

AUTOMATED FUEL MANAGEMENT OPTIMIZATION FOR FAST REACTORS

Michael Jarrett and Florent Heidet

Argonne National Laboratory
9700 S Cass Avenue, Lemont, IL 60439

mjarrett@anl.gov, fheidet@anl.gov

ABSTRACT

The Versatile Test Reactor (VTR) is expected to operate in a persistent non-equilibrium state due to inter-cycle variations in experimental loading. The goal of planning and optimizing the fuel loading for this mode of operation can differ from equilibrium cycle optimization. In this work, a general algorithm for optimizing a core reload of a fast reactor with respect to some objective function is developed. The objective function used in this work is a preliminary model that is defined to capture most of the core parameters expected to be of interest, but elements could be added or subtracted as needed for different types of problems. The optimization method is a discrete evolutionary algorithm. Instead of using diffusion or transport to evaluate each potential core configuration that is considered during the execution of the optimization method, the necessary inputs to the objective function (k -effective and assembly power distribution) are evaluated approximately by treating the reloaded configuration as a small change to the previous configuration, for which a diffusion or transport solution has already been calculated. This approximate calculation facilitates evaluation of the objective function for several hundred potential configurations without a neutron transport solution, which would be a significant bottleneck in the optimization method. In the results, the evolutionary algorithm demonstrates good responsiveness to the tuning of the parameters of the objective function.

KEYWORDS: sodium fast reactor, test reactor, fuel optimization

1. INTRODUCTION

Developing a fuel reloading pattern for a nuclear reactor is a task traditionally performed by an expert with considerable knowledge. By following some basic principles and intuition, a person can typically develop an effective reloading pattern for equilibrium operation. However, when a reactor needs to be operated outside of equilibrium cycle conditions, it can be much more difficult for a human to specify the best pattern of fuel management. Manually ensuring optimal performance with respect to some desirable operating conditions or parameters while also tracking adherence to limits on linear heat rate, fuel burnup, and excess reactivity is a daunting task. Automation is a virtual necessity to achieve optimization within these constraints.

In this work, an automated method for optimizing the fuel management of the Versatile Test Reactor (VTR) is sought. The development of capabilities required for the ultimate objective is still in progress, but this work demonstrates the proof-of-concept for most of the main components. The optimization method is a discrete evolutionary algorithm.

The VTR is a planned 300 MWt sodium-cooled fast reactor, with the mission to enable accelerated testing of advanced reactor fuels and materials required for advanced reactor technologies. [1] Since the VTR is a test reactor, the experimental loading can vary significantly between cycles, and this experimental loading is expected to have a sizeable impact on the core physics, especially when fuel or absorber materials are loaded into test assemblies. Because of this variability, the optimization method developed must work on a cycle-by-cycle basis and be capable of adapting to changing experimental loading and reactor conditions.

2. BACKGROUND

2.1 VTR Fuel Management

When planning the fuel loading of the VTR, several parameters are likely to be taken into consideration, and all are also dependent on the experimental loading to varying degrees. First and foremost, the excess reactivity at beginning-of-cycle needs to be at a level that can be safely managed by the control system, as well as large enough to maintain criticality to the end of the planned cycle length. The fast flux fluence in the test volume should generally be maximized, especially in the central test assembly location. The assembly power distribution should be well-matched with coolant inlet orifice distribution so that there are no large outlet coolant temperature differences between assemblies, which could lead to undesirable thermal “striping” (i.e., cycling) of the upper core structural materials. Additionally, the required number of fresh fuel assemblies at each fuel reloading should be minimized to reduce the demands placed on both the fuel manufacturing facility and the spent fuel handling and storage areas.

2.2 Fuel Loading Optimization Methods

Optimization of nuclear reactor fuel loading has been explored thoroughly. Early on, deterministic approaches such as linear [2], quadratic [3], dynamic [4], and approximation [5] programming were used. These have largely been supplanted by heuristic methods such as simulated annealing [6], genetic algorithms [7-9], and others [10]. In many instances there are particular details unique to the specific problem at hand that indicate different approaches, methods, and algorithms to achieve the desired outcome. Variables to be optimized can include power distribution [7], fuel burnup [9], required enrichment [6], conversion ratio [8], and more. There are many parameters that can be varied to optimize these outputs, including fuel enrichment, fuel dimensions, fuel loading location and/or shuffling pattern, axially or radially heterogeneous enrichment zoning within fuel assemblies, and total core size. The size of the search space for both input and output variables can be staggeringly large; for this reason, many of these variables are often chosen to be fixed, and the optimization focuses on a small number of input variables, targeting optimization of only a few outputs.

