. This pattern has a relatively flat radial power profile – good for staying below thermal safety limits – but this pattern is less fuel efficient because more neutrons escape the core region without causing further fission reactions. The choices of which used fuel to reload, and which loading pattern to use introduces more design choices.

Existing commercial PWR cores generate around 3300 megawatts of thermal energy and consist of around 190 fuel assemblies, each about 3.7 meters in height. Even though several cores may be located at a single nuclear power plant, each one usually has its own fuel pool, which is used to store fresh fuel as well as used fuel. NuScale Power is a company that is commercializing a pressurized water small modular reactor (SMR). Their Nuclear Regulatory Commission (NRC) certified design generates 160 megawatts of thermal energy from 37 fuel assemblies. Each fuel assembly has the same width and depth as in a commercial reactor, but is only two meters in height. NuScale envisions four, six, or even twelve of these cores *sharing* a fuel pool. The shared fuel pool enables a novel design choice – the ability to choose used fuel from a *different* reactor.

SMR developers, like NuScale Power, are trying to demonstrate a new generation of reactors with standardized design. The hope is for *economy of multiples* to outperform the traditional *economy of scale* – building a single reactor up to the 1500 megawatts-thermal (MW_t) scale. While SMRs benefit from increased standardization and simplified design, the small size of the core increases costs. Because the core has more surface area per volume, more neutrons leave the core without causing further fissions. This means less fuel efficiency and a power profile that is less uniform than current reactors.

SMR developers are also trying to design for flexible power operation or *load follow*. Conventional nuclear in the United States has generally operated in baseload mode, where power output is at or near 100% for the entire cycle. Load following is when power is adjusted up or down to match the electricity demand of the grid. With increasing variable renewable energy sources on the grid, more value is being placed on this flexibility to ramp power up and down. However, core designers have to design for the core to be able to operate at full power for the duration of the cycle. This introduces decreased fuel efficiency if the core is operated in load

follow mode.

The *shared* fuel pool can help solve both of these problems. Fuel assemblies near the edge of a small core will produce less energy than those in the center. Fuel assemblies in a load follow core will have excess energy after a cycle than those used in a baseload core. With the shared fuel pool, these assemblies and their characteristics can be tracked and swapped from core to core. Each fuel assembly can potentially be used more efficiently reducing costs and waste.

So, even after a nuclear power plant system has been designed and built, considerable flexibility exists in the choices of reload core design. These choices include:

- the number of fresh fuel assemblies
- the number of *types* of fresh fuel assemblies including:
  - the fresh fuel enrichment and location
  - the fresh fuel burnable absorber type, location, and concentration
- which used fuel assemblies to retain for the next cycle
- which (if any) used fuel assemblies to load from other cores
- the core loading pattern

The reload core design should meet regulatory and safety criteria. After meeting these requirements, one or more parameters of interest can be optimized. The goal of *in-core* fuel management is to optimize the choices of new fuel design, which old fuel to use again, and the loading pattern. The goal of *out-of-core* fuel management is to minimize the cost of generating energy over several cycles. Because fuel assemblies are used in several cycles before being retired, it is possible to over-optimize one cycle but then have the next cycle cost more than if the two cycles are optimized together. For commercial power reactors, the overarching goal is often to minimize the cost of fresh fuel while meeting cycle length targets.

These many design decisions lead to complicated impacts on measures of safety and the cost of operating a nuclear power plant. A core designer will need to explore a vast design space to find good solutions for the next few reloads of a group of reactors. *Heuristics*, one tool, are succinct rules that sum up expert knowledge about a system. An expert knows, for instance, that to use fuel more efficiently, placing the twice-burned fuel in the core periphery will

improve power production and also improve neutron economy. But small PWR cores sometimes have different behaviors than large PWRs, so some heuristics might not be applicable in the same way from design to design. *Software*, another tool, can automate calculations and use optimization theory to rapidly find good solutions. With the increasing complexity of design choices and the scarcity of SMR expert experience, heuristics still serve an important role, but the human-machine team will likely perform better, yielding reproducible solutions.

Optimization basics

The field of optimization provides a robust framework for finding "good" designs. In its simplest form, a *design variable*, x, is sought to optimize an *objective function*, $f(x)$, such that the minimum (or maximum) of the function can be found. Mathematically robust solutions can be found for many optimization problems, but the nuclear core design problem is relatively complicated. Design variables span many types, from discrete (e.g., the number of pins with burnable absorber) to continuous (U-235 enrichment level) to combinatorial (e.g., the loading pattern). This work uses 3D neutron diffusion software to model the physics of the system including pressure, temperatures, flow rate, and nuclear reactions. The simulation is used to compute the objective function and *constraints* (e.g., safety parameters), so the mapping from design variable to objective function is not linear or analytic. In fact, the core design problem is nonlinear, discontinuous, and multimodal. Optimality of a solution is not guaranteed without prohibitively expensive computations, so at best solutions can be labeled "good." Good solutions correspond to those that meet all constraints while minimizing the objective function(s).

Software that can automate this process is valuable, especially when expert knowledge is less developed or the problem scope becomes less intuitive, as in the multi-reactor shared fuel pool system. While many stochastic algorithms exist and have been applied to the nuclear core design problem (including Particle Swarm [2, 3, 4, 5], Cuckoo [6], Gravitational Search [7], etc.), genetic algorithms and simulated annealing have found the most success to date.

Genetic algorithms encode design variable choices into "genes" that form an individual, or a potential solution [8]. A group of individuals is created and tracked and the simulation software uses an individual's genes as inputs to generate outputs of interest (constraints and objective functions). These outputs are then combined into an evaluation of "fitness" of each individual.

The most fit solutions pass their design variable choices to the next generation by taking genes from two or more parents as well as introducing mutations. The combination of crossover and mutation allows the algorithm, assuming population size and other hyper-parameters are adequately chosen, to search for the best solution both locally and globally.

Simulated annealing algorithms emulate the process of cooling a crystalline material in which atoms find their lowest energy arrangement as temperature is slowly reduced [8]. One or more sets of design variable inputs are first evaluated. If the objective function is better than the other solutions, the new solution becomes the best solution. If the solution is not the best, then according to some probability, this solution is still accepted as the new best solution. This probability decreases as the number of iterations increases, so towards the end of the calculation only a better solution is accepted. Accepting sub-optimal solutions near the beginning of the calculation allows for a global search early in the process, while the later strictness focuses the search on the most promising local design spaces. Given sufficient time, the algorithm eventually will converge to a global optima.

However, both genetic algorithms and simulated annealing methods can be computationally expensive with much of the later time being spent with little or no improvement in the objective function. Knowing something about the physics of the system is valuable for efficiently searching the design space.

Core physics terms and safety limits

Enrichment is the ratio of U-235 to total uranium in the fuel. Current NRC regulations limit the maximum enrichment for use in a commercial power reactor to 5%. In practice, pellets are manufactured up to 4.95% enrichment. "Although no real justification can be found," this limit was established early in the development of the industry when most reactors were using around 3% enriched fuel [9]. Commercial reactors today are utilizing 4.95%, so there is more interest in changing the limit. University research reactors and many advanced reactor designs use up to 20% enrichment so there is precedent and industry effort to increase the limit. With improved fuel products and operational test assemblies, the NRC is preparing to evaluate enrichments up to 8% for existing commercial reactors to use in reloads in the coming years [10]. These designs generally use burnable absorbers to reduce their beginning of cycle (BOC) reactivity.

Reactivity is a measure of how the multiplication of neutrons occurs. Negative reactivity can be introduced into the core by adding a neutron absorber (e.g., adding soluble boron or inserting control rod), removing fuel, or increasing the flow of neutrons out of the core. Positive reactivity can be introduced into the core by adding fresh fuel, depleting a burnable absorber, or by withdrawing a control rod. A fuel assembly without burnable absorber has its highest reactivity when it is fresh. As the fuel is exposed to neutrons and fissions occur, the fuel depletes and its burnup increases.

Burnup is a measure of the thermal energy generated normalized to the mass of uranium initially present in the fuel. Burnup units are often given in gigawatt-days of thermal energy per metric ton uranium (GWd/MTU or GWd/tU). Fuel assembly fabricators use safety analysis to establish limits on the level of average burnup in a fuel pellet, pin, or assembly. The purpose of these limits is to retain fission gases in the pellet, and maintain clad integrity in normal and accident scenarios. Burnup higher than about 60 GWd/MTU in fuel introduces further concerns about clad integrity, pellet thermal conductivity, and internal rod pressure as fission gases diffuse out of the pellet [11]. The burnup level also correlates with decay heat that must be removed from assemblies in the fuel pool.

A similar measure of heat generation is used for cycle length. Cycle length is measured in effective full power days (EFPD). It is related to burnup as follows:

$$Burnup = \frac{EFPD * ThermalPower}{UraniumMass}$$

$$Burnup = \frac{365 EFPD * 160 MW_t}{9.2 tons} = 6300 \frac{MWd}{MTU}$$

Reaching a target cycle length is often an important requirement of core design. Cycle length strategy varies around the world. The operator or utility will decide on an outage schedule, often 12-, 18-, or 24-month cycles [12]. This is driven by both seasonal variations in the demand of electricity and the economic advantage of having fewer outages. Depending on the seasonality of other generation sources in the local area (e.g., high hydro generation in spring as snowpack melts), spring and fall both see decreased electricity demand and lower wholesale prices. Any downtime of a reactor should coincide with the least profitable time-of-year if the

reactor had been operating. Other considerations include the availability of outage workers and the flexibility of the core design to extend a few extra weeks if needed.

