12-2016

Algorithms and Methods for Optimizing the Spent Nuclear Fuel Allocation Strategy

Gordon Matthew Petersen
University of Tennessee, Knoxville, gpeters9@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Gordon Matthew Petersen entitled "Algorithms and Methods for Optimizing the Spent Nuclear Fuel Allocation Strategy." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

Steven E. Skutnik, Major Professor

We have read this dissertation and recommend its acceptance:

G. Ivan Maldonado, James Ostrowski, Xueping Li

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Algorithms and Methods for Optimizing the Spent Nuclear Fuel Allocation Strategy

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Gordon Matthew Petersen

December 2016

Acknowledgements

This work would not have been possible without the support and encouragement from the people around me. I would like to thank Dr. Steven Skutnik for accepting me as a student and providing a steady stream help and encouragement throughout the entire process. Whenever I came away from a meeting with Dr. Skutnik, I felt rejuvenated and excited about the work I was doing. I would like to thank the rest of my committee members, Dr. Ivan Maldonado, Dr. James Ostrowski, and Dr. Xueping Li for providing thoughtful insight on my dissertation topic.

Additionally, I would like to thank Robby Joseph for mentoring me through my undergraduate and graduate work. He provided thoughtful analysis, important professional information, and was always available to mentor me through an important point in my professional development. I would like to thank Josh Jarrell, Riley Cumberland, Rob Howard, and the rest of the Used Fuel System Group at Oak Ridge National Laboratory for providing the information and freedom to work through my graduate studies.

I would also like to thank my friends and fiancée, Betsy, for being there for me through these intense years. Finally, I am eternally grateful to my parents for instilling morals and the value of hard work in me throughout my childhood.

Abstract

Commercial nuclear power plants produce long-lasting nuclear waste, primarily in the form of spent nuclear fuel (SNF) assemblies. Spent fuel pools (SFP) and canisters or casks that sit at an independent spent fuel storage installation (ISFSI) at the reactor site store the fuel assemblies that are removed from operating reactors. The federal government has developed a plan to move the SNF from reactor sites to a Consolidated Interim Storage Facility (CISF) or a geological repository. In order to develop a predictable pick-up schedule and give utilities notice of an impending pickup from a reactor site, the federal government developed a queuing strategy based on the first-in-first-out algorithm, known as oldest fuel first (OFF). The OFF algorithm allows the federal government to remove SNF from reactor sites in the same order the assemblies came out of the reactor. While an OFF allocation strategy may result in a fair approach, it is far from the most cost-effective approach.

The problem with accepting SNF using an OFF algorithm is that a handful of sites are no longer producing power and exist only to store the SNF they produced. This is an expensive process, which results in an annual cost of ~$8M [22]. Utilizing different algorithms to reduce the amount of time these shutdown reactors keep SNF on site may reduce the total system costs for the federal government.

A greedy algorithm, genetic mutation algorithm, simulated annealing algorithm, and an integer programming formulation were all developed to reduce the number of years that reactors were shut down with SNF on site.

Table of Contents

List of Tables

List of Figures

Chapter One

Introduction and General Information

Commercial nuclear power plants produce long-lasting nuclear waste, primarily in the form of spent nuclear fuel (SNF) assemblies. Spent fuel pools (SFP) and canisters or casks that sit at an independent spent fuel storage installation (ISFSI) at the reactor site store the fuel assemblies that are removed from operating reactors. The federal government has developed a plan to move the SNF from reactor sites to a Consolidated Interim Storage Facility (CISF) or a geological repository. The federal government has a contract with the utilities to remove fuel from reactor sites using an Oldest Fuel First (OFF) or first-in-first-out approach. The government was to begin removing SNF from reactor sites in 1998 [1]. Since the federal government has been unable to begin moving SNF, the expected amount of SNF at reactor sites has vastly increased. Given the present situation of SNF management in the U.S., developing a more optimized allocation strategy has potential for significant cost savings.

1.1 Spent Nuclear Fuel in the United States

Nuclear power production has the highest energy density compared to all competing sources of energy. However, it produces a waste product that will be around for hundreds of thousands of years. The Nuclear Waste Policy Act (NWPA) of 1982 specified that DOE would begin to take

possession of SNF from private utility companies beginning no later than January 31, 1998 [1]. The NWPA established the Nuclear Waste Fund, funded by a tax on nuclear-generated electricity, and paid for by utilities in order to fund future disposal of SNF. Consequently, the Office of Standard Contract Management was created to be responsible for interactions relating to the litigation and settlements under the Standard Contacts with the nuclear industry and the management of Nuclear Waste Fund activities [30]. By mid-2013 the utilities had contributed over $28B to the fund (including accrued interest), but there is still no solution to storing SNF away from reactor sites [4].

In addition to the Nuclear Waste Fund, the Standard Contract specifies the default order or allocation strategy that the Federal government will remove SNF from reactor sites using an OFF allocation strategy [2]. The OFF allocation strategy gives reactors priority based on the date of discharge of spent fuel assemblies from the commercial nuclear power reactor. The contract also gives the government the ability to prioritize SNF from reactors that have reached the end of their useful life or shutdown permanently [2].

Congress clarified the NWPA in 1987 passing an amendment specifying that the Yucca Mountain site in Nevada would serve as the nation’s sole geological repository. After much political debate and no SNF removal from the majority of reactor sites, the Obama Administration proposed defunding the Yucca Mountain project in the FY2010 budget and subsequently established the Blue Ribbon Commission (BRC) on America’s Nuclear Energy. This organization was tasked to develop a new SNF disposal and management policy. In 2012, the BRC recommended eight main points including a consent-based siting approach for an interim storage facility and a final geological repository [3].

1. Consent-based siting
2. New organization to implement waste management program
3. Access to funds from Nuclear Waste Fund
4. Prompt efforts to develop geologic disposal facility
5. Prompt efforts to develop consolidated storage facilities
6. Prompt efforts to prepare for large-scale transport
7. Support for continued U.S. innovation
8. Active U.S. leadership in international efforts to address SNF

The Department of Energy responded by recommending the following program [26]:

- "Sites, designs and licenses, constructs and begins operations of a pilot interim storage facility by 2021 with an initial focus on accepting used nuclear fuel from shut-down reactor sites;
- Advances toward the siting and licensing of a larger interim storage facility to be available by 2025 that will have sufficient capacity to provide flexibility in the waste management system and allows for acceptance of enough used nuclear fuel to reduce expected government liabilities; and
- Makes demonstrable progress on the siting and characterization of repository sites to facilitate the availability of a geologic repository by 2048" [26].

As of 2016, DOE has yet to remove any SNF from reactor sites, prompting utilities to file a number of lawsuits against the federal government for failing to meet their obligation under the Standard Contract. Currently around 72,000 MTHM (Metric Tons Heavy Metal) of SNF have been produced from commercial nuclear power [4]. Based on the current reactor lifetimes, the total estimated amount of SNF produced will be close to 140,000 MTHM. At the end of 2013,

there had already been thirty-three lawsuits settled for $2.7B and twenty-six final judgments had been awarded $0.99B [4]. If the federal government utilizes an OFF allocation strategy and does not start removing SNF until 2021, the estimated future government liability for breach of contract will be $23.7B, raising the total to just over $26B [4]. The money that is awarded to utilities is not provided out of Nuclear Waste Fund but from the Judgment Fund coming directly from the taxpayers, because the federal government pays lawsuits from the Judgment Fund.

1.2 SNF Storage in the United States

The utilities developed storage options surpassing their initial design in order to continue operating the nuclear power plants. SNF is currently stored either in spent fuel pools or in dry cask storage as shown in Figure 1.

Figure 1: On the left is a spent fuel pool filled with spent fuel assembly racks [5]. On the right are vertical storage overpacks storing dry canisters [6]

Initially utilities built spent fuel pools to handle a couple offloads from the reactor under the assumption the SNF would be picked up by the federal government before these pools were close to capacity [5]. Since the federal government failed to remove the fuel, spent fuel pools began to near maximum capacity. The pressure to keep the reactor online forced utilities to provide a solution to their storage problem, thus developing a dry storage alternative.

The spent fuel pool provides shielding from radiation, acts as a heat sink for the SNF, and maintains geometry and spacing to stay well below criticality limits [6]. A dry storage system must also address these concerns. The most challenging problem for a dry storage system to address consists of removing a large amount of heat from the SNF. For this reason, the SNF is generally stored in a spent fuel pool for at least five years before dry storage is an option. After five years, the decay heat produced by the SNF significantly declines, allowing a dry storage system to remove sufficient heat to prevent zirconium hydriding, which can occur in the presence of very high cladding temperatures and steam [6].