In response to the robust diversity in problem specifications, nuclear scientists have tapped into a wide range of approaches and methods to handle fuel optimization. Because of the diversity, it is difficult to achieve unification of the available methods in a single code or toolset. DAKOTA [11] is perhaps the most generalized tool currently used for nuclear fuel optimization [8]; it offers several gradient and nongradient-based methods for optimization. Despite the many capabilities of DAKOTA, many opt for the flexibility of developing their own methods and codes for their specific applications [9,10]. Using in-house code focused on the target problem enables more tuning and control over fine-grain details, which can ultimately improve code capabilities and functionality.

In this work, an algorithm is developed that is very strongly influenced by the concept of “Evolutionary” or “Genetic” optimization algorithms. [12] This algorithm is deployed to maximize an objective function

defined specifically for the VTR; it depends on the radial power peaking factor, estimated core reactivity, number of fresh fuel assemblies loaded, and average discharge burnup.

2.3 Objective Function Evaluation

The most expensive part of optimization is often the evaluation of the objective function. For fuel loading optimization, the quantities of interest (reactivity, power distribution) would in theory require a neutron transport solution for each potential configuration. Iterative optimization can involve evaluation of hundreds or thousands of core configurations to achieve convergence. Even with a relatively inexpensive method such as diffusion, the time and computing resources required for this are significant and would likely be deterrent. For this work, an approximate evaluation of the reactivity and power distribution [13] is used to obtain an approximate evaluation of the objective function for an alternate configuration given a base-case diffusion solution from DIF3D-VARIANT [14]. This method can approximately evaluate the fitness of dozens of core configurations in less than 1 second, while each multigroup diffusion solution in DIF3D-VARIANT requires on the order of 1 minute.

3. OPTIMIZATION METHOD

3.1. Objective Function Definition

The function to be optimized is among the most important features of an optimization algorithm. Four input variables are targeted for optimization in this work, although there are others that could just as well be included, depending on the desired output. The four input variables are:

1. Assembly radial power peaking factor (w)
2. Proximity of estimated core k-effective to the targeted k-effective (x)
3. The average discharge burnup (in % FIMA) of assemblies removed from the core (y)
4. The number z and type i of fresh fuel assemblies loaded into the core (z_i)

The value of the objective function, or the fitness, F , is defined in Eq. (1):

$$F = \frac{A}{(w-1)^2} + Bg(x, \sigma_x) + Cy - h(\mathbf{z}). \quad (1)$$

The order and form of the contribution for each input variable (linear, Gaussian, quadratic, etc.) is chosen based on engineering judgment. The particular choices that have been made here are not of material importance. There is a significant amount of flexibility in defining the fitness function in whatever way one may desire.

The coefficients A , B , and C are defined by the user in the input; these effectively set the relative importance of each associated variable. The excess reactivity fitness is a modified Gaussian distribution with a plateau in the middle; this effectively sets a target range instead of a specific target value. It is defined in Eq. (2):

$$g(x, \sigma_x) = \begin{cases} 1 & |x| - x_0 < \Delta x_0 \\ e^{-\left(\frac{|x|-x_0}{\sigma_x}\right)^2} & \text{otherwise} \end{cases}. \quad (2)$$

x is the difference between the target k-effective and the estimated k-effective, x_0 is a horizontal offset, and Δx_0 is the width of the plateau in the middle. These parameters can be tuned to improve performance or achieve a desired effect; for this work they are defined from the standard deviation in Eq. (3a) and Eq. (3b):

$$\Delta x_0 = \frac{\sigma_x}{2}, \quad (3a)$$

$$x_0 = -\Delta x_0. \quad (3b)$$

Although there is not yet any plan to use multiple driver fuel types in the VTR core, the optimization cases are much more interesting if there are at least a few discrete choices for fuel type. The 3 fuel types used in this work are the reference fuel design, a version with 110% of the reference fuel volume fraction, and one with 95% of the reference volume fraction. The fuel volume fractions are 0.358, 0.377, and 0.339, respectively. The fuel volume fraction is defined as the proportion of the cross-sectional area of the fuel assembly that contains fuel. All three fuel types use the same ternary metal fuel, U-20Pu-10Zr, with 72% fissile (Pu-239 or Pu-241) plutonium. Each fuel type i is assigned a value h_i for the function $h(\mathbf{z})$, which effects a preference for fuel assembly types with respect to one another. The function is a simple inner product of the importance of the type, h_i , with the number of that type loaded, z_i , shown in Eq. (4):

$$h(\mathbf{z}) = h_1 z_1 + \dots + h_n z_n. \quad (4)$$

Since the goal is to maximize the fitness, and $h(\mathbf{z})$ is subtracted from the fitness, fuel types associated with a higher h_i are less likely to be selected to be loaded in a given location.