Throughout a cycle, many systems are analyzed by safety engineers. A few of the most important safety concepts for the core design problem will be described. *Criticality* refers to the condition of the neutron chain reaction. Several neutrons result from each fission reaction, but only one needs to cause another fission to maintain criticality. When the reactor is shutdown, a combination of control rods and soluble boron in the coolant maintains the reactor in a *subcritical* condition. Starting up a reactor again or increasing power is done by withdrawing control rods from the core in small steps (around 1 cm/step). When designed properly, this brief *supercritical* condition settles back to criticality through negative feedbacks.

Reactors are designed to dampen these supercritical conditions partly through what are called temperature coefficients. A negative *fuel* temperature coefficient (FTC) reflects the behavior that as fuel temperature increases, reactivity decreases because more neutrons are absorbed in the resonance region of the fertile nuclides in the fuel. A negative *moderator* temperature coefficient (MTC) similarly decreases reactivity if the water, which slows down or moderates the neutrons, increases in temperature. This is because of the thermal expansion and lower density of the water. There are fewer nuclei for neutrons to collide with and transfer kinetic energy. However, with soluble boron in the water, core designers do have to be careful. At the beginning of the cycle (BOC) when the reactivity of the core is highest, soluble boron concentration is also at its highest. If the borated water thermally expands, boron density also decreases and the MTC can become positive. In practice this means there is a maximum BOC concentration of boron [13].

Beyond neutronic feedback, it is important to accurately predict thermal hydraulic quantities. The region filled with water between four rods in a PWR is called a *channel*. The water coolant in that channel must be able to transfer heat away from the core to the rest of the system. Hot channel *peaking factors* are ratios of the *high* power channels to *core average* power channels [14]. The heat flux peaking factor, F_Q is a ratio of linear heat generation rates (kW/ft). The enthalpy rise peaking factor, $F_{\Delta h}$, is the ratio of integrated linear heat generation rates over the height of the rods (kW). Transient occurrences, like loss of coolant accidents (LOCA) and departure from nucleate boiling (DNB) conditions, that may damage the core or

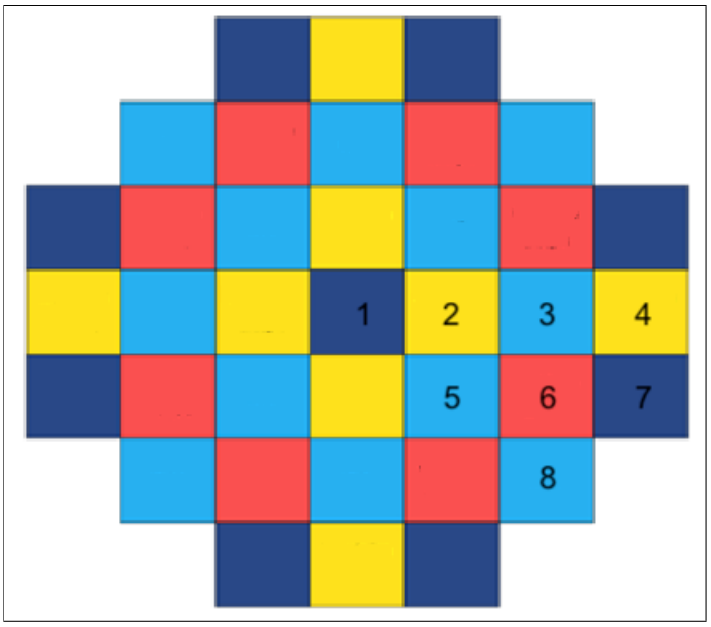

Figure 2: A 37 fuel assembly core with eighth core symmetry has 8 distinct batches. (Image: Adapted from NuScale DCA Control Rod Locations map [15])

otherwise impair the ability of the core to be cooled are analysed and limits for these peaking factors are established. A core that meets these peaking factor limits (with associated safety margins) should be able to operate at steady state without risk from these occurrences.

In PWRs, boiling within the core is not desirable. Boiling enhances conditions for corrosion of the cladding and deposition of boron from the water to the rod surfaces depressing power locally. In the NuScale core, heat is transferred from the core by natural circulation, the tendency for hotter less dense water to rise and cooler water to sink. The pressure and temperature in the core are similar to those of conventional PWRs, but natural circulation means a lower coolant flow rate and lower power density.

A *batch* is a group of fuel assemblies with the same design (e.g., enrichment) that are inserted into a core at the same time and experience the same burnup over a number of cycles. In large PWRs, especially with the original out-in loading patterns, the power profile was relatively flat. The power generation (and therefore burnup) in the center region is nearly equal to the periphery. This definition of batch and the loading of one third or one fourth of the core with fresh fuel assemblies has led to so called 3-batch or 4-batch strategies.

In a small core with less power uniformity, a core that replaces a third of the fuel each cycle would have more than 3 batches [1]. For instance, with one eighth core symmetry, a 37

[1]I will sometimes use the "n-batch" phrase for reload strategy, but the actual number of batches will be

fuel assembly core can be grouped into eight different batches (see Figure 2). The single fuel assembly in the center is its own batch. The others are grouped into five sets of four assemblies and two sets of eight assemblies. As the symmetry of the problem changes, so will the number of batches. For example, a quarter symmetric core would have up to ten batches. These batches will be tracked together and sometimes split up to accommodate symmetry limitations or to efficiently use the center fuel assembly. For example, an eight-assembly batch may be used for two cycles with the same burnup history, but then be split up into two four-assembly batches that then experience different burnups during the third cycle. Alternatively, in large PWRs, each batch in a 3-batch scheme may experience three cycles, but then a single assembly from the thrice-burned batch may be chosen for a fourth cycle in the center position of the core. This center fuel assembly becomes its own batch because it then has a different burnup history.

Each cycle fuel is removed from the core and stored (though sometimes only briefly...) in the fuel pool. The *fuel pool*, sometimes called a spent fuel pool or used fuel pool, requires additional safety analysis and limits. The specific arrangement of used and fresh fuel in the pool must maintain subcriticality. The racks hold many fuel assemblies and may or may not contain neutron absorbing materials for tighter packing and saving space [16]. Sufficient circulation of coolant is required to remove decay heat. Each fuel pool generally has enough room for all discharged fuel to cool for an extended amount of time (e.g., 5 years) and enough spare room to unload all fuel from operating core(s). The exact size, dimensions, and safety analysis will largely be outside the scope of this proposal.

Literature Review

Researchers have been optimizing the nuclear core since at least 1970 [17, 18]. Early research relied on the mathematically robust tools of the operations research field. These techniques are guaranteed to find optima, but the objective function must be linear, so approximations were made. In these early days, it was common to homogenize core materials and assume linearly decreasing reactivity with increased burnup [1]. With increasing computational capability, researchers began to move towards 2D and then 3D models. Operational experience and better models led utilities to reclaim safety margin and uprate licenses were issued to allow for increased

determined by the definition.

power levels. As the optimization problem became more complex – nonlinear objective functions and more constraints – core designers turned to stochastic algorithms like simulated annealing and genetic algorithms [19].

Optimization can focus on a number of scales because the physics of the problem is multi-scale. At the atom-scale, different materials must be tracked by number density. As the core experiences nuclear reactions, these number densities change – fuel atoms decrease in number and fission product atoms increase in number. Neutrons in the system interact with every nuclide at different rates. At the core-scale, core designers must analyze heat transfer and structural mechanics. Limiting boiling and ensuring heat removal are important considerations.

Computational resources have been too small to optimize at all scales of analysis, so researchers have worked on subsets of the problem. The main subsets are: a single fuel assembly lattice, the in-core problem, and the out-of-core problem. A fourth, generally unexplored optimization scale, is the multi-reactor system. This literature review will focus on optimization efforts in PWRs, but similar work has been done for Boiling Water Reactors (BWRs) [20, 21, 22, 23], research reactors [24, 25], and non-LWRs [26, 27].

Lattice optimization is concerned with finding the best moderator-to-fuel ratio (an indicator of reactivity associated with changes in the moderator) and geometry of the lattice (e.g., fuel pins arranged in a 15x15 lattice vs. 17x17 lattice) [28, 29]. Cores are usually designed for a specific fuel assembly dimension. A core with 193 fuel assemblies will always require 193 fuel assemblies. However, the inside dimensions of a fuel assembly could change. Extra water at the edges of a fuel assembly or the placement of guide tubes (normally filled with water) can have dramatic local effects on the neutron population due to the extra moderation of neutrons. Fuel assemblies generally moved from thicker and fewer rods (e.g., 15x15) to more rods of thinner dimensions (17x17). Spacers themselves were optimized to limit physical fretting and increase mixing of the coolant. Once a fuel assembly's structural geometry is fixed, the placement of burnable absorbers in certain pins and the degree of enrichment can be optimized to keep power peaking factors – ratios of local power density compared to a core average power density – low. The focus is often on the fuel assembly lattice multiplication factor assuming a reflective boundary (k_∞). This optimization allows the designer to use neutrons efficiently once the fuel assemblies are combined into a full core configuration. For example, Martin-del-Campo et al.

use genetic algorithms to optimize a BWR lattice [30] while recent work by Radaideh et al. uses a neural network approach [31]. The work in this dissertation assumes a fixed assembly geometry and optimizes at the larger scales.