Dry storage casks or canisters can be oriented and stored vertically or horizontally. The dry storage casks are generally an all-in-one storage solution, while a canister utilizes a storage overpack (Figure 2). An overpack is a protective concrete device that encases the canister. A canister may have a storage overpack, transportation overpack, and disposal overpack during its lifetime. The canisters or casks that are being used to store SNF must be removed from the sites in order to repurpose the reactor after it is decommissioned. The most popular loading technique for current utilities utilizes a Dual Purpose Canister (DPC). This particular canister is licensed for both storage and transportation alleviating the need to repackage the SNF into a canister that is suitable for transportation once the removal process begins. Loading canisters and moving them to an ISFSI is a time consuming process.

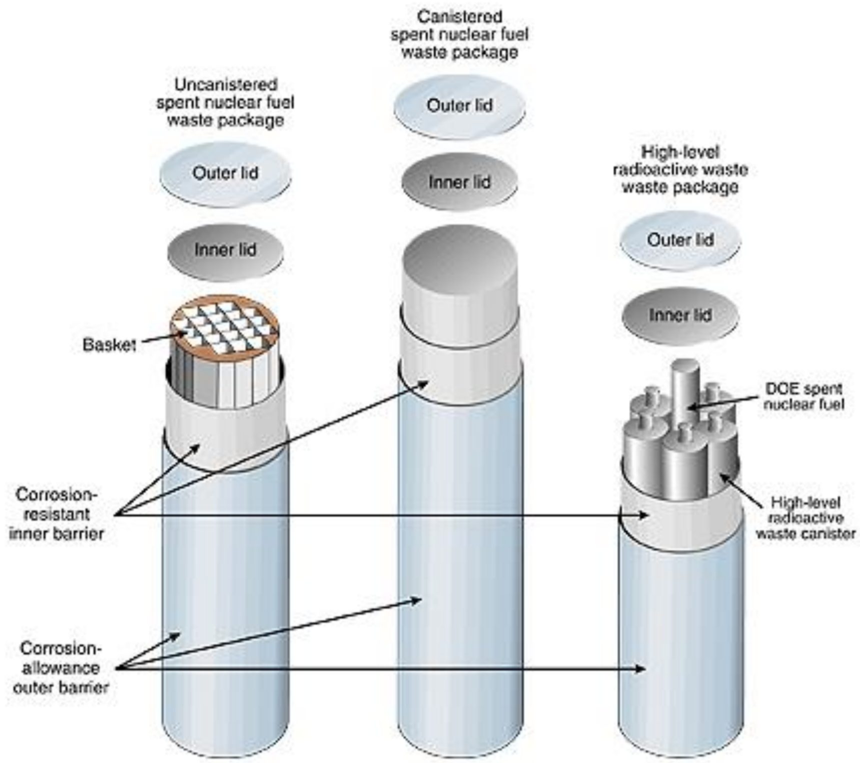

Figure 2: The left image shows a cask system and the center system is a canister-based system. The cask system is generally loaded all in one and can be stored as is, but the canister system must be packaged in an outer barrier or overpack [30]

The operations often occur in the same space and use the same equipment meaning the vast majority of steps much be completed sequentially and canisters cannot be loaded in parallel.

The high-level steps to load a canister are performed as follows [74]:

1. Preparation of a canister for fuel loading
2. Insert canister into transfer cask
3. Place canister and transfer cask into fuel pool
4. Load fuel into canister (17-29 hours) [74]
5. Remove the loaded canister/transfer cask from the fuel pool
6. Decontaminate cask exterior
7. Drain small amount of water from canister/cask cavity, then weld and inspect inner lid (vacuum or forced helium drying system) (12-48 hours) [75]
8. Install canister outer closure plate (9 hours) [74]
9. Transfer canister from transfer to storage cask (15 hours) [74]
10. Store Cask

Each of these operations can be broken down further in the Appendix, "Node Descriptions". The entire time spent loading a canister is generally a week. Loading a number of canisters can be difficult for operating reactors, since their first priority is to produce power. Shutdown reactors may be able to load more canisters in a particular year given the constraints on equipment and the lack of competing priorities.

As previously stated, the Standard Contract specifies an OFF allocation strategy, which only makes use of the fuel discharge date. The first reactor that discharged fuel will be the first reactor the federal government will remove SNF from using a first-in-first-out (FIFO) algorithm. This

inherently generates a queuing order based on discharge date requiring the allocation strategy to pick up SNF from many different reactors that also discharged fuel before it can return to pick up SNF from the first reactor [2]. Once a canister is removed from the reactor, it will be taken to a CISF or to a geologic repository. It should be stressed that the allocation strategy itself is independent of the SNF destination, be it an interim storage facility or a geologic repository. Assuming a constant acceptance rate, the allocation strategy is unaffected by destination of the SNF packages. Rather, the allocation strategy itself determines the rate at which reactor sites can be cleared, and thus determines the potential outstanding federal liabilities under the Standard Contract.

1.3 Past Evaluations of Allocation Strategy

Previous analyses have evaluated aspects of the allocation strategy. Three works comprise a majority of the understanding of varying the allocation strategy:

1. Spent Fuel Receipt Scenario Study by Ballou, Montan, and Revelli [8]
2. A Proposed Acceptance Queue for Shutdown Nuclear Power Reactors by Nesbit and Nichols [7]
3. Waste Management System Architecture Evaluations by Nutt, Trail, Cotton, Howard, and van den Akker [27]

Ballou, Montan, and Revelli evaluated a total of about 100 possible schedule scenarios that were variations of OFF (categorizes as First in First Out: FIFO), “Last In – First Out” (LIFO), or a combination of the two in 1990 to enhance the post-closure performance of the waste packages and the engineered barrier system. The enhancement is derived from the judgment that the

assurance of integrity of the waste packages can be improved if the borehole (drift) walls can be maintained at a temperature in excess of the local unconfined boiling point at the repository elevation for three hundred years [8]. Ballou et al. assumed an acceptance rate based on available inventory, a waste package configuration, and a repository design. They decided to use the integrated energy contained in each year's fuel receipts as the "optimization parameter," because it most closely relates to the long-term effectiveness of the repository as it looks at the energy deposited in the host rock per unit area. Ballou et al. used a list of scenarios to "optimize" for the integrated energy instead of attempting to find the optimal allocation strategy by perturbing the system or creating an algorithm. In Ballou et al.'s optimization strategy, a user is responsible for giving the code inputs corresponding to the order SNF is removed from reactor sites and subsequently placed in the repository. The code does not change these inputs and is not capable of determining a better strategy. The user must analyze the output and decide what could be done to better the results. In this way, an optimized strategy can be found. The results from this study focused on the feasibility to accept and place SNF in a way that would distribute the energy output in the repository to balance the temperature distributions within the disposal panels based on the tailored characteristics age, burnup, and spent fuel type (PWR or BWR) [8].

Nesbit proposes a waste acceptance queue for shutdown nuclear reactors in his 2015 paper [7]. He cites a longest shutdown plant first (LSPF), OFF, closest plant first, ease of site access, least fuel first, on site storage mode, and a shutdown vs. decommissioned sites approach as the possible options. He asserts that having a designated acceptance strategy will ensure an orderly and predictable removal of SNF from shutdown sites, provide the most cost-efficient removal of SNF, and remove SNF in an equitable way. Nesbit makes no assumptions concerning canister

size or type and assumes an acceptance rate much like Ballou et al., assuming strategies to remove SNF will make an easily identifiable acceptance queue based on one parameter [7].

The 2014 Waste Management System Architecture Evaluation (WMSA) considers sixteen different SNF allocation scenarios [27]. The evaluation used a combination of an acceptance rate (3,000 MTHM, 4,000 MTHM, and a variable rate) and an acceptance priority or allocation priority (OFF, and four other approaches using a site-specific allocation). The analysis attempted to ship the youngest possible SNF from a reactor site to adequately model the assumption that reactors will attempt to get rid of the SNF with the highest thermal load first. The amount of SNF delivered to a CISF from a reactor site may be less than what is allocated to the reactor if no canisters may ship due to thermal constraints.

The four site-specific allocations employed in the evaluation are as follows [27]

1. Site-specific allocation giving priority to current shutdown reactor sites, reducing the transfer of SNF from the pools to onsite dry storage (thereby reducing costs from additional dry storage modules), and removing all SNF from remaining shutdown sites in the order of license expiration date as soon as possible while maintaining the overall allocation/acceptance rate at 3,000 or 4,500 MTHM/yr.
2. Site-specific allocation giving priority to current shutdown sites while only accepting SNF from sites after shutdown with the overall allocation/acceptance rate at 3,000 MTHM/yr or 4,500 MTHM/yr.
3. Site-specific allocation giving priority to current shutdown sites and clearing remaining sites of SNF 5 years after last reactor at a site ceases operation while maintaining a steady acceptance rate of 4,500 MTHM/yr.