3.2 Evolutionary Algorithm

The optimization algorithm is essentially an evolutionary algorithm applied to optimize an objective function with discrete input variables. The method is described in Algorithm 1.

An “individual” $P_{l,n}$ in a generation l is a single fuel reloading specification, which consists of a list of what operation is performed in each fuel location: type of fresh fuel loaded, or no change. A generation consists of many individuals (approximately 30) with some variation in characteristics. The fitness $F_{l,n}$ of each individual is evaluated quickly using the approximate evaluation method, and the individuals are sorted by fitness. The top individuals (approximately 3-5) from a generation are chosen as “survivors” that will be transferred to the next generation. The rest of the generation is populated by using some combination of the traits of the survivors. The inheritance is weighted in favor of survivors with higher fitness. Two significant differences between this algorithm and a more standard genetic algorithm [7,9] are that offspring are produced from multiple parents and mutations occur only in the initial generation. These features may be modified at a later time if it is found that the performance of the algorithm is not satisfactory.

This algorithm is implemented in a python code named OTERR (Optimization of TEst Reactor Reloading). The development is currently focused on application to the VTR, but the underlying assumptions and principles upon which it operates should make the code applicable to most fast reactors. Because of fundamental differences in core physics, it is unlikely that the same methods would function correctly for thermal spectrum reactors.

Algorithm 1 Discrete Evolutionary Algorithm for Maximizing Core Fitness

```

1: Begin with initial generation  $l = 0$ , with a population size  $N_0$ .
2: Generate an initial fuel reloading  $P_{0,1}$  using some deterministic algorithm.
3: for  $n \in [2, N_0]$  do
4:   Produce a copy  $P_{0,n}$  of the initial reloading  $P_{0,1}$  with the fuel types and locations
   randomly varied where possible (mass mutation).
5: end for
6: for  $l \in [0, L]$  do
7:   for  $n \in [1, N_l]$  do
8:     Evaluate core fitness  $F_{l,n}$  of the individual core reloading  $P_{l,n}$  using Eq. (1).
9:   end for
10:  Sort the individuals by fitness  $F_{l,n}$ , removing duplicates, and select the top  $M_l$ 
  for survival to the next generation.
11:  Vector of surviving individuals:  $\mathbf{S}_l = [s_{l,1} \dots s_{l,M_l}]$ 
12:  if ( $l < L$ ) then
13:    for  $n \in [M_l + 1, N_{l+1}]$  do
14:      Create individual “offspring”  $P_{l+1,n}$  by selecting traits from survivors  $\mathbf{S}_l$  of
      previous generation. Traits are selected by sampling a specified “influence”
      CDF:  $\mathbf{I}_l = [I_{l,1} \dots I_{l,M_l}]$ .
15:      Mutate individual traits of  $P_{l+1,n}$  at specified mutation rate  $r_l$ .
16:    end for
17:  else
18:    Final generation  $L$ . The highest-fitness individual  $P_{L,n}$  is chosen as the next
    core reloading.
19:  end if
20: end for

```

4. RESULTS

The expectation for the evolutionary algorithm is that core parameters of interest, such as excess reactivity, radial assembly power peaking, or number of fresh fuel assemblies used, can be influenced to a significant degree by the chosen user input values defining the objective function. Additionally, it is desired that an optimal solution is found for a given objective function. Attaining the optimal solution with respect to the exact objective function when using an approximate evaluation of the objective function requires that the approximate evaluation be sufficiently accurate. The results in the present work serve as demonstration of the ability to influence core parameters. Evaluating the accuracy of the approximate objective function and determining whether the global optimum solution is obtained is not part of the present work, although it will eventually be an important component of the overall method development.