In-core optimization aims to optimize the design of fresh fuel (enrichment and burnable absorber) and the loading pattern – the arrangement of used and fresh fuel assemblies. Loading pattern studies include Lin et al. using tabu search [32], Israeli et al. using a genetic algorithm [33], and Zameer et al. using a particle swarm algorithm [2]. Early studies often explored only a small number of fresh fuel options. Some in-core studies may not include an explicit optimization of the loading pattern. The FORMOSA-B software optimizes BWR in-core fuel management [20], while Jang et al. use a layer-by-layer learning algorithm [34].

Out-of-core studies focus on the optimization of economics over some planning horizon for a single reactor, usually at least three cycles. The out-of-core problem tracks each fuel assembly through all of the cycles in which it produces energy. Without this larger view, in-core optimization can over-optimize a single cycle at the cost of making the next cycle more expensive. Out-of-core optimization is also concerned with procurement of fuel services, scheduling, and other back end costs like dry cask storage. Several out-of-core optimization codes exist, usually relying on a simulated annealing algorithm. Ottinger and Maldonado have developed LWROpt [35, 36] and Su, Anderson, and Turinsky have developed OCEON-P [37]. The Studsvik software company and Kropaczek have developed the COPERNICUS software [38].

There has been less research effort on a multi-reactor optimization framework. Core designers at the Almaraz NPP have investigated the swapping of fuel between reactors with a shared fuel pool [39], but only for the last cycles before shut down and decommissioning. With the arrival of SMRs with many cores sharing a fuel pool, new definitions and solutions to the core reload optimization problem present the opportunity for significant cost savings.

Proposed Methodology

The main goal of this dissertation is to use automation and optimization to learn more about the design of SMR cores over multiple cycles and the potential benefits of a shared fuel pool. This understanding will enable more affordable carbon-free energy and increased fuel efficiency in SMRs.

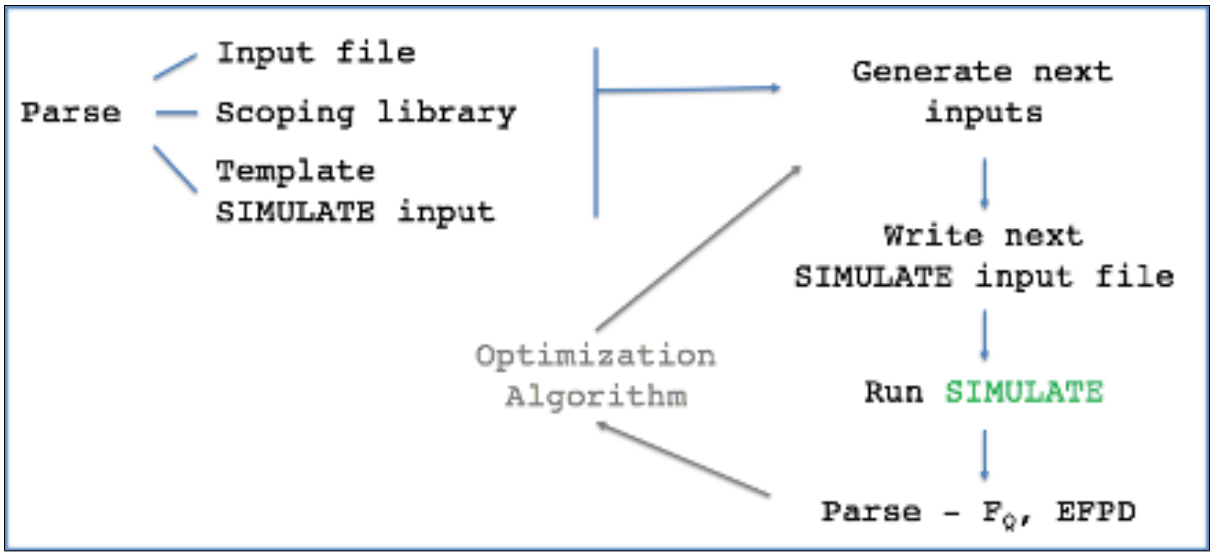

Figure 3: Shows the flow of the core design automation for a single cycle (inner loop).

CASMO-4 and SIMULATE-3 are nuclear core modeling codes developed by Studsvik Scandpower [40, 41]. CASMO-4, a 2D lattice transport code, is used to generate a scoping library with discrete gadolinia designs for a 17x17 fuel assembly. In this library, U-235 enrichment is interpolated and can be treated as a continuous or a discrete variable. SIMULATE-3 is a nodal diffusion code used to model the 3D core. In the preliminary model for this work, each assembly is treated as four nodes (two-by-two) radially and fifteen nodes axially. The core is modeled in quarter-core symmetry with reflecting boundaries to reduce computation time. A Python package, called Optimus, has been developed to read various input files, generate SIMULATE-3 input cases, execute them, and parse relevant results (see Figure 3).

The template SIMULATE input files describe the NuScale 37 fuel assembly, 160 megawatt thermal (MW_t) design based on the Design Certification Application (DCA) submitted to the NRC [15]. Description of the reflector regions is informed by a study of the Multi-Application Small Light Water Reactor (MASLWR), a predecessor of the NuScale module, using greater than 5% U-235 enrichments [42]. Because the reflector region does not multiply neutrons, the enrichment levels of the adjacent core have only a small impact on cross sections. Subsequent to the initial work on this project, NuScale Power has announced new product offerings. The number of fuel assemblies will remain the same, but power, temperature, and other secondary conditions may be modified. The code will handle these changes easily because no change in the geometry is expected.

The scoping library assembles a variety of cross sections for standard 17x17 fuel assemblies. It includes enrichments from 1% to 7% and 29 distinct gadolinia patterns. The number of gadolinia pins varies from 0 to 28 in increments of 4. The U-235 enrichments are reduced in gad pins due to differences in thermal properties of the gadolinia and uranium oxide. This precludes these pins from being limiting in the safety evaluation. The weight percent of gadolinia in these pins ranges from 0 to 8 in increments of 2. The library also includes reflector cross sections for the radial, top, and bottom regions adjacent to the core.

The number of design variables varies with the number of *types* of fuel assemblies in each in-core cycle (e.g., assuming one eighth core symmetry, 13 assemblies could have three types grouped by 1, 4, and 8 assemblies). Each fuel assembly type is assigned an enrichment, a gadolinia pattern, and a number of fuel assemblies. The choice of used fuel could be added as a design variable, but the default action is to choose fuel with the most remaining reactivity. The loading pattern is the last in-core design variable. The out-of-core design variable is the number of fresh fuel assemblies.

Given these inputs and their ranges, optimization is the process of minimizing or maximizing an objective function given constraints. A maximum value for the heat flux hot channel peaking factor, F_Q is used as a safety constraint. The maximum boron concentration constraint is used to prevent the MTC from becoming positive at BOC. The cycle length must reach a minimum target, reported as effective full power days (EFPD). For example, a minimum of 694 EFPD is required to meet the criteria for a 24 month cycle with 95% capacity factor.

The objective function to minimize is the new fuel cost (million $) for the single cycle (inner loop) optimization. For the multi-reactor multi-cycle (outer loop) problem, the objective is to minimize fuel cycle cost normalized to electricity production ($/MWh). Other constraints (e.g., burnup limit) and objective functions (e.g., maximize neutron economy) will be added in future studies as the Python code matures.

After choosing the new fuel design and which fuel is retained from previous cycles (and other reactors), a loading pattern is obtained to describe the arrangement of the fuel. The small size of an SMR core limits the size of this search space. Naively, a core with 37 fuel assemblies and with one-eighth core symmetry has 8 assemblies with 8! = 40,320 permutations. However, imposing eighth core symmetry creates batches of one, four, or eight assemblies and the number

of viable patterns is just just 36. Core symmetry is important to avoid axial tilt and xenon oscillations in large PWRs, but a smaller core may be more resilient to these changes.

Using a heuristic based on the reactivity of a fresh or used fuel assembly and the small number of loading patterns greatly reduces the design space variations and allows for a simple stochastic sampling approach for the initial design space search. Further work can generalize this loading pattern generator to include quarter-core symmetric designs (or even asymmetric cores), but will require a more sophisticated approach. A branch and bound method or a heuristics-based fuel assembly swapping method might make the challenge feasible to tackle with limited computational resources.

A simple cost algorithm will be used to estimate the front-end fuel cycle costs: ore, conversion, enrichment, and fuel fabrication [43]. Prices for these services in 2011 are used to calculate the cost of a batch of fresh fuel, however these prices can be easily updated to current conditions if desired. The time-value of money and financing costs (interest and discount rates) are neglected at this early stage, but may be added. Manufacturing costs for added complexity are ignored as well as the lost revenue and cost of an outage. Back-end fuel cycle costs will be ignored, but may be important depending on a country's high level waste strategy.

Dakota is a multi-use tool developed by Sandia National Laboratory [44]. It is used to guide the creation of SIMULATE input files and use the objective function evaluations to create better inputs for the next iteration. Several optimization algorithms exist within the Dakota package and they can be swapped relatively easily. The inner loop calculation is a single-cycle in-core optimization loop. Then an outer loop optimizes on the number of fresh fuel assemblies that each reactor will require for each cycle.

As each single cycle optimization is completed, the best evaluation will be saved and Optimus will parse out all fuel assembly information for potential reinsertion in another cycle. Typically 2/3 of the fuel assemblies in the current cycle will be used in the next cycle and are reinserted. With many used fuel batches to be called upon this outer loop will be the best way to keep track of and use them wisely. The code can model a one or two reactor system optimized over a multi-cycle planning horizon. The code will eventually be expanded to optimize up to 12 reactors over several cycles.