4. Site-specific allocation giving priority to current shutdown sites, eliminating additional transfer of SNF from pools to on-site dry storage once acceptance begins, and clearing remaining sites for multi-reactor sites five years after the last reactor at a site ceases operation over a ten-year period (from five years before to five years after the last reactor at a site ceases operation).

These allocation strategies all assume an allocation strategy prioritizing "orphan" sites that have already shutdown by the time a pilot CISF begins accepting SNF [27]. In order to be eligible to be an orphan site, no reactor on site can be operating. If a site has three shutdown reactors and a fourth operating reactor, it is not considered an orphan site. In this scenario, a pilot interim storage facility operates specifically to remove the SNF from the orphan sites before beginning to accept SNF from all sites with a specified allocation strategy. Because these orphan sites no longer produce power, they provide no value other than storing SNF. An allocation strategy prioritizing orphan sites should be acceptable under the Standard Contract [2]. Once the facility begins accepting SNF, it is unclear if the Standard Contract will be able to give priority to future shutdown sites. Under the current assumption of OFF, it is assumed that these future shutdown sites will be treated the same as operating sites [15, 27]. The evaluations used these different acceptance rates and allocation strategies to study the impacts on shutdown reactor years, handling operations at a reactor, and handling operations at a CISF. The following conclusions were presented from the analysis [27]:

1. Site-specific allocation/acceptance strategies could have significant benefits with respect to at-reactor logistics and costs. These strategies can possibly allow for more efficient clearing of SNF from the reactor sites than an OFF allocation strategy.

2. Accelerating acceptance could potentially be the most efficient approach for reducing shutdown reactor years. However, aggressively removing SNF may not be possible due to the at-reactor constraints in moving SNF.
3. Additional evaluation of acceptance strategies is necessary to better represent when SNF can be moved from reactor sites.
4. Thermal or radiation exposure limits could have a significant impact on the ability to clear SNF from reactor sites. These constraints are well understood and documented for DPC systems, but transporting other systems relies on a variety of assumptions.
5. The evaluation does not model expected dose rates, which may prevent a canister from shipping.

1.4 Gap Analysis

The DOE spent fuel receipt paper [8] focused on developing allocation strategies that could evenly distribute the thermal output in a repository. It used hand-developed scenarios to determine strategies that lie in between FIFO and LIFO. The user changed the repository acceptance scenario in order to represent a different scenario. Analyzing the different scenarios provided a better acceptance strategy than FIFO or LIFO. At-reactor impacts and a cost-benefit study were not performed on these scenarios [8].

Nesbit cites a number of different strategies in his paper for shutdown reactor sites [7]. He determines that changing the allocation strategy of shutdown reactors does not violate the Standard Contract because the Standard Contract allows for prioritized removal of SNF from shutdown sites [2]. The paper determines some example allocation strategies based on the

recommendation from the DOE in “Preliminary Evaluation of Removing Used Nuclear Fuel from Shutdown Sites” [29]. The proposed allocation strategy for shutdown reactor sites is based on qualitative arguments that are not supported with a cost-benefit analysis.

The evaluations supported for the WMSA combine Nesbit’s and Ballou’s papers by analyzing different strategies using a number of scenarios. In addition to the combination of the methods, the system architecture evaluation analyzes different allocation strategies using relevant parameters in terms of at reactor operations and a cost-benefit evaluation for the different scenarios. Allocation strategy and acceptance criteria were used as variables. The different allocation strategies were implemented using a guess-and-check method. This involved systematically varying the allocation schedule in an attempt to achieve a certain metric. Utilizing guess-and-check is inefficient and must be changed if some variables change. The modelling software used for the evaluation is unable to include operational loading limits at operating or shutdown reactor sites. This may skew the results to increase loading at a particular reactor far above its allowable limit [27].

1.5 Problem Statement

Although there has been previous work suggesting general strategies for already shutdown reactors [9] and separate work utilizing scenarios to examine different allocation strategies [27], there has been no attempt to optimize the entire allocation strategy for removing SNF from reactors on a systems level. This work creates a method to determine the optimal allocation strategies for the minimization of the number of years SNF stays at shutdown reactor sites. Shutdown reactor years are defined in equation 1.5.1 where N is the number of reactor sites, Y_{FR}

is the year the site is completely cleared, and Y_{FD} is the year the reactor stops producing power (final discharge).

$$\textbf{\textit{Shutdown Reactor Years}} = \sum_i^N Y_{FR} - Y_{FD} \qquad \textbf{(1.5.1)}$$

An OFF allocation strategy removes SNF from reactor sites by date of initial discharge. As fuel is discharged from a reactor, it is added to a reactor queuing order. This makes the allocation strategy remove SNF from a multitude of different sites instead of focusing on clearing sites. Using an OFF strategy (FIFO) does not promote removing SNF from shutdown reactor sites, which allows the number of shutdown reactor years to be much larger than alternative strategies. The Center for Advanced Nuclear Energy Systems estimates a value of $8M per reactor shutdown year [22]. Reducing the number of shutdown reactor years could therefore result in a significant reduction in system cost.

Reducing the total cost for removing SNF from the different reactor sites may help to promote activity in the disposition of SNF. As more operations and cost benefit research is done to remove SNF from reactor sites, the government's plan may become clearer. As the details of the plan come to fruition, the Department of Energy may gain credibility with the different stakeholders as well as put public sentiment in their favor. A major goal in disposing of SNF is to continue to prove that nuclear power is a safe alternative from start to finish compared to other sources of energy.

Chapter Two

Mathematical Methods and Algorithms

Integer programming, simulated annealing, and aspects of a genetic algorithm are all identified as methods for optimization that are utilized in developing an optimal allocation strategy. These methods use both heuristics and analytics to achieve an answer. Solving a problem using heuristics allows the problem to use past analysis to improve on the answer, but does not guarantee the answer is correct.

2.1 Integer Programming

A Mixed Integer Program (MIP) or Integer Program (IP) is a constrained optimization problem, in which a set of values $(x_1,\ x_2, \dots , x_n)$ is found which maximizes or minimizes a linear objective function *z*, while satisfying a system of simultaneous linear equations and/or inequalities. To be classified as an Integer Programming problem at least one of the variables must be restricted to integer values. Mathematically a mixed integer program is expressed in equations 2.1.1-2.1.4[16]:

$$(\boldsymbol{MIP})\ \boldsymbol{Maximize\ z} = \sum_{j} \boldsymbol{c_j x_j} + \sum_{k} \boldsymbol{d_k y_k} \qquad \textbf{(2.1.1)}$$

$$subject\ to \sum_j a_{ij}x_j + \sum_k g_{ik}y_k \le b_i \quad (i = 1, 2, \dots, m) \tag{2.1.2}$$

$$x_j \ge 0 \qquad (j = 1, 2, \dots, n) \tag{2.1.3}$$

$$y_k = 0, 1, 2, \dots \qquad (k = 1, 2, \dots, p) \tag{2.1.4}$$

Another way to describe a mixed integer problem is described in equations 2.1.5-2.1.7 below in terms of bound and linear constraints. Each representation of the problem implies the same basic formulation

Objective: $$minimize\ C^T x \tag{2.1.5}$$

Constraints: $$A\,x = b\ (linear\ constraints) \tag{2.1.6}$$

$$I \le x \le u\ (bound\ constaints) \tag{2.1.7}$$

$$some\ or\ all\ of\ x_j\ must\ take\ integer\ values\ (integrality\ constraints)$$

The integrality constraints employed allow the capture of the discrete nature of various decisions by the model. In many cases a decision variable can be restricted to 0 or 1, called a binary variable, where it can be used to decide if an action took place or not, such as shipping a container or building a warehouse.

Mixed Integer Linear Programming problems are generally solved using a linear-programming based branch-and-bound algorithm. The steps to solve a Linear Program (LP) based branch-and-bound problem are as summarized as follows:

- Begin with original MIP and remove all integrality restrictions.
- Solve the resulting LP.
- If the resulting solution satisfies all the integrality restrictions, then this is the solution.

- If result does not satisfy all integrality restrictions, pick a fractional value in the LP problem and constrain it to use an integer on either side (rounding up or down). This creates two MIP problems instead of just the original.
- Steps 2-4 are repeated using the new MIP until all the conditions of integrality have been met.
- The optimal value is chosen from the new MIPs.

The technique in step one creates an LP, which is called the linear programming *relaxation* of the original MIP. This LP can be solved. If the resulting solution meets the conditions of integrality, then this is the optimal solution. A much more probable result has at least one of the integrality conditions in a fractional form. At this point, one of the fractional variables is chosen to satisfy the condition of integrality. The fractional result is rounded both down and up to achieve two bounding integer values. This results in two separate MIPs that branch off from the original MIP. These two new MIPs (nodes) are treated just as the original MIP was treated until all the conditions of integrality have been satisfied. At this point, the optimal solution is known, because all the branches have determined values.