The results presented here are average values obtained over a relatively long multicycle simulation under a fixed core objective function and set of constraints. Starting from a fresh core, the multicycle reloading algorithm is run for 40 consecutive cycles, each one lasting 100 days at full power. The first four cycles are thrown away as the core is converging to an approximate equilibrium state; core performance data is accumulated for the 5th through 40th cycles and averaged. REBUS [15] is used for the depletion calculations.

4.1 Peaking Factor

The influence of the peaking importance factor (A) on average beginning-of-cycle (BOC) radial peaking is shown in Fig. 1. The error bars indicate the standard deviation of the peaking factor. The average peaking factor decreases as the importance of peaking reduction increases, but it is only reduced by 6% even when the power peaking importance factor is relatively large. This shows that the objective function has influence over the core loading, but the degree to which peaking factor can be reduced through core loading is limited. In each case, the cycle-to-cycle variation in BOC peaking factor is much wider than the separation of the averages.

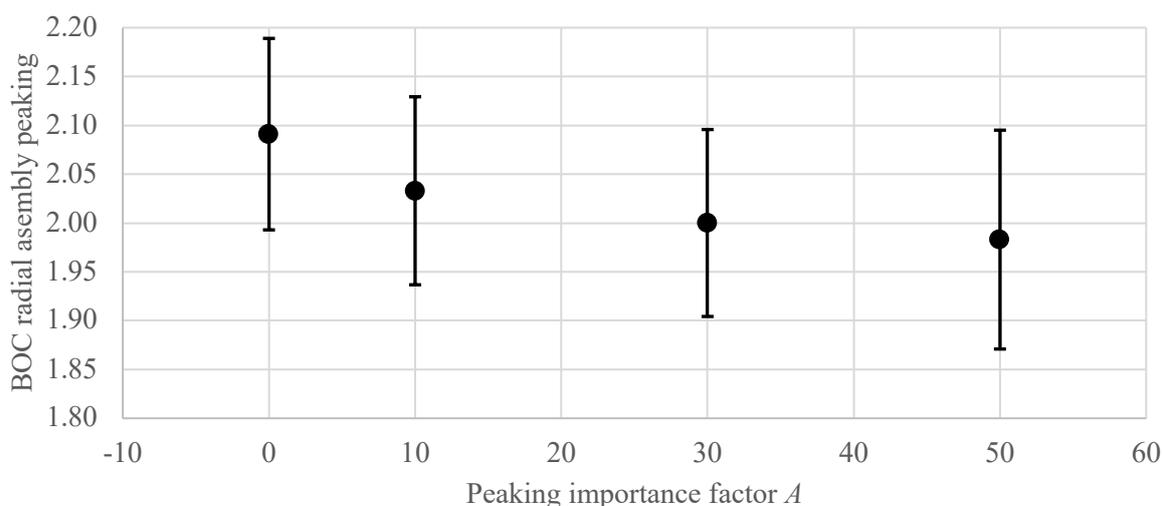


Figure 1. Average core radial peaking factor at BOC

4.2 Excess Reactivity

The excess reactivity score is defined by a coefficient B and a standard deviation σ_x . The effect of these two parameters is shown in Fig. 2. The standard deviation does not have a consistent effect on the proximity to the target k-effective achieved, but the coefficient B does. The average distance from the target k-effective decreases slightly as B is increased. This is the distance between the target k-effective and the *estimated* k-effective of the new core, not the actual k-effective calculated with DIF3D.

Adherence to the target reactivity can be improved by increasing the importance coefficient, but it can also be improved by modifying the function. The standard deviation of the Gaussian σ_x , the offset x_0 , and the width of the plateau Δx_0 all may affect how well the algorithm converges to the target k-effective.