Preliminary Research and Results

This chapter presents the research progress made in putting together the automation of the optimization process and highlights the research included in my first conference paper (see Appendix). The Optimus Python package automates the generation of SIMULATE-3 input files, executes them, and parses key output values. The code runs serially and each SIMULATE-3 evaluation, from beginning of cycle (BOC) to end of cycle (EOC) exposure takes about 1.5 seconds.

Prior to interfacing with Dakota, two studies were performed with different enrichment limits. First, the U-235 enrichment was limited to the range 5.0 and 7.0%. To maximize the cycle length (as an objective function) within the F_Q constraint, the loading pattern converges to a low leakage pattern in which twice-burned fuel is placed around the periphery of the core.

Second, the U-235 enrichment was limited to the current 5.0% licensing limit for the existing nuclear power plant fleet in the United States. To maximize the cycle length within the F_Q constraint, the loading pattern reaching 1146 EFPD (3.1 years) has the fresh fuel again in the middle ring. All twelve of the twice-burned fuel assemblies reside in the periphery illustrating the lower leakage pattern is valuable in this low power density core. The trade off between F_Q and cycle length for both studies is illustrated in Figure 4. Note these are pre-optimization results.

After interfacing Optimus with the Dakota software, a conference paper was written and peer reviewed (see Appendix for full paper). The paper discusses a two core multi-cycle system. Dakota and Optimus are used to optimize for fuel cycle cost for each reactor individually over 10 years. These results are compared to the same cores operating in a tandem manner so that fuel can be passed between them. With the increased overhead of tracking used fuel batches by reading and writing to file and with Dakota computations, each case evaluation takes about 7 seconds. Results show that, in this narrow case study, the shared fuel pool case outperforms the independent fuel pool case. Fuel costs in the shared fuel pool case are 2.4% lower and fuel utilization is increased 0.8%.

These demonstrations begin to show that the Optimus package interfacing with Dakota, SIMULATE-3, and the CASMO-4 library can produce good core designs more rapidly than stochastic sampling of the design space described earlier. Nested feedback loops use Dakota to

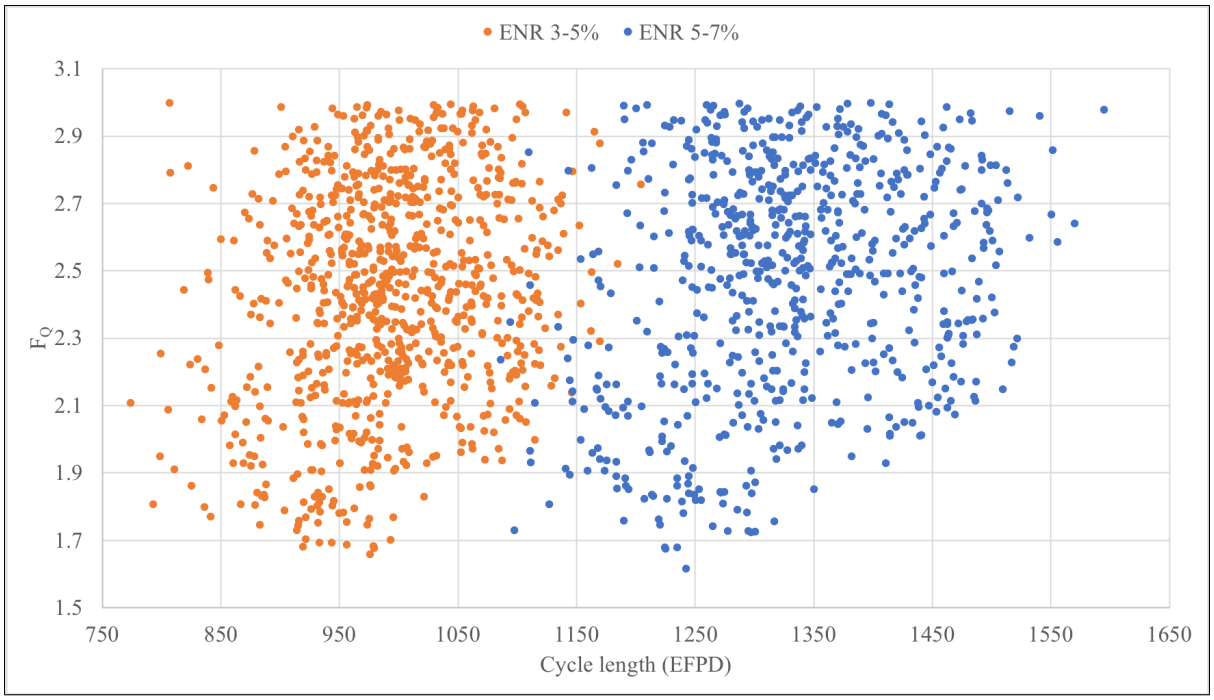

Figure 4: Demonstrates the rapid generation and evaluation of constraints for the two cases.

efficiently explore batch size (outer loop) and the in-core design space (inner loop). Overhead computation is considerable, so there is likely room for improvement (bringing 7 seconds per case closer to the SIMULATE evaluation time of approximately 1.5 seconds per case).

Potential Challenges

There are three foreseeable issues with the plan moving forward. First, the algorithm to generate loading patterns can quickly grow the problem to unmanageable size. Relaxing eighth core symmetry to just quarter core symmetry, 9! = 362,880 loading patterns exist and to test them all in concert with all the other design variables would be computationally-prohibitive. Much of the literature is devoted to solving just the loading pattern challenge in large reactors, so there are likely ways to use heuristics to make this problem manageable.

Second, the used fuel options will grow with increasing time. The choice of fuel to reinsert is itself a design variable which means that each additional cycle may require more computational time. However, this may not be a large issue since many of the fuel batches will have similar properties (similar burnup and remaining reactivity). This is contrary to the loading pattern differences which cause step differences in the objective functions due to the importance of

neutron economy.

Third, the computational needs of optimizing 12 reactors for up to 20 cycles each will be high. The first problem tackled was only 2 reactors with a combined total of 15 cycles (inner loops), but even with limited discrete options, one outer loop takes 3 hours. Depending on the time to converge the outer loop, these calculations may take too long. Other batch sizes will be explored in future papers, but for the first paper a 3-batch reload is assumed such that each new cycle incorporates only 12 or 13 new fuel assemblies (out of 37 in the core). However, these estimates have not included the impact of parallelizing the SIMULATE evaluations which many optimization methods can accommodate. With the university's computational resources, these challenges should be resolved in a reasonable time.

Research Questions and Timeline

There are several research questions that will lead to new understanding of SMR core design and the opportunities of a shared fuel pool. These will be down-selected and serve as the basis for the three papers in a manuscript dissertation.

Research questions:

1. Does the multi-core software find good designs in a reasonable time?
    (a) On a small problem, does the software reproduce the best design, given constraints, more quickly than trying all combinations?
    (b) What optimization algorithms and hyper parameters can achieve good designs most efficiently?
2. Are there advantages (e.g., increased fuel utilization, lower fuel cycle costs) to sharing a fuel pool?
    (a) How do these advantages scale with the number of reactors sharing a fuel pool?
    (b) Is there an advantage in planning the first cycles of each reactor?

Secondary (or post-doctoral) research questions:

1. What loading patterns are viable for equilibrium and transient cycles?

(a) Is core symmetry necessary?

2. In a shared fuel pool, is there an advantage for energy recovery after load following?

3. In a 1-batch core, what core design will maximize cycle length?

   (a) What advantages and challenges are associated with asymmetric fuel assemblies?

   (b) What advantages and challenges are associated with axially varying fuel assemblies?

   (c) What advantages and challenges are associated with using fuel enrichments between 5 and 8%?

4. Which heuristics from large PWR operating experience translate to small PWRs and what are new heuristics?

I expect this research will show a clear advantage for using optimization for efficient exploration of the design space and an advantage for cores sharing a used fuel pool compared to independent operation. Especially in scenarios where cores are not operated according to forecasts, the added flexibility of swapping fuel will increase fuel utilization and reduce fuel costs.