Additional logic can be applied to reduce the number of branches that must be solved. After the LP relaxation problem has a solution that satisfies all the conditions of integrality, the node is termed *fathomed*, meaning no more branches need to split from this node. If this solution is the first node satisfying all the constraints of the original MIP, it is now the incumbent solution. The incumbent solution changes each time a more optimal value is found in another node. In order for a node to be fathomed, the solution could be deemed infeasible or the result of the LP relaxation produces a value that is less optimal than the incumbent solution [32].

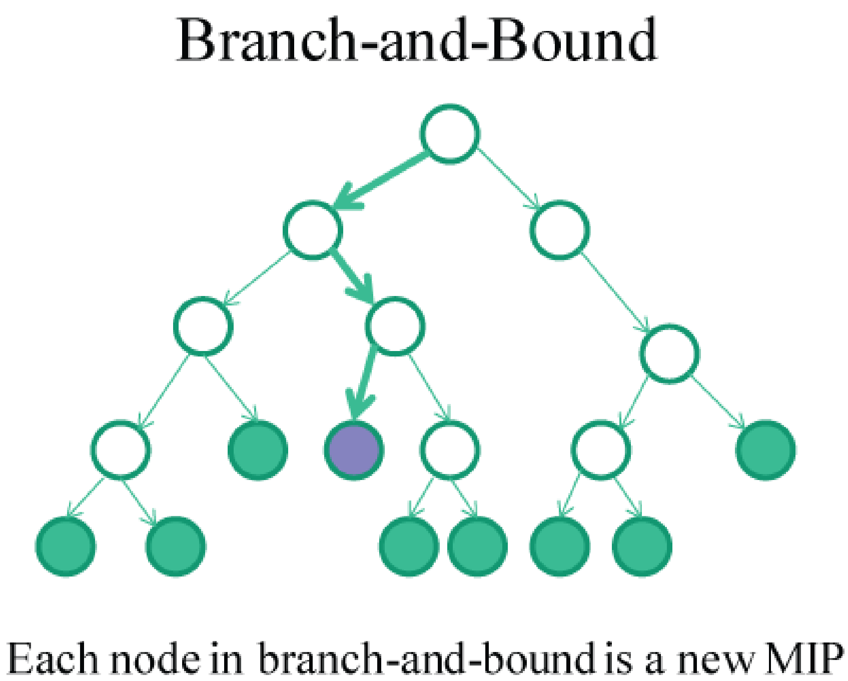

Figure 3: Branch-and-Bound demonstration

Capabilities of MIP algorithms have greatly improved in recent years by using *presolve, cutting planes, heuristics,* and *parallelism.* Presolve refers to reducing the problem before the start of the brand-and-bound procedure. These reductions are intended to tighten the problem's formulation as well as reduce its overall size. A common practice in presolve is attempting to combine constraints in order to achieve variables that must be constant. If the reduction is not caused by a condition of integrality, then it is classified as an LP-presolve reduction. Another common practice is to use the condition of integrality to remove variables alltogether. This can occur when the sum of two integer variables equals anything less than one. Although the statement is valid, it only works if both variables are zero, thereby they can be removed from the entirety of the problem [32].

Utilizing cutting planes in solving a MIP is more complicated than the branch-and-bound method, but many of the improvements in the capability for algorithms to solve MIPs are due to the cutting plane method. In the following MIP where $S := \{(x, y) \in Z_+^n \times R_+^p : Ax + Gy \leq b\}$, let P_0 be the natural relaxation of S.

$$\textbf{MIP:} \quad \max\{\boldsymbol{cx} + \boldsymbol{hy}: (\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{S}\} \tag{2.1.8}$$

$$\max\{\boldsymbol{cx} + \boldsymbol{hy} : (\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{P_0}\} \tag{2.1.9}$$

Let z_0 be the optimal value and (x^0,y^0) an optimal solution. To utilize the cutting plane method, an inequality $\alpha x + \gamma y \leq \beta$ that is satisfied by every point in S must be found. A valid inequality that is violated by the optimal solution is a cutting plane separating the optimal solution from S. If $\alpha x + \gamma y \leq \beta$ was a cutting plane then

$$\boldsymbol{P_1} := \boldsymbol{P_0} \cap \{\boldsymbol{x}, \boldsymbol{y}): \boldsymbol{\alpha x} + \boldsymbol{\gamma y} \leq \boldsymbol{\beta}\} \tag{2.1.10}$$

The following cutting plane algorithm can be implemented.

- Solve the linear program $\max\{cx + hy : (x, y) \in P_i\}$
  - If the optimal solution (x^i,y^i) belongs to S, this is the optimal solution.
  - Otherwise solve the separation problem in which you find a cutting plane that separates (x^i,y^i) from S. Set $P_{i+1} := P_i \cap \{x, y): \alpha x + \gamma y \leq \beta\}$ and repeat the first step.

Figure 4 gives a demonstration of how cutting planes work. They tighten the formulation by removing undesirable fractional solutions. This is similar to the presolve method, but cutting planes work during the solution process and do not have the side effect of creating additional sub-problems [33]. Heuristics is very helpful when the problem cannot be solved to a provable optimality. The MIP may be too difficult or there may be a user-imposed time restriction that the algorithm can run.

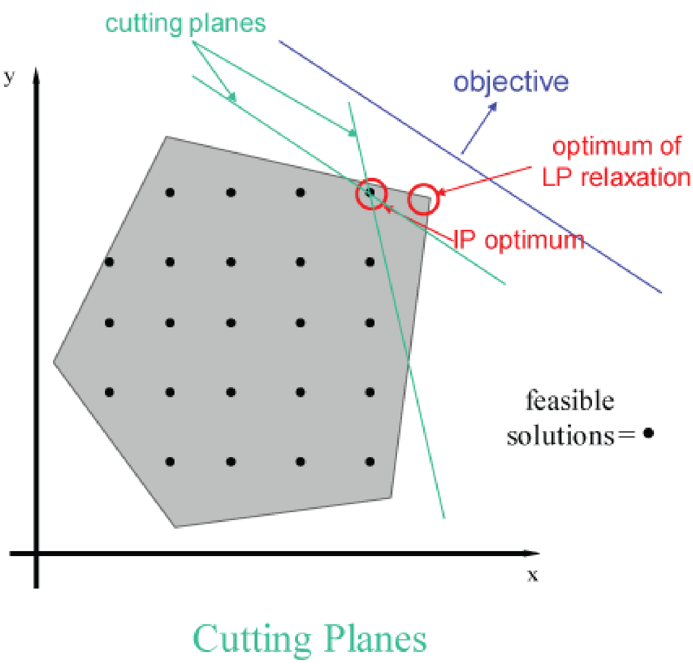

Figure 4: Cutting Planes demonstration

Either way, it is important to have the best possible feasible solution when the run is terminated. A good incumbent value helps to remove unnecessary branches from nodes, because the incumbent value must be less optimal than the LP solution to continue branching. A common practice is to do a little extra work at some nodes to see if a good integer feasible solution can be extracted, even though integrality has not yet been achieved due to branching. If many of the integer values are close to being integers, it may be good to round to the value in which they are hovering around.Then these values can be fixed and the resulting LP relaxation can be solved so that the integer variables will completely converge to a successful solution satisfying all the constraints [32].

Another way that solving MIP has progressed is in using parallelism. This results in running different branch nodes that can be processed independently. The root node presents limited parallelism opportunities since every branch stems from the original MIP. If large search trees are used, parallelism can effectively exploit multiple different cores, while the root node is constrained in the ability to use multiple cores [32].

2.2 Simulated Annealing

Simulated annealing emulates the process of cooling metal. The temperature is reduced slowly with steps long enough to reach thermodynamic equilibrium at each increment, instead of cooling the temperature at a constant rate. The annealing process brings the solid to a lower energy state after raising the temperature. For many materials, the lower energy results in a regular, crystal-like atomic structure. The annealing can be summarized in the following steps

1. Raise the temperature very high in order to bring the solid to a point of fusion
2. Cool the solid to a solid state with minimal energy utilizing a specific temperature reduction plan

The simulated annealing algorithm utilizes a probabilistic method first proposed by Kirkpatrick, Gelett, and Vecchi in 1983 for finding the global minimum of a function that may contain many local minima [18]. It consists of a finite set of a discrete-time inhomogeneous Markov chain. The algorithm has the probability to go in the direction opposite of improved results. Depending on the “Temperature,” the algorithm may select a solution to the function that is worse than a previous solution in order to increase the probability of finding the global optimum instead of a local optimum [18].