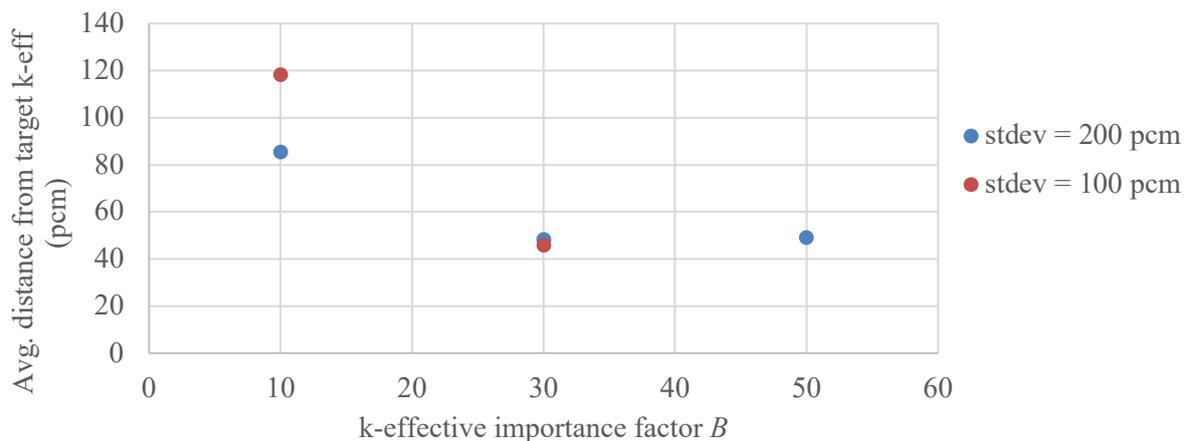


Figure 2. Average distance from target k-effective at BOC

4.3 Assembly Type Costs

The effect of assigning unbalanced costs to the fuel assembly types is shown in Fig. 3. In each case, two of the assembly types have a low cost ($h = 1.0$) and one has a high cost ($h = 10.0$). When an assembly has a high cost, that fuel type is used much less frequently. This shows that the algorithm is strongly responsive to the fuel type component of the objective function. The expensive fuel type is used only when the other types are unavailable because of the thermal hydraulic constraints. In practice, each fuel type can be assigned a cost based on the cost of manufacture, required fuel enrichment and mass, and the algorithm can find the most cost-effective fuel loading.

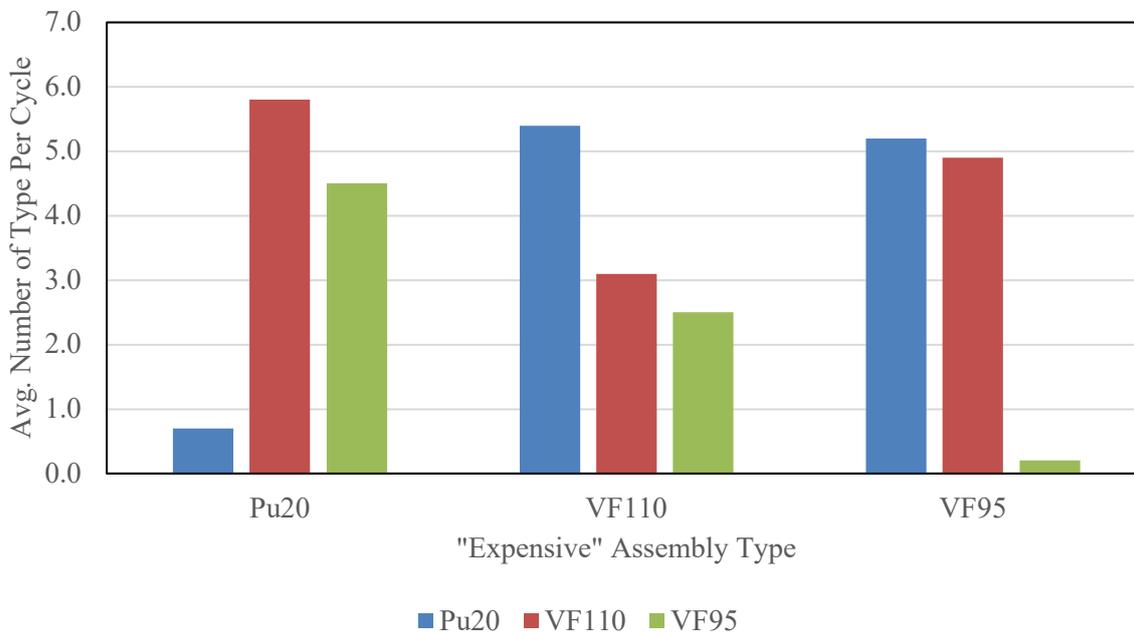


Figure 3. Average number of each assembly reloaded

5. CONCLUSIONS

A discrete evolutionary algorithm was developed for optimizing the fuel reloading of fast reactors with respect to an objective function that can be approximately evaluated dozens of times per second. Core parameters such as the assembly power distribution and k-effective are evaluated approximately by calculating the effects of relatively small changes to the previous configuration, for which there is an available diffusion or transport solution. This approximate evaluation keeps the overall optimization time down, so that it requires less than one minute to arrive at an optimized fuel reloading specification.