I plan to complete a manuscript PhD dissertation and defend in June 2022 (see Figure 5). A first working paper (see Appendix) has been accepted and a presentation is being drafted for the Mathematics and Computation conference in October 2021. Further working papers will be presented at the 2021 ANS Winter Meeting and PHYSOR in 2022. Manuscripts will be

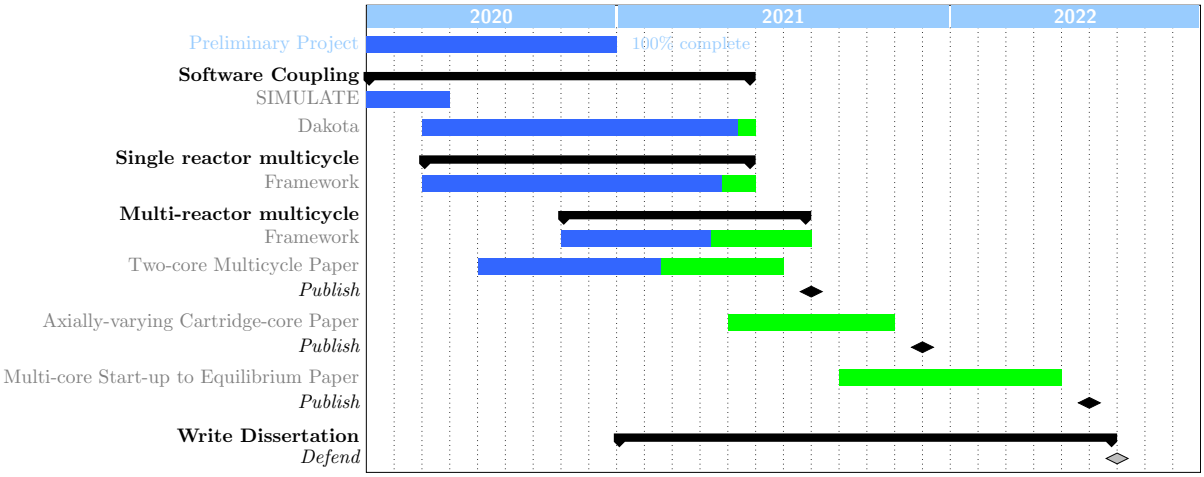

Figure 5: Proposed research milestones

submitted to peer reviewed journals such as *Nuclear Engineering and Design*, *Nuclear Science and Engineering*, and *Annals of Nuclear Energy*. I am an intern at NuScale Power and I will be working outside of the company's proprietary information in order to facilitate publishing. There is a large quantity of publicly available information from the NRC and industry journals. Most remaining inputs can be estimated to give meaningful results without disclosing proprietary information. In the event of using proprietary information, only generic results will be shared in the papers and NuScale will provide additional screening of prospective articles.

Conclusion

Rapid generation and evaluation of core designs has been demonstrated. Multi-cycle two-reactor optimization has been demonstrated with the Python package interfaced with Dakota, a multipurpose software tool with optimization algorithms. In addition to extending the scope to include new constraints and objective functions, design variables will expand to include axial variation in fuel assemblies and multi-cycle batch size (e.g., 12 fuel assemblies for one cycle, then 9 assemblies in the next, etc.). This research has demonstrated potential cost savings in the new approach and will explore new research questions around a vast new optimization problem and enable greater fuel efficiency and more affordable nuclear energy.

Acknowledgements

Thank you to Dr. Todd S. Palmer and the engineers in the core and fuels group at NuScale Power for their continued mentorship. Thank you to the folks in COE Tech Support, especially Chris Thompson, for tirelessly supporting graduate students in their efforts at computational research.

References

[1] MJ Driscoll and T Downar. *Linear Reactivity Model*. American Nuclear Society, La Grange Park, Illinois, 1989.

[2] Aneela Zameer, Muhammad Muneeb, Sikander M Mirza, and Muhammad Asif Zahoor Raja. Fractional-order particle swarm based multi-objective pwr core loading pattern optimization. *Annals of Nuclear Energy*, 135:106982, 2020.

[3] Anderson Alvarenga de Moura Meneses, Marcelo Dornellas Machado, and Roberto Schirru. Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor. *Progress in Nuclear Energy*, 51(2):319–326, 2009.

[4] Marcel Waintraub, Roberto Schirru, and Claudio MNA Pereira. Multiprocessor modeling of parallel particle swarm optimization applied to nuclear engineering problems. *Progress in Nuclear Energy*, 51(6-7):680–688, 2009.

[5] Roberto P Domingos, Roberto Schirru, and Cláudio MNA Pereira. Particle swarm optimization in reactor core design. *Nuclear Science and Engineering*, 152(2):197–203, 2006.

[6] R Akbari, D Rezaei Ochbelagh, A Gharib, JR Maiorino, and F D'Auria. Small modular reactor full scope core optimization using cuckoo optimization algorithm. *Progress in Nuclear Energy*, 122:103271, 2020.

[7] Wagner Figueiredo SACCO, Cassiano RE de Oliveira, Cláudio Márcio do Nascimento Abreu PEREIRA, et al. The gravitational attraction algorithm: a new metaheuristic applied to nuclear reactor core design optimization problem. 2005.

[8] Jasbir Singh Arora. *Introduction to optimum design*. Elsevier, 2004.

[9] International Atomic Energy Agency. Light water reactor fuel enrichment beyond the five per cent limit: Perspectives and challenges. Technical report, IAEA-TECDOC-1918, 2020.

[10] Nuclear Regulatory Commission. Appendix a: Fuel burnup and enrichment extension preparation strategy. Technical report, ML19242E192, 2019. Draft.

[11] PE MacDonald, JS Herring, DC Crawford, and LE Neimark. Technical basis for the proposed high efficiency nuclear fuel program. Technical report, Lockheed Martin Idaho Technologies Co., Idaho National Engineering and ..., 1998.

[12] International Atomic Energy Agency. Reload design and core management in operating nuclear power plants. Technical report, IAEA-TECDOC-1898, 2020.

[13] James J Duderstadt. *Nuclear reactor analysis*. Wiley, 1976.

[14] JR Lamarsh. Introduction to nuclear reactor theory, 2nd ed., 1983.

[15] NuScale Power. Nuscale standard plant design certification application, chapter four reactor. Technical report, ML20036D348, 2020. Rev 4.

[16] Spent fuel pools. https://www.nrc.gov/waste/spent-fuel-storage/pools.html. Accessed: 2021-04-27.

[17] Kotaro Inoue. Fast-reactor-core design optimization by linear programming. *Nuclear Science and Engineering*, 39(3):394–398, 1970.

[18] Samuel H Levine. In-core fuel management educational module. *Nuclear Technology*, 53(3):303–325, 1981.

[19] Paul J Turinsky and Geoffrey T Parks. Advances in nuclear fuel management for light water reactors. In *Advances in nuclear science and technology*, pages 137–165. Springer, 2002.

[20] Brian R Moore, Paul J Turinsky, and Atul A Karve. Formosa-b: A boiling water reactor in-core fuel management optimization package. *Nuclear technology*, 126(2):153–169, 1999.

[21] Faisal Yousef Odeh. *Optimum core design studies for long-life small modular boiling water reactors*. PhD thesis, Purdue University, 2016.

[22] Faisal Y Odeh and Won Sik Yang. Core design optimization and analysis of the purdue novel modular reactor (nmr-50). *Annals of Nuclear Energy*, 94:288–299, 2016.

[23] Keith Everette Ottinger. Multi-cycle boiling water reactor fuel cycle optimization. 2014.

[24] Evert B Schlünz, Pavel M Bokov, and Jan H van Vuuren. Research reactor in-core fuel management optimisation using the multiobjective cross-entropy method. In *Proceedings of the 2014 International Conference on Reactor Physics (PHYSOR 2014), Kyoto*, 2014.

[25] Benjamin R Betzler, David Chandler, David H Cook, Eva E Davidson, and Germina Ilas. Design optimization methods for high-performance research reactor core design. *Nuclear Engineering and Design*, 352:110167, 2019.

[26] Edouard Hourcade, Frédéric Jasserand, Karim Ammar, and Cyril Patricot. Sfr core design: a system-driven multi-criteria core optimisation exercise with triad. In *Fast Reactors and Related Fuel Cycles: Safe Technologies and Sustainable Scenarios (FR13). COMPANION CD-ROM. Proceedings of an International Conference*, 2015.

[27] Ryan Stewart and Todd S Palmer. Utilizing a reduced-order model and physical programming for preliminary reactor design optimization.

[28] Denis Janin. *Core Design and Optimization of the High Conversion Small Modular Reactor*. PhD thesis, Technische Universität München, 2018.

[29] M Zaidabadi Nejad and GR Ansarifar. Optimal design of a small modular reactor core with dual cooled annular fuel based on the neutronics and natural circulation parameters. *Nuclear Engineering and Design*, 360:110518, 2020.

[30] Cecilia Martin-del Campo, Juan Luis François, Roberto Carmona, and Ivonne P Oropeza. Optimization of bwr fuel lattice enrichment and gadolinia distribution using genetic algorithms and knowledge. *Annals of Nuclear Energy*, 34(4):248–253, 2007.

[31] Majdi I Radaideh, Isaac Wolverton, Joshua Joseph, James J Tusar, Uuganbayar Otgonbaatar, Nicholas Roy, Benoit Forget, and Koroush Shirvan. Physics-informed reinforcement learning optimization of nuclear assembly design. *Nuclear Engineering and Design*, page 110966, 2020.

[32] Chaung Lin, Jiing-Iuan Yang, Kuan-Jan Lin, and Zheng-De Wang. Pressurized water reactor loading pattern design using the simple tabu search. *Nuclear science and engineering*, 129(1):61–71, 1998.

[33] Ella Israeli and Erez Gilad. Novel genetic algorithm for loading pattern optimization based on core physics heuristics. *Annals of Nuclear Energy*, 118:35–48, 2018.

[34] Chang Sun Jang, Hyung Jin Shim, and Chang Hyo Kim. Optimization layer by layer networks for in-core fuel management optimization computations in pwrs. *Annals of Nuclear Energy*, 28(11):1115–1132, 2001.

[35] Keith E Ottinger, P Eric Collins, Nicholas P Luciano, and G Ivan Maldonado. Smr fuel cycle optimization and control rod depletion using nestle and lwropt. *Proceedings of ANFM*, 2015.

[36] Keith E Ottinger and G Ivan Maldonado. Smr fuel cycle optimization using lwropt. *Journal of Nuclear Engineering and Radiation Science*, 3(1), 2017.

[37] Kenneth A Anderson, Paul J Turinsky, and Paul M Keller. Improvement of oceon-p optimization capabilities. *Transactions*, 96(1):201–202, 2007.