A simulated annealing method is made from the following elements:

1. A finite set *S*.
2. A real-valued cost function *J* that is defined on the finite set *S*. Let *S** be a subset of *S* representing the set of global minima of the function *J*.
3. For each $i \in S$, a set S(i) is a subset of S-$\{i\}$, called the set of neighbors of *i*.
4. For every *i*, a collection of positive coefficients q_{ij}, $j \in S(i)$ where the sum of all elements *j* of $S(i)$ $q_{ij}=1$. It is assumed that *j* is an element of *S(i)* only if *i* is an element of *S(j)*.
5. A decreasing function $T: N \rightarrow (0, \infty)$ known as the cooling schedule. *N* must be a set of positive integers and *T(t)* is the temperature at a certain time *t*.
6. An initial "State" *x(0)* is an element of *S*.

When the previous elements are applied, the simulated annealing algorithm consists of a discrete-time inhomogeneous Markov chain *x(t)*. The evolution of the chain for the current state $x(t) = i$ uses a neighbor *j* of *i* at random. The probability that any particular element *j* of *S(i)* is selected is equal to q_{ij}. When j has been determined, the next state *x(t+1)* can be determined using equations 2.2.1-2.2.3:

$$\boldsymbol{If\ J(j) \leq J(i),\ then\ \ x(t+1) = j.} \tag{2.2.1}$$

$$\boldsymbol{If\ J(j) > J(i), then\ x(t+1) = j\ with\ probaility = e^{-\frac{J(j)-J(i)}{T(t)}}} \tag{2.2.2}$$

$$\boldsymbol{otherwise\ x(t+1) = i} \tag{2.2.3}$$

It is evident from the probability equation that the simulated annealing algorithm is more probable to accept a solution that is not an improvement when the Temperature is high. Since the Temperature continually decreases, the chances to escape a local optimum to find a global

optimum decrease. The simulated annealing algorithm essentially becomes a local search algorithm at very low temperatures. If the Temperature were to remain constant, the simulated annealing algorithm could theoretically become a combinatorial algorithm by testing every combination of parameters. One way to accomplish this combinatorial approach is by simulating the Markov chain until it reaches equilibrium, which is known as the Metropolis algorithm [34].

Using the Metropolis algorithm, a sequence of solutions can be generated in the state space by equating the admissible solutions with the possible states of the solid and the optimization function with the energy of the solid. The simulated annealing algorithm coupled with the Metropolis algorithm can be used to generate effective solutions. If the Temperature or the Metropolis algorithm's parameters are not set broadly enough, the simulate annealing algorithm will only act as a local search which occasionally makes moves which will lead to a cost increase but never leaves a local optimum. The point of these upward moves is to escape from local optima, but this will not happen without the proper parameters.

The performance of the simulated annealing algorithm is cited in many studies. The main result in finding necessary and sufficient conditions for convergence is due to Hajek's theorem.

> *THEOREM 1 [35]. We say that state i communicates with S* at height h if there exists a path in S (with each element of the path being a neighbor of the preceding element) that starts at i and ends at some element of S* and such that the largest value of J along the path is J(j) +h. Let d* be the smallest number such that every element i in S communicates with S* at height d*. Then, the SA algorithm converges if and only if* $\lim_{t\to\infty} T(t)=0$ *and*

$$\sum_{t=1}^{\infty} exp\left[\frac{-d*}{T(t)}\right] = \infty. \tag{2.2.4}$$

This essentially states that if an infinite number of attempts are made to escape from a local minimum, then the probability of escape is guaranteed. As the number of attempts gets smaller due to a lower Temperature or faster "cooling schedule" then the probability of escaping a local minimum decreases. The simulated annealing algorithm can be used to solve a large number of combinatorial optimization problems having a stochastic convergence to an optimal solution, but is problematic when there are several quasi-optimal solutions.

2.3 Genetic Algorithm

Genetic algorithms are inspired by Charles Darwin's Theory of Evolution in the 19th century. According to his theory, a population of individuals evolves through sexual reproduction. The offspring that have certain characteristics best suited to their environment are able to get more resources than others. This leads them to reproduce more, which further enhances the trait that is best suited to the environment.

Genetic algorithms were first proposed by Holland [38] and Jong [39] in 1975, although a case can be made that some of the ideas appeared as early as 1957 [40] through the simulation of genetic systems. Initially, the genetic algorithm was utilized as an adaptive search algorithm, but it has mostly been tasked as a global optimization algorithm for both combinatorial and numerical problems [36]. In 1989, Koza termed genetic programming [41][42], which is the application of genetic algorithms. More recently, the term evolutionary algorithms has been used by researches to include evolution strategies, evolutionary programming and genetic algorithms as the computational framework is very similar [36]. In this work, the phrase genetic algorithm/genetic programming will continue to be used.

A genetic algorithm is realized by specifying the search space and identifying the heuristic function [19]. It emphasizes genetic encoding of potential solutions into chromosomes and includes genetic operators to these chromosomes. As in many solution methods, this transforms the problem from one space into another space. The success of utilizing a genetic algorithm is highly dependent on the genetic representation. A representation that can be searched efficiently will perform much better than poor individual representation [36].

One specific type of algorithm called a canonical genetic algorithm also known as a simple genetic algorithm uses a binary representation with one point crossover and bit-flipping mutation. The binary representation will model each individual by a binary bit (0 or 1).

A point crossover for binary strings *x* and *y* with length *n* first generates a crossover point between 1 and *n*-1 uniformly at random. This point will be known as *r*. The first offspring consists of the first *r* bits of the *y* string and the last *n*-*r* bits of the *x* string. The mutation occurs by bit, meaning every bit of the individual has a certain probability of flipping from 0 to 1 or from 1 to 0.

In order to use the genetic algorithm for a specific problem, six elements are required:

1. A coding principle for the chromosome that connects each point of the state space to the data structure while including all the necessary information from these points.
2. A mechanism for generating the initial population must be capable of uniformly distributing a population of individuals to act as a base for future generations.
3. A criterion capable of judging the suitability or fitness of the individual compared to other individuals for the environment must be decided.

4. A selection principle that allows statistical identification of best individuals must regulate the selective process to a variable degree effectively.
5. Operators that perform crossover and mutation to diversify the population must be used. The crossover operator mixes the genes of individuals in the population while the mutation operator creates new genes.
6. A dimension parameter that specifies the population size, number of generation to simulate and the probability to apply operators must be specified.

Figure 5 and the following steps illustrate the simple genetic algorithm: [37][58]

1. Generate an initial random population $P(0)$ and set i=0
2. Evaluate the fitness of each individual in $P(i)$
3. Select parents from $P(i)$ (P1 and P2) based on each parents fitness using the formula below given the fitness as $f_1, f_2, \ldots, f_n$ for the fitness of n individuals

$$\boldsymbol{p_i = \frac{f_i}{\sum_{j=1}^{n} f_j}} \qquad \textbf{(2.3.1)}$$

4. Apply the crossover to the selected parents
5. Apply mutation to the new individuals that had been crossed over
6. Replace parents by offspring to produce generation P(i+1)
7. Repeat steps 2-6 until the specified time has run out or a condition satisfying the criterion is met.

There are three main ways to generate the initial population. If no prior exists concerning the optimum state space, the individuals may be randomly generated using a uniform distribution for each component in the state space. These individuals must still satisfy the initial constraints. In

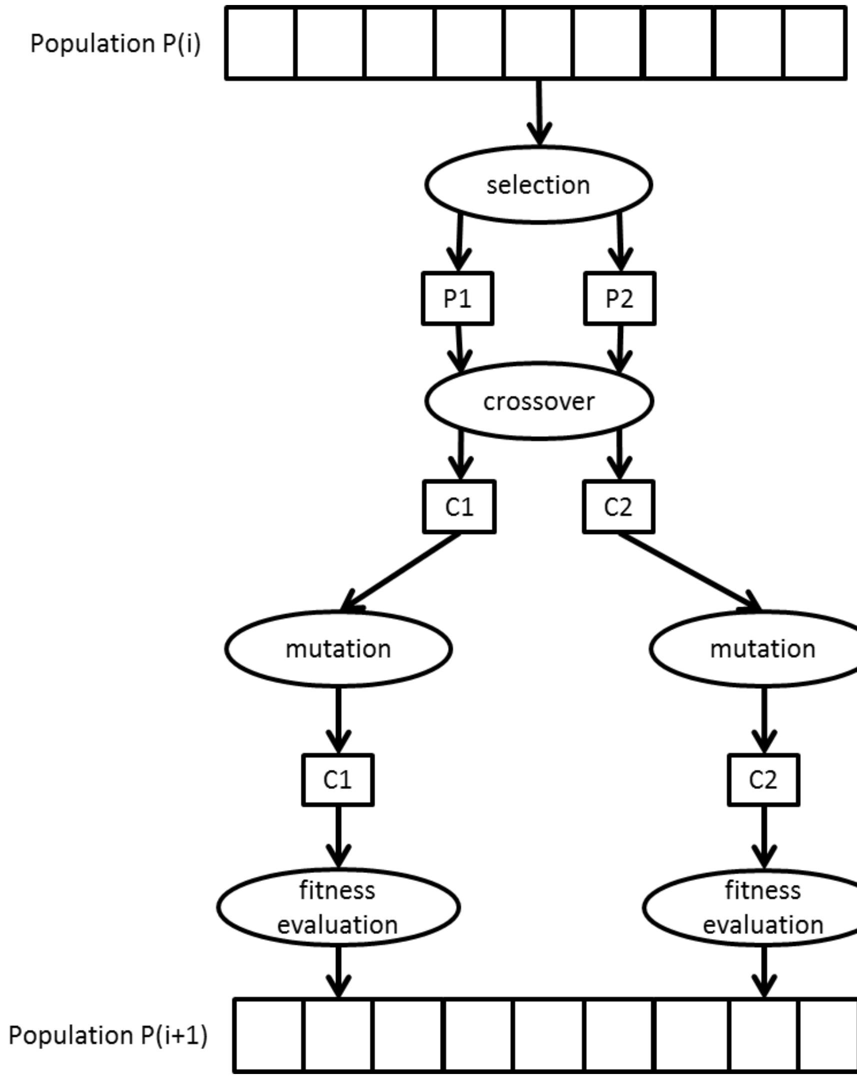

Figure 5: Process flow diagram for a Genetic Algorithm [58]