The algorithm developed for this work is capable of responding to the objective function and improving upon the initial configuration fitness significantly. There is a strong response to fuel assembly values $h(z)$, but only a modest response to the power peaking factor A and reactivity factor B . The average discharge burnup was only marginally affected by the factor C , but the results are not shown here.

With intelligent selection of iteration parameters such as generation size, number of survivors, influence, and number of generations, this algorithm is very likely to find a local maximum. If the initial population is not diverse enough, it is possible that the global maximum within the problem boundaries will not be found. Introduction of mutation and informed adjustment of algorithm parameters can help avoid being trapped in local extrema and improve convergence to global extrema, although there is no guarantee that this algorithm will produce a global optimum. While the global optimum would be the preferred solution, it is not necessarily the only acceptable solution in this application. When determining a suitable fuel loading pattern, any solution that meets the requirements and is locally optimized with respect to the defined preferences is satisfactory.

The specific features of the objective function depend on the coefficients chosen by the user, and these features were not studied in this work. Understanding the form and features of the objective function and applying this knowledge to tailor the optimization method is a topic for future work that could be of significant benefit.

ACKNOWLEDGMENTS

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne National Laboratory’s work was supported by the U.S. Department of Energy, Office of Nuclear Energy under contract DE-AC02-06CH11357. The work reported in this summary is the results of R&D studies supporting a VTR concept, cost, and schedule estimate for DOE-NE to make a decision on procurement in the future. As such, it is pre-decisional.

REFERENCES

1. G. Youninou, F. Heidet, A. Abou Jaoude, S. Bays, T. Fei, G. Palmiotti, and M. Smith, March 2018 VTR Reference Core Concept, ECAR-4072, Project 32833, April 2018.
2. T.O. Saouar, “Application of Linear Programming to In-Core Fuel management Optimization in Light Water Reactors,” *Nucl. Sci. Eng.*, **46**, 274 (1971).
3. D. Tabak, “Optimization of Nuclear Reactor Fuel Recycle via Linear and Quadratic Programming,” *IEEE Trans. Nucl. Sci.*, **NS-15**, 60 (1968).
4. I. Wall and H. Fenech, “The Application of Dynamic Programming to Fuel Management Optimization,” *Nucl. Sci. Eng.*, **22**, 285 (1965).
5. H. Motoda, “Optimization of Control Rod Programming and Loading pattern in Multiregion Nuclear Reactor by the Method of Approximation Programming,” *Nucl. Sci. Eng.*, **49**, 515 (1972).
6. G.T. Parks, “An Intelligent Stochastic Optimization Routine for Nuclear Fuel Cycle Design,” *Nuclear Technology*, **89**(2), 233-246 (1990).

7. F. Alim, K. Ivanov, and S.H. Levine, “New genetic algorithms (GA) to optimize PWR reactors Part I: Loading pattern and burnable poison placement optimization techniques for PWRs,” *Annals of Nuclear Energy*, **35**(1), 93-112 (2008).
8. J. Washington and J. King, “Optimization of plutonium and minor actinide transmutation in an AP1000 fuel assembly via a genetic search algorithm,” *Nuclear Engineering and Design*, **311**, 199-212, (2017).
9. A.V. Sobolev, A.S. Gazetdinov, and D.S. Samokhin, “Genetic algorithms for nuclear reactor fuel load and reload optimization problems,” *Nuclear Energy and Technology*, **3**, 231-235 (2017).
10. S. Liu, J. Cai, “Studies of fuel loading pattern optimization for a typical pressurized water reactor (PWR) using improved pivot particle swarm method,” *Annals of Nuclear Energy*, **50**, 117-125, (2012).
11. B.M. Adams et al., “Dakota: A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis Version 5.4 Theory Manual,” Sandia National Laboratories (2013).
12. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Boston, MA, USA (1989).
13. M. Jarrett and F. Heidet, “Fuel Loading Optimization and Planning for the Versatile Test Reactor,”
14. G. Palmiotti et al, “Variational nodal transport methods with anisotropic scattering,” *Nuclear Science and Engineering*, **115**, 233-243 (1993).
15. B. J. Toppel, “A User’s Guide to the REBUS-3 Fuel Cycle Analysis Capability,” ANL- 83-2, Argonne National Laboratory (1983).