[38] David J Kropaczek. Copernicus: A multi-cycle optimization code for nuclear fuel based on parallel simulated annealing with mixing of states. *Progress in Nuclear Energy*, 53(6):554–561, 2011.

[39] J.A. Bermejo L. Encinas, A. Ortego. Last cycle optimization activities at cnat. Studsvik CMS Users Group, 2019.

[40] Malte Edenius, Kim Ekberg, Bengt H Forssén, and Dave Knott. Casmo-4, a fuel assembly burnup program, user's manual. *Studsvik0SOA-9501, Studsvik of America, Inc*, 1995.

[41] Studsvik Scandpower. *SIMULATE-3: Advanced Three-Dimensional Two-Group Reactor Analysis Code-User's Manual (University Release)*, 2009.

[42] Alexey Soldatov and Todd S Palmer. A five-year core for a small modular light water reactor. *Nuclear science and engineering*, 167(1):77–90, 2011.

[43] Nicholas Tsoulfanidis. *The Nuclear Fuel Cycle*. American Nuclear Society, 2013.

[44] Keith Dalbey, Michael S Eldred, Gianluca Geraci, John Davis Jakeman, Kathryn Anne Maupin, Jason A Monschke, Daniel Thomas Seidl, Laura Painton Swiler, Anh Tran, Friedrich Menhorn, et al. *Dakota A Multilevel Parallel Object-Oriented Framework for Design Optimization Parameter Estimation Uncertainty Quantification and Sensitivity Analysis: Version 6.12 Theory Manual.*, 2020.

Appendix - MC Conference Paper

The remaining pages in this document are a paper for The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering.

ANS M&C 2021 - The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering · Raleigh, North Carolina · October 3–7, 2021

MULTI-REACTOR, MULTICYCLE OPTIMIZATION OF CORE RELOAD DESIGN FOR AN SMR POWER PLANT

Ben Lilley, Todd S. Palmer

School of Nuclear Science and Engineering
Oregon State University
1791 SW Campus Way
Corvallis, OR 97331

lilleybe@oregonstate.edu, todd.palmer@oregonstate.edu

ABSTRACT

Finding a "good" solution for the core reload design problem is difficult without expert knowledge and software automation. Small modular reactors (SMRs) with a shared used fuel pool demonstrate a novel opportunity for fuel cycle optimization. A Python package has been developed and interfaced with optimization software and core modeling software, to automate design space exploration for the multi-reactor multicycle problem. A case study demonstrates 2.4% cost savings for enabling the shuffling of fuel assemblies between reactors.

KEYWORDS: core design, fuel management, loading pattern, multi-reactor, optimization

1. INTRODUCTION

The goal of in-core fuel management, in a batched reload scheme common to Light Water Reactors (LWRs), is to choose the shuffle pattern, the specific used fuel to employ in the next cycle, new fuel U-235 enrichment and burnable absorber placement for a given cycle such that safety constraints are met over the next operating cycle [1]. The goal of out-of-core fuel management is to optimize an objective (e.g., minimize cost of energy) over several cycles by choosing the number of new fuel assemblies. Core designers need to explore a vast design space to find "good" solutions (optimality is not guaranteed) for the next few reloads of a reactor or power plant. Software that can automate this process is valuable, especially when expert knowledge is less developed, or the problem scope becomes less intuitive.

There have been some investigations into optimizing small modular reactor (SMR) cores [2]. Additionally, core designers at the Almaraz NPP have investigated the shuffling of fuel between reactors with a shared used fuel pool for lower cost end-of-plant-life [3]. However, with the arrival of SMRs with many modules sharing a used fuel pool, new definitions and solutions to the core loading optimization problem present the opportunity for significant cost savings.

2. METHODS

CASMO4 and SIMULATE3 are nuclear core modeling codes developed by Studsvik Scandpower [4-5]. In this work, CASMO4, a 2D lattice transport code, is used to generate a scoping library with discrete gadolinia designs. U-235 enrichment is interpolated, but to keep the problem size manageable, is treated as a discrete

variable. SIMULATE3 is a nodal diffusion code used to model the 3D small modular reactor. In the model for this work, each fuel assembly is treated as four nodes (two-by-two) radially and fifteen nodes axially.

The core is modeled in quarter-core symmetry, with eighth-core symmetry of the fuel assemblies (neglecting rotation). A Python package has been developed to read input files, generate SIMULATE cases, execute them, and parse relevant results. This package interfaces with Dakota, a multipurpose software tool from Sandia National Lab [6]. Dakota enables the exploration of several different optimization algorithms that can handle various levels of complexity (multimodal, noisy, nonlinear) in optimization problems. The computational flow of the in-core calculation is shown in Figure 1.

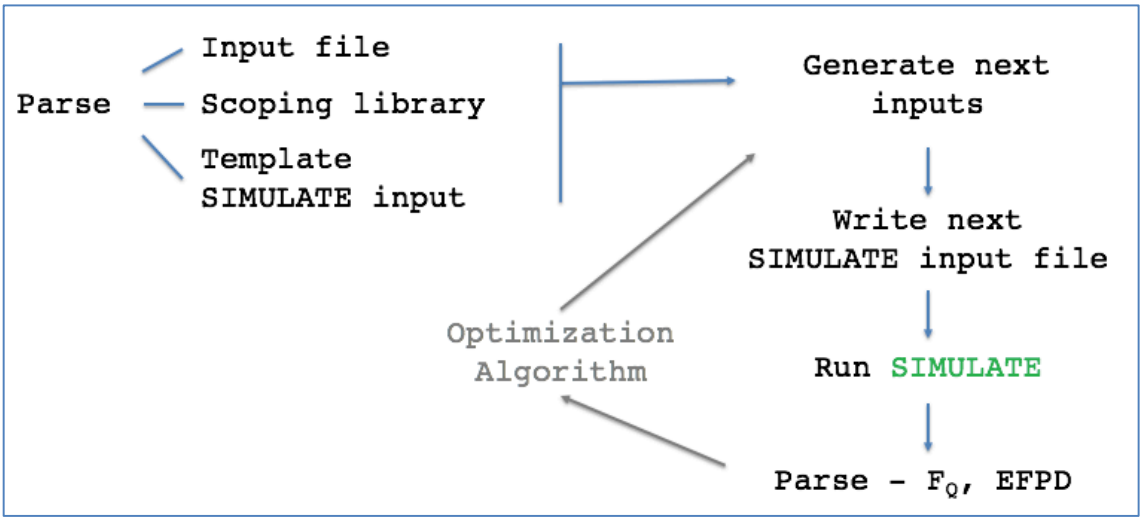

Figure 1: Shows the pseudocode for the core design automation.

The template SIMULATE input file describes a NuScale reactor with 37 fuel assemblies each 200 cm in height. The 160 megawatt thermal (MWt) natural circulation design is based on the Design Certification Application (DCA) submitted to the Nuclear Regulatory Commission (NRC) [7]. Description of the reflector regions is informed by a study of the predecessor Multi-Application Small Light Water Reactor (MASLWR) using greater than 5% enrichments [8]. Though other batch sizes will be explored in the future, a 3-batch reload is assumed such that each new cycle incorporates 12 or 13 new fuel assemblies, and the resulting out-of-core problem (outer loop) is a binary choice (12 or 13) for each reload cycle.

The scoping library assembles a variety of cross sections for standard 17x17 fuel assemblies. It includes enrichments from 1% to 5% and 29 distinct gadolinia patterns. The number of gadolinia pins varies from 0 to 28 in increments of 4. The weight percent of gadolinia in these pins range from 0 to 8 in increments of 2. The library also includes reflector cross sections for the radial, top, and bottom regions adjacent to the core.

2.1. The Optimization Problem

Optimization is a mathematical process for minimizing or maximizing an objective function given constraints. Though the number of choices varies with the number of types of fresh fuel assemblies (e.g., assuming symmetry, 13 assemblies could have three types that include 1, 4, and 8 assemblies or a single type including all 13 assemblies), the decision variables are defined generally in Table I.

Table I. Design Variables

Variable	Type	Range or size
New fuel U-235 enrichment	Discrete	1-5%
Gadolinia loading (number of pins and weight percent)	Discrete	29
Shuffle pattern	Discrete	36

The heat flux hot channel peaking factor, F_Q is used as a safety constraint set at the level of a typical large PWR. A second constraint is the cycle length, reported as effective full power days (EFPD). In this case study, an in-core optimization must meet a minimum cycle length of 347 EFPD or 694 EFPD corresponding to 1 or 2 years of full power operation at 95% capacity factor. A final constraint is placed on beginning of cycle boron concentration to ensure a negative moderator temperature coefficient.

The objective function of the in-core problem is to minimize the cost of fresh fuel (measured in millions of dollars). For the out-of-core problem, the goal is to minimize the fuel cycle cost (FCC) measured as the cost of electricity in dollars per megawatt-hour ($/MWh$_e$). Other constraints (e.g., a maximum exposure limit) and objective functions (e.g., minimize neutron leakage) will be added in future studies as the Python code matures.

Fresh fuel costs and fuel cycle costs are computed as a combination of costs from the front end of the fuel cycle. These include uranium ore, conversion, enrichment, and fuel fabrication [9]. Typical values for these services in 2011 are taken from Tsoulfanidis [10]. Back-end costs, outage costs, and financing costs may be addressed in future versions.