cases that prior information indicates an optimal subdomain, individuals should be randomly generated within this subdomain to accelerate convergence. In some cases, it is too difficult to randomly generate individuals corresponding to the known constraints. In these cases, the constraints can be instituted by utilizing penalties. An individual not meeting a constraint incurs a penalty to reduce its fitness [58].

The crossover operators exist to enrich the diversity of the population by changing the genes of chromosomes [53]. Conventionally, a crossover takes place when two parents generate two children, but crossovers can work with N parents and K children. The first type of crossover used in genetic algorithms involved chromosome slicing or cutting the two parents in two pieces and crossing the piece over to the other parent[58] [38]. Using this principle, the parents can be divided in a number of different sub-chains. Enough sub-chains can be developed from the parents to effectively create a process to randomly inject genes from the parents into the children [54]. This method works well for discrete problems. Another type of crossover that is typically employed for continuous problems is called barycentric crossovers[58]. This crossover selects two genes in each of the parents at the same position. These are subject to a weighting coefficient suited to the domain extension of the genes (minimum and maximum value of each gene).

Mutation operators enrich the population gene space ensuring that the genetic algorithm is capable of considering all points in the state space. In discrete problems, a gene in the chromosome is randomly selected and replaced with a new one by the mutation operator. This also works for continuous problems, but random noise is added, while ensuring the gene stays in its domain of extension. Utilizing adaptive mutation operators allow the mutation rate to be optimized by coding it directly in the chromosome. Coding the mutation directly into the chromosome will only work in spaces of low dimension [58].

Genetic algorithms do not require the derivative of the objective function like many other optimization techniques. In addition, the genetic algorithm can be used on entire systems instead of just models as long as there is access to a computed or simulated "fitness" in evaluating each chromosome and that proposed individuals remains within the domain of operation. The purpose of selection in the genetic algorithm is to identify the best individuals and remove the worst individuals utilizing statistics. There are a number of different specific examples for selection that suit different types of problems [58]. A sample list of certain selection strategies are listed below:

- Roulette wheel selection [55]
- Stochastic remainder without replacement selection [55]
- Selection by rank [55]
- Stochastic tournament [56]
- Adaptive selection [57]

Genetic Algorithms have been applied in many different situations ranging from scheduling [43], adaptive control [44], travel [45], transportation [46], shape synthesis [47], neural networks [48], molecular synthesis [49], and filtering [50] in both the medical [51] and air traffic control [52] fields. The wide variety of uses by the genetic algorithm is a testament to its usefulness.

2.4 Combinatorial Algorithm

Combinatorial algorithms are classified into three divisions. Generation algorithms construct all the combinatorial structures of a particular type. The combinatorial structures that may use the generation algorithm include subsets, permutations, partitions, trees, and Catalan families. This

algorithm lists all possible objects in a certain order. In some instances, it is necessary to predetermine the position of an object in a generated list without having to generate the whole list. This utilizes a process called *ranking* [59].

Enumeration algorithms compute the number of different structures of a particular type. Each generation algorithm is also an enumeration algorithm, but each enumeration algorithm is not a generation algorithm. In a generation algorithm, it is always possible to count the number of objects generated, but objects cannot be generated just from a particular count. Equation 2.4.1 gives a simple enumeration algorithm [59]

$$\binom{\boldsymbol{n}}{\boldsymbol{k}} = \frac{\boldsymbol{n}!}{(\boldsymbol{n}-\boldsymbol{k})!\boldsymbol{k}!} \tag{2.4.1}$$

A search algorithm finds at least one example of a structure of a particular type if the structure does exist. One variation of the search algorithm is an optimization algorithm capable of finding an optimal structure of a given type. This requires a “cost” to measure a particular structure. This specific algorithm is for many optimization problems classified as NP-hard. An NP-hard problem cannot guarantee an optimal solution in polynomial time, but it can search for a particular structure that meets all the necessary parameters [59].

Two subsets of the generation algorithm are the sequential generation and the ranking algorithm. The sequential generation can produce the desired output in a lexicographic order and generate the objects using a minimal change algorithm. Both of these algorithms utilize the previous object or successor. A ranking algorithm determines the place or rank an object has among other objects given an order. This allows complex data that has many relations be accessed using a single index [59].

The search algorithm can be used to solve a variety of problems, but four of the most common types are listed below [59].

1. **Decision Problem:** Answers a "yes" or "no" question
2. **Search Problem:** Produces the value of the decision problem
3. **Optimal Value Problem:** Finds the largest target profit for a decision problem answering "yes"
4. **Optimization Problem:** Finds an array of answers satisfying the constraints

2.5 Pareto Optimization

A Pareto Curve is a set of all possible solutions not dominated by the other solutions in order to see the trade-off between different objective functions. It is used predominantly in multi-criteria optimization. In a multi-criteria minimization problem, with $\gamma \geq 1$ objective functions $G_i, i = 1, \ldots, \gamma$, its Pareto curve P is all γ-vectors meeting the criteria $v = v_1, \ldots, v_\gamma \epsilon P$ if a feasible solution *s* exists for $G_i(s) = v_i$ for all *i*, and no other feasible solution *s'* where $G_i(s') \leq v_i$ for all *i*. The Pareto curve attempts to simultaneously minimize multiple objective functions in determining an optimal solution. If a solution is not found which minimizes all the objective functions, the exact Pareto curve is deemed infeasible. In many occasions, it is infeasible to compute the exact Pareto curve. For these cases, an approximate Pareto curve can be utilized that acts as a set of cost vectors of feasible solutions. This requires every feasible solution *s* have a feasible solution *s'* with cost vector from $P_{(1+\varepsilon)}$ where ε>0 and $G_i(s') \leq (1 + \varepsilon)G_i(s)$ for all i=1,…,γ [60].

In the early 1990s, efficient evolutionary multi-objective optimization methods were developed to find multiple Pareto-optimal solutions in a single simulation run [62, 63, 64]. Classical generative methods were suggested as early as the 1980s [65]. The generative principle utilizes a multi-objective optimization problem that is scaled to a single-objective function using the parameters. One way of scaling uses a weighted sum approach for the relative weights of objective functions. Another scaling technique utilizes a vector of ε values for converting objective functions into constraints called the epsilon constraint approach [61]. The Tchebyshev method combines these approaches by using a weight vector to form the objective function. After forming the objective function into a parameterized single objective optimization problem, it may be solved. If this function is solved to optimality, then it will converge to a Pareto-optimal solution in every occurrence [66].

Chapter Three

Tractable Validation Model

The tractable validation model (TVM) simulates removing SNF from reactor sites to demonstrate the effectiveness of different algorithms in reducing the total number of shutdown years incurred by the system. The goal of the TVM is to validate the implementation of the optimization algorithms on a problem space small enough such that the true optimum is analytically known via exploration of all permutations (via a combinatorial algorithm). By validating the optimization algorithms against a space where the solution can be analytically known, they can then be applied to larger, more representative systems where the number of permutations is too large for a combinatorial algorithm to effectively process. This provides a true optimal solution as a baseline for the other algorithms to achieve.

The TVM receives inputs specifying when reactors discharge assemblies as well as the burnup and enrichment of an assembly. Other inputs give data for canisters and directions for selecting a canister to load based on the pool and year. The TVM utilizes Java version 8.91 and follows an object-oriented programming approach.

3.1 Object-Oriented Programming

The TVM utilizes object-oriented programming to replicate similar objects and to give certain objects ownership of others. A reactor owns the pools and the ISFSIs that are on site. The pools

own the assemblies contained within its walls just as canisters own the assemblies packaged inside. The hierarchal approach is a fundamental concept of the TVM, because the simulation can manipulate and track objects to determine the fitness of a particular solution. The fitness variables become objects, which help determine the optimal solution for the scenario.

Object oriented programming is a programming paradigm that utilizes “objects” that may contain data populating different fields owned by a class of object. In order to setup and perform differing operations “methods” are employed which act similar to functions. The most popular object oriented programming approaches are class-based, which means objects are instances of classes [67]. The methods may also be contained in a class but could operate without a class on its own using an input (can be void) and an output (can be void).

The advantage to object-oriented programming draws upon the fact that computers are state machines, meaning that a finite collection of attribute values from a finite range characterize the machine at any time. A machine also has a finite set of rules that determine the transition of it from one state to the next. The most important states of a machine are state variables where some attributes are internal (private) and some attributes are external (public) [68].

The computer is most useful when it can reproduce or simulate behavior that is interesting such as removing canisters from reactor sites. By reproducing this behavior, the user can model an external entity on a finite deterministic state machine. This requires mapping features of the entity to features of the machine effectively modeling the entity as a finite state machine. Private and public member types are mapped onto corresponding data types in the computer where they can characterize the dynamic behavior of the entity in terms of state transitions. These transitions

are implemented as functions. The definition of a function or method intrinsic to the object containing a data variable is encapsulation [68].

In order to create an object, first declare a template for the object called a class to the compiler. The class instructs the compiler about the fields and methods. Once a class is declared, the compiler constructs an object where certain settings can be implemented giving the object different attributes. The same template can create multiple objects, and they become dynamic by calling methods that contain the appropriate parameters [68].

3.2 Inputs for the TVM

The TVM requires five data sheets in order to run: the 'Fuel Projection Table', the 'BWR Heat Table', the 'PWR Heat Table', the 'Canister Info Table', and the 'Canister Matching Table'. Each one of these tables must be formatted correctly in order to run the optimization model. Table 1 shows the fields that are detailed in The Fuel Projection Table.

The TVM assigns these attributes to assembly objects within the model in order to differentiate between different assemblies. The amount of MTU per assembly helps determine whether the reactor is a PWR or a BWR. A BWR has a value less than 0.3 MTU while a PWR has a value greater than 0.3 MTU. The burnup, enrichment, and age of the SNF help determine the thermal output of the SNF by using linear interpolation on heat curves.

The fuel projection table is set up to allow different pools to populate reactors. The pools contain the assemblies, which are the most basic elements in the TVM. In some instances, SNF was

Table 1: Fuel Projection Attributes

Fuel Projection Attributes	Description
Batch ID	Order of discharge from reactor to spent fuel pool
CALVIN_RX_ID	The identification number for the reactor
MTU	The amount of Uranium in the batch (Metric Tons Uranium)
NUM_ASSM	Number of assemblies in a batch
Burnup	The amount of power produced from a quantity of Uranium [69]
Enrichment	The percentage of fissionable Uranium [70]
Discharge Year	The year in which the assembly was discharged from the reactor
Pool_ID	Utilized in testing the code in CALVIN (Different than Pool Identification Number)
CALVIN_ID	The identification number for the pool
Dry_Year	The year in which the assembly was loaded into dry storage

moved to a different storage location from where it was discharged. In order to model this phenomenon, the fuel projection table allowed users to create imaginary pools at a reactor.

These pools act just like any other pool, but they contain SNF from another reactor. Creating imaginary pools allows the user to more adequately model the current SNF system. A further explanation of imaginary pools is in section 3.2.

The BWR and PWR Heat Tables have the same format but have different values. The BWR heat table describes the heat curves to use when dealing with SNF from a BWR while the PWR heat table describes the heat curves to use when dealing with SNF from a PWR. Table 2 shows the fields that are in the heat curves. The Burn Curve has twelve separate columns representing the following burnups: 0.001, 0.01, 0.1, 1, 10, 20, 30, 40, 50, 60, 70, and 75 GWD/MTHM.

Table 2: Heat Curve Attributes

Heat Curve Attributes	Description
Age	The amount of time in years SNF has been discharged
Enrichment	The percentage of fissionable uranium [70]
Burn Curve [X]	The thermal output produced by an assembly for a particular burnup

The thermal output represented in Table 2 by Burn Curve [X] is given for a particular burnup, the enrichment of the assembly, and the age of the assembly. The final thermal output of the assembly is found by linearly interpolating between the two nearest thermal outputs and burnups. This is discussed in further detail in section 3.4.

The Canister Info Table gives information that corresponds to a canister. Each canister has certain attributes detailed in Table 3 that change when a canister can be loaded, removed from the site, and how many assemblies can fit in a canister.

Table 3: Canister Info Attributes

Canister Info Attributes	Description
Canister ID	Canister identification number
Number of Assemblies	The maximum number of assemblies a canister can contain
Thermal Storage Limit	The maximum allowable thermal output a canister may have in order to store
Thermal Transportation Limit	The maximum allowable thermal output a canister may have in order to transport off site.
Canister Type	Binary variable 0 for BWR and 1 for PWR

The attributes obtained from the Canister Info table help build the canister object in order to store and remove SNF from reactor sites.

The Canister Matching Table provides instructions to determine which canister is associated with a given pool for a given year. The attributes are in Table 4.

Table 4: Canister Matching Attributes

Canister Matching Attributes	Description
Year	The particular year a canister is utilized
Pool Number [X]	The pool needed to find the correct canister

The Canister Matching Table provides the TVM with an easy lookup to determine which canister should be used for a particular pool in a particular year.

3.3 Objects in TVM

The TVM utilizes an assembly, canister, pool, ISFSI, reactor, Allocate_Year_ISFSI, reactor site, and removal object. These objects contain different attributes and defining characteristics set by the object's template. Tables 5-12 describe the objects' attributes. Figure 6 contains the hierarchy of the objects within the TVM.

The reactor site object owns all the other objects associated in the logistics. A reactor site may have multiple reactors on site, or it may just have one. The ISFSI, stemming from the reactor, owns canisters, which contain assemblies. There can only be one ISFSI for every reactor, but an ISFSI can hold multiple canisters and a canister can hold multiple assemblies. On the other side, the reactor owns pools, which own the assemblies that are located within.

This includes assemblies that have yet to be discharged from the reactor vessel. A reactor may own multiple pools and a pool can own multiple assemblies. A visual interpretation of the reactor site is in Figure 7. The top illustration contains two reactors and an ISFSI with canisters, which contain assemblies. The bottom illustration shows the pools containing assemblies inside one of the reactors.

The assembly is the basic unit of operation in the TVM. The Fuel Projection table provides data to complete the list of attributes in Table 5. The attributes classify the assemblies into different reactors and different pools. The burnup, enrichment, and discharge year calculate the thermal output of each assembly together. When an assembly is still in the reactor, the assigned thermal output is a value well beyond the heat limit for any canister.