New fuel is split into one or two types. If a center fuel assembly is included, this single fuel assembly is assigned both an enrichment and gadolinia loading. The remaining 12 fuel assemblies are assigned a second enrichment and gadolinia loading. Though initially identical, these 12 fuel assemblies will experience different burnup over their first cycle and are treated as two separate batches, one batch with four assemblies and one batch with eight. The remaining 25 or 24 fuel assemblies are chosen from the available used fuel (from any reactor). The fuel is tracked according to batch and those batches with the greatest remaining reactivity are chosen.

After choosing the fuel assemblies to be loaded, a shuffle pattern is determined to describe the movements of the fuel. The small size of an SMR core limits the size of this search space. Naively, a core with 37 fuel assemblies and with one-eighth core symmetry has 8 assemblies with 8! = 40,320 permutations. However, imposing eighth-core symmetry such that assemblies are gathered in batches of four or eight, the number of patterns considered is just 36. This greatly reduces the design space variations for this case study.

The case study follows the limitations above and includes two reactors. The first is operated on one-year cycles while the second is operated on two-year cycles. Cycle dates are staggered because only one reactor at a time will be refueled. The second reactor is started after the end of the first reactor's first cycle to illustrate that only the very first reactor in a multi-pack needs to have all fresh fuel. In this case study, the first cycle of the second reactor has 37 fresh fuel assemblies to facilitate comparison between independent used fuel pools and shared used fuel pool.

An approximation is made to enable the swapping of fuel between two reactors. More recent versions of SIMULATE have the ability to swap fuel between more than one reactor. However, in this shared fuel pool case study, the two reactors are modeled as a single reactor operated in series with pseudo-dates and identical operating conditions. Rules for concurrent real dates are added to ensure that fuel from an earlier cycle can only be loaded in one of the reactors at a time. These pseudo-dates, in combination with a

shutdown cooling calculation that computes isotopic changes in the fuel while not in an operating reactor, introduce uncertainty.

Bahadir demonstrated that to some extent, positive and negative reactivity changes during shutdown cancel out during typical outage times of 7 to 100 days. After 100 days further shutdown cooling introduces net negative reactivity. Systematically lengthening most shutdown cooling times will push batches into this negative reactivity region, so final cost estimates will likely be *overestimated* as extra U-235 will be needed to compensate. The largest SIMULATE errors will occur for fuel assemblies that are re-inserted four or more years after their last operating cycle. [11]

Table II: The planning horizon shows the number of fresh fuel assemblies, the beginning of cycle (BOC) dates, and the target cycle length. Reactor 1 operates ten annual cycles and reactor 2 operates five biannual cycles.

Reactor	Cycle	Number of fresh fuel assemblies	Year	Month	Day	Cycle length (EFPD)	Pseudo-BOC date
1	1	37	2020	Jan	1	347	1-Jan-00
1	2	12 or 13	2021	Jan	1	347	1-Jan-01
2	1	37	2021	Jan	15	694	1-Jan-02
1	3	12 or 13	2022	Jan	1	347	1-Jan-04
1	4	12 or 13	2023	Jan	1	347	1-Jan-05
2	2	12 or 13	2023	Jan	15	694	1-Jan-06
1	5	12 or 13	2024	Jan	1	347	1-Jan-08
1	6	12 or 13	2025	Jan	1	347	1-Jan-09
2	3	12 or 13	2025	Jan	15	694	1-Jan-10
1	7	12 or 13	2026	Jan	1	347	1-Jan-12
1	8	12 or 13	2027	Jan	1	347	1-Jan-13
2	4	12 or 13	2027	Jan	15	694	1-Jan-14
1	9	12 or 13	2028	Jan	1	347	1-Jan-16
1	10	12 or 13	2029	Jan	1	347	1-Jan-17
2	5	12 or 13	2029	Jan	15	694	1-Jan-18

The out-of-core calculation iterates on reload size for each cycle in the planning horizon (see Table II). An in-core calculation (inner loop) completes around 100 SIMULATE cases using a genetic algorithm with default settings from Dakota. Genetic algorithms are a group of stochastic global search optimization algorithms that apply mutation and crossover between individuals (sets of design input variables) in a population. The best of these cases is then used as an input to future cycles in both reactors.

3. RESULTS

Each SIMULATE case takes about 6.7 seconds including all the overhead of reading and writing and interfacing. A single in-core cycle optimization loop takes approximately 12 minutes running in serial. A case study has been performed to compare two reactors with *independent* used fuel pools and two reactors with a *shared* used fuel pool.

3.1 In-core optimization

The in-core optimization loop has either three or five input variables to optimize depending on the number of types of new fuel. Enrichment has the largest impact on fuel costs, but increased gadolinia content also decreases the fuel cost objective function because fuel pins with gadolinia have reduced fissile content. Gadolinia and shuffle choice impact the total peaking factor (F_Q) output but may not be explored sufficiently if the F_Q constraint is not active. A constraint is active when the optimum would be different if the constraint did not exist.

Genetic algorithms are stochastic optimization strategies that are relatively slow to converge. With a small set of discrete input parameters, convergence can be demonstrated more quickly. Three examples of convergence are given in Figures 2 to 4. The full core is reloaded (Figure 2) and sufficient U-235 is needed to reach the two-year cycle length while meeting safety constraints. In Figure 3, only 13 fuel assemblies are fresh. The required enrichment is lower due to the need for reaching only one year of full power operation. Figure 4 shows typical convergence of the new fuel cost in a cycle optimization.

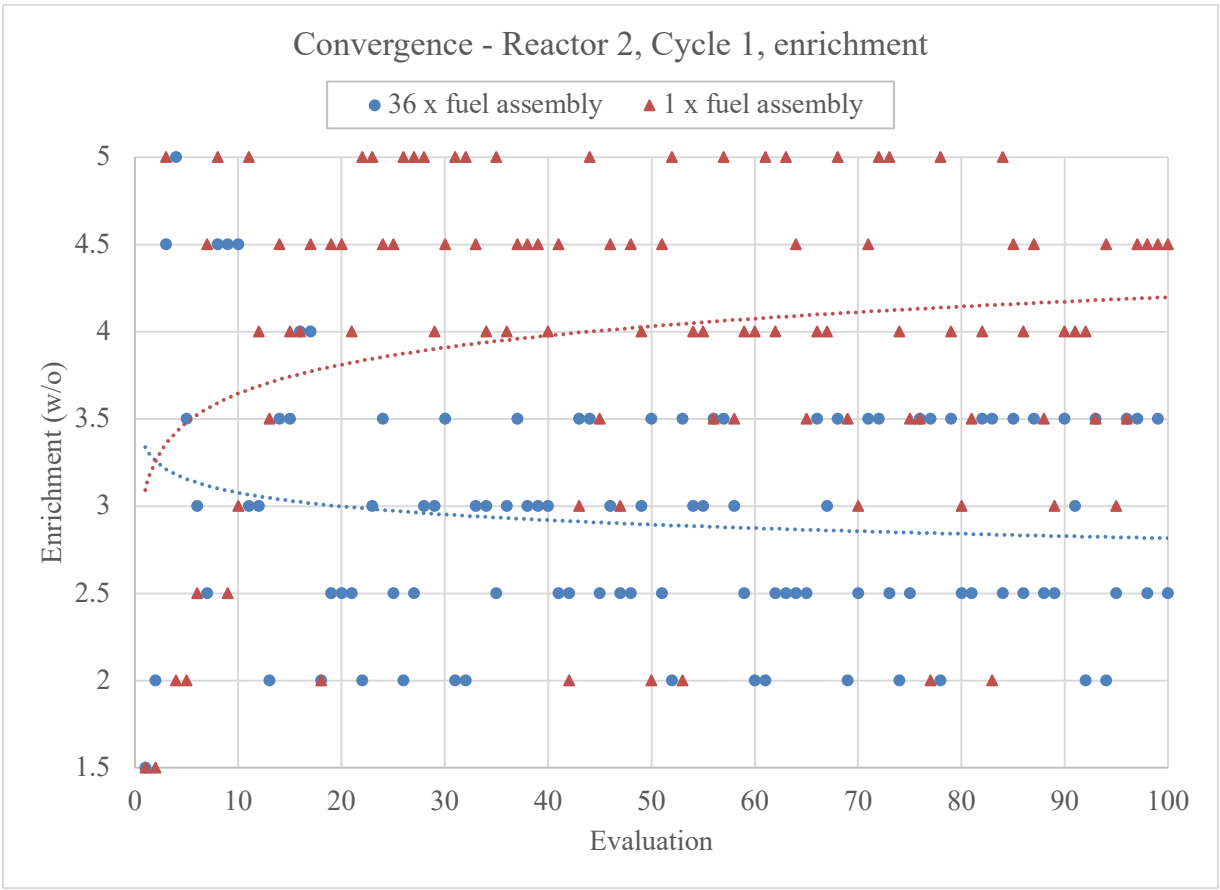

Figure 2: The two enrichment input parameters for the first cycle of reactor 2.

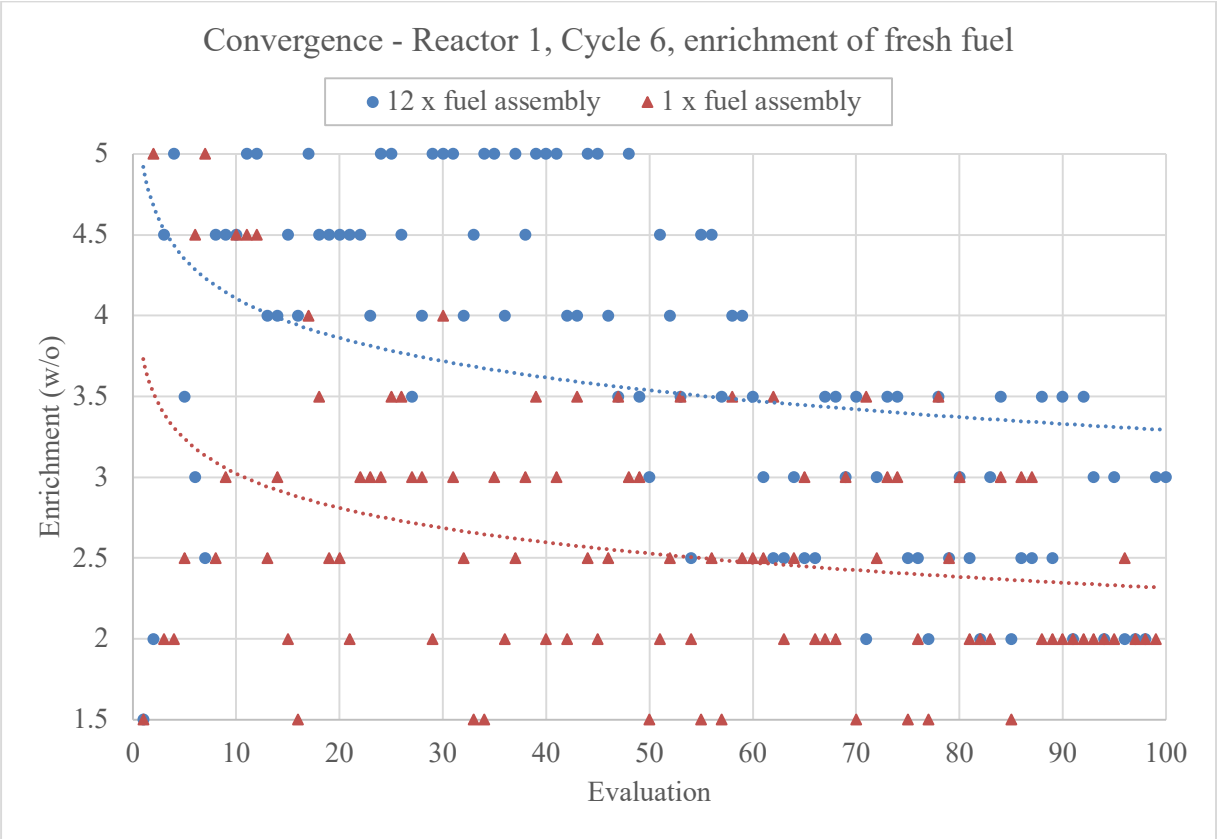

Figure 3: The two enrichment input parameters for the sixth cycle of reactor 1.

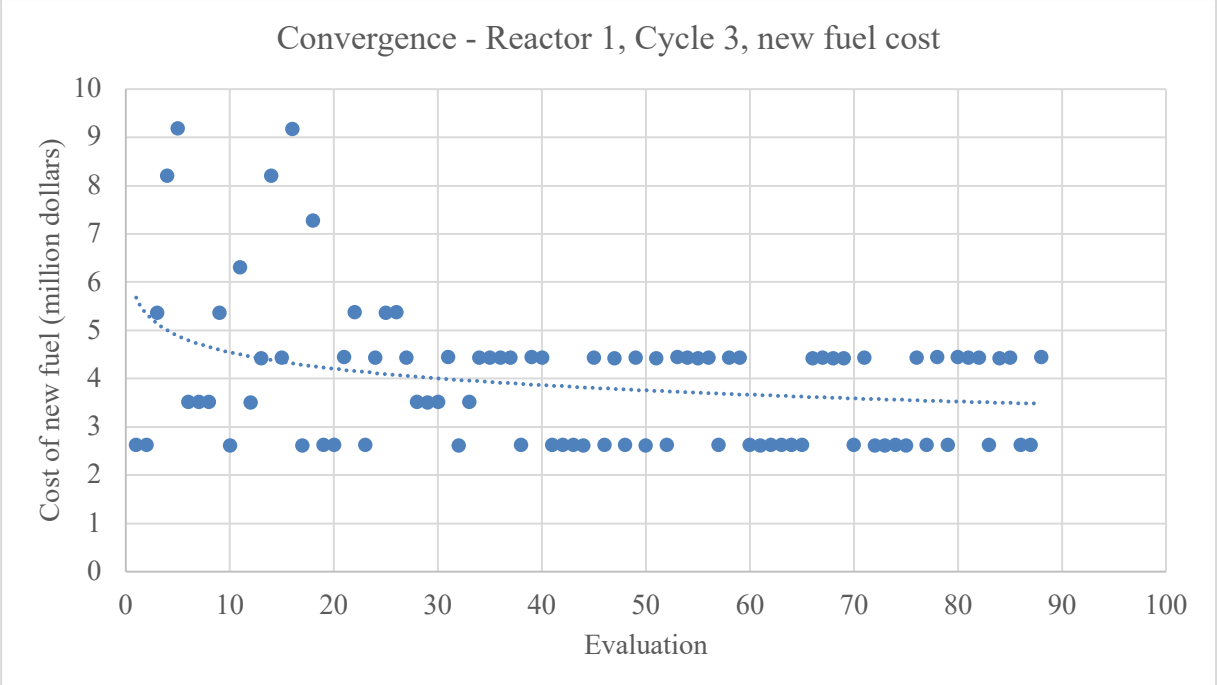

Figure 4: The convergence of the objective function for the third cycle of reactor 1.

3.2 Multi-reactor multicycle comparison

The "Independent" case in Table III is analogous to combining the operating costs and parameters of two independent reactors where each single reactor runs a fixed cycle length for many cycles. A Dakota calculation is performed for each reactor, 10 cycles for one and 5 cycles for the second, and the results combined.

The "Shared" case is a third Dakota calculation of 15 cycles in which fuel can be swapped. The out-of-core strategy in both cases is fixed with all even cycles loading 12 fresh fuel assemblies and all odd cycles loading 13 fresh fuel assemblies. (Further exploration of out-of-core strategies will be performed after the code is parallelized.) The multi-reactor shared fuel pool case outperforms the independent case with 2.4% lower fuel costs and 0.8% greater fuel utilization. As noted earlier, the shutdown cooling times are overestimated in the shared case and will likely lead to even lower costs. Future work will quantify this difference.

Table III: Results from the two strategies. "Independent" combines the results of reactors with independent fuel pools. "Shared" reports the results of two reactors with a shared fuel pool.

Reactor	Target cycle length (EFPD)	FCC ($/MWh$_e$)	Number of fuel assemblies	Average Exposure (GWd/t)
1	347	12.877	149	15.67
2	694	11.478	87	26.85
Independent	**n/a**	**12.177**	**236**	**19.79**
Shared	**n/a**	**11.884**	**236**	**19.95**
% change	n/a	-2.4	0	+0.8

4. CONCLUSION

Core design automation has been demonstrated for a multi-reactor multicycle system using a Python package interfaced with Studsvik core modeling software and Dakota, a multipurpose software tool from Sandia National Lab. A case study of a multi-reactor system with different cycle length strategies and a shared fuel pool demonstrates 2.4% savings in fuel cycle costs over the independent used fuel pool case.

The optimization problem scope will be extended to include new design variables, constraints, and objective functions. Future work will demonstrate full shuffle pattern exploration for better neutron economy, other new fuel batch sizes (e.g., 12 fuel assemblies for one cycle, then 9 assemblies in the next, etc.), and more out-of-core reload strategies. The loss of accuracy due to the shutdown cooling pseudo-dates may be avoided by using the most current version of SIMULATE, but future work will take care to quantify the impact of this approximation.

ACKNOWLEDGMENTS

The authors wish to thank the engineers in the core and fuels group at NuScale for their mentorship.

REFERENCES

1. P.J. Turinsky, "Nuclear fuel management optimization: A work in progress." *Nuclear technology,* **151.1** pp. 3-8 (2005).
2. K.E. Ottinger, G.I. Maldonado, "SMR Fuel Cycle Optimization Using LWROpt." *Journal of Nuclear Engineering and Radiation Science,* **3.1** (2017).

3. L. Encinas, A. Ortego, J.A. Bermejo, “Last cycle optimization activities at CNAT.” CMS Users Group. (2019).
4. M. Edenius, et al., "CASMO-4, a fuel assembly burnup program, user’s manual." Studsvik, (1995).
5. Studsvik Scandpower, “SIMULATE-3: Advanced Three-Dimensional Two-Group Reactor Analysis Code-User's Manual (University Release).” Studsvik, (2009).
6. B.M. Adams, et al., “Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.12 User’s Manual.” Sandia National Laboratories, (2020).
7. NuScale Power, “NuScale Standard Plant Design Certification Application, Chapter Four Reactor.” ML20036D438 Rev. 4. (2020).
8. A. Soldatov, *Design and Analysis of a Nuclear Reactor Core for Innovative Small Light Water Reactors.* Oregon State University, (2009).
9. International Atomic Energy Agency, *Reload Design and Core Management in Operating Nuclear Power Plants*. IAEA-TECDOC-1898, IAEA, Vienna (2020).
10. N. Tsoulfanidis, *The Nuclear Fuel Cycle*. American Nuclear Society, (2013).
11. T. Bahadir, “Modeling of shutdown cooling reactivity effects with SIMULATE” *Proceedings of the 2014 Physics of Reactors Conference (PHYSOR ’14)*, (2015).