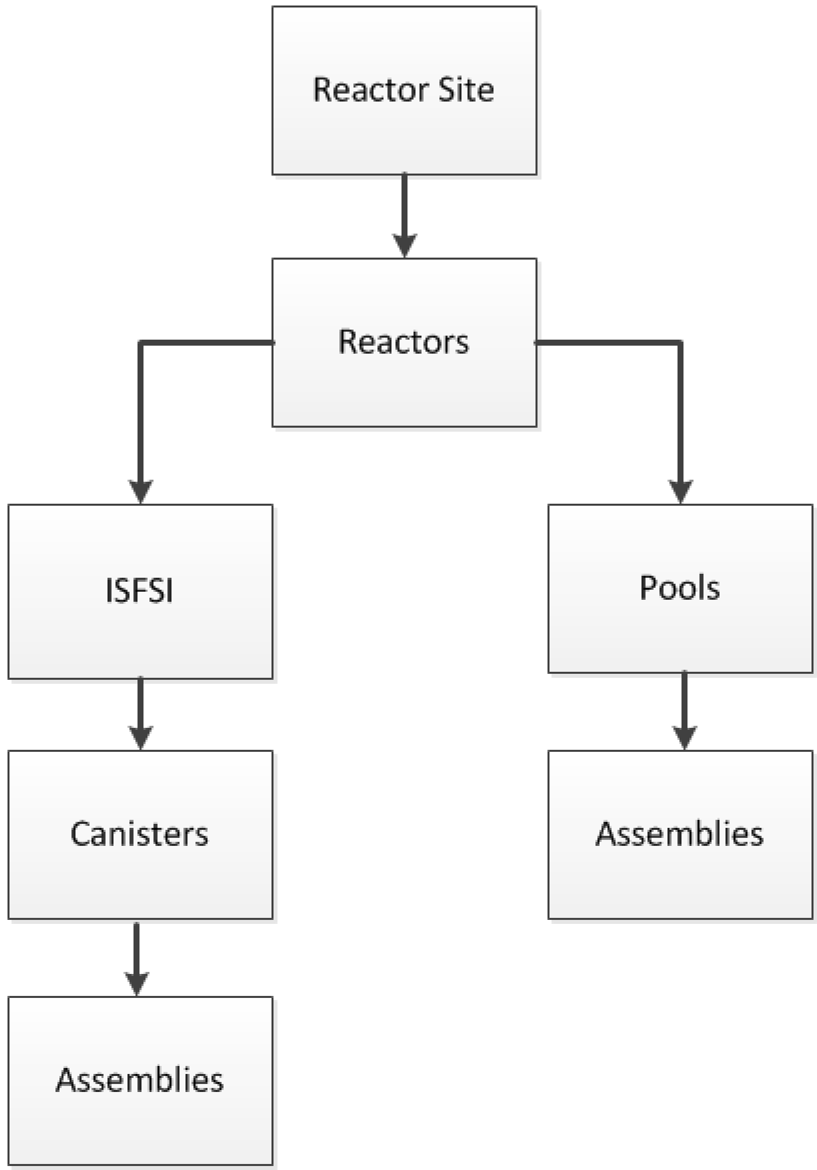

Figure 6: The hierarchy of the TVM

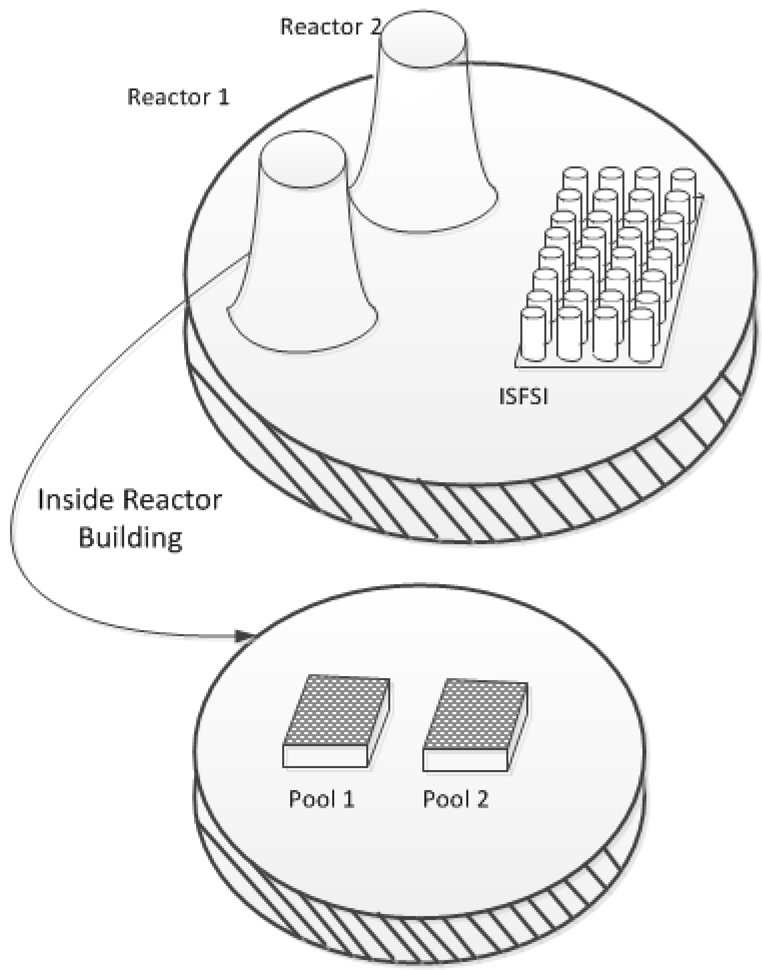

Figure 7: Example Reactor Site with reactors, pools and an ISFSI.

Table 5: Attributes for the Assembly Object

Attributes	Description
AssemblyIndex	The assembly number in order from when it is discharged from the reactor
Calvin_RX_ID	The reactor identification number
Pool_ID	The pool identification number
Burnup	The burnup of the particular assembly in the reactor core
Enrichment	The enrichment of the assembly
DischargeYear	The year in which the assembly was removed from the reactor
DryStorageYear	The year in which the assembly was moved to dry storage
Thermal_Output	The thermal output in Watts
Reactor_Type	0 for a BWR and 1 for a PWR
Shipped	Binary variable of whether or not the assembly has been shipped
FailedToShip	Binary variable of whether the assembly failed to ship in a year
Interim_Storage	Binary variable of whether the assembly is in interim storage at the reactor site

The high thermal output assigned to the assembly prevents the removal of the accidental removal of an assembly that has not yet left the reactor.

The pool is another object that the TVM uses. It is responsible for grouping assemblies correctly so they are loaded from a single pool. This prevents the consolidation of assemblies from a single reactor but different pools. The basis behind this assumption stems from the difficulties of moving assemblies between pools at most reactor sites. Some reactor sites have the capability to move assemblies between pools, but this is not considered in the TVM. The TVM assumes that each pool can only use assemblies stored within the pool to load canisters. Table 6 gives a list of pool attributes associated with the pool.

The pool attributes help the TVM keep track of the location of like assemblies and provides the number of shippable canisters from the pool. In order to remove assemblies from pools, the TVM incorporates a canister object.

The canister contains information provided by the Canister Info table. A method converts the canister info into canister attributes, which give a detailed description of loading and shipping practices, into canister data fields. The canister also provides a place and unit for assemblies stored outside of the pool. The canister attributes are listed in Table 7.

The canister object acts as a transportation and storage container for the model. It is the fundamental unit in the allocation strategy based on the assumption that the time to load, move, and ship a canister would be independent of size. This assumption traces back to the advancements in welding techniques in loading assemblies in pools at reactor sites and the incredibly long amount of time it takes to load a single dry storage canister [76].

Table 6: Pool Attributes

Attributes	Description
AssemblyList	An array of assemblies that are contained within the pool
Reactor_ID	The reactor identification number for the location of the pool
Pool_ID	The pool identification number
Shutdown_Year	The date the last assembly is discharged into the pool
Cans_from_Reactor	A dynamic variable that counts the number of canisters a reactor has shipped
Pool_Capacity	The maximum number of assemblies a pool is capable of holding
Assemblies_in_Pool	A dynamic variable counting the number of assemblies still left in a given pool
Shippable_Cans	A dynamic variable calculating the number of canisters the pool is capable of shipping based on thermal output

Table 7: Canister Attributes

Attributes	Description
Can_size	The maximum number of assemblies a canister can contain
Can_ID	Canister identification number
Type	Binary Variable with 0 as BWR and 1 as PWR (only one type of assembly can be loaded in a canister)
Can_heat_limit_store	The heat limit (Watts) for which a canister may be stored
Can_heat_limit_trans	The heat limit (Watts) for which a canister may be shipped
AssemblyArray	An array of assembly objects contained within the canister

Under this assumption, the allocation strategy has a basic unit of canisters with a limit on the number of canisters.

In addition to being the basic unit of allocation, the canister also holds assemblies in dry storage at an ISFSI. The ISFSI object acts as a location for assemblies that are neither in the pool or shipped. These assemblies are contained within a canister, which are still contained at the reactor site. Since every canister will hold a group of assemblies, the canister is the basic unit of the ISFIS. The TVM assumes that no repackaging takes place at the reactor site. This assumption means that a loaded canister will never change its internal array of assemblies. The only dynamic part of the canister takes place at the assembly level where the thermal output changes based on the year. It is possible that a canister will have a higher storage heat limit than transportation heat limit resulting in a canister at the ISFSI that cannot be shipped for several more years. The original purpose of the ISFSI was to increase the amount of storage utilities had at reactor sites. When pools were beginning to fill, and it was evident the federal government could not remove the SNF from the sites in time, utilities developed an alternative to the spent fuel pools called the

ISFSI. Most reactor sites have started using ISFSIs. In order to account for the canisters already at an ISFSI, the TVM gave each reactor access to one ISFSI object. The assemblies included in the ISFSI in canisters were not included in the pool, but they were still on the reactor site. To acknowledge this problem, the ISFSI object removes assemblies from the pool array list, but it adds them to a canister in the ISFSI within the reactor object. In addition to dry storage specifications given by the Fuel Projection table, the model has the capability to remove assemblies when the pool is approaching its capacity. In this event, the reactor offloads assemblies into canisters and puts them at an ISFSI. The ISFSI attributes are listed in Table 8.

Table 8: ISFSI Attributes

Attributes	Description
Canisters	An array of canisters that are contained at an ISFSI
Reactor_ID	The reactor identification number

The canisters are the smallest unit at an ISFSI, which explains the array of canisters contained at the ISFSI. Only one ISFSI can be at a reactor site, and the ISFSI should only store SNF from a single reactor or parent reactor. In some (rare) cases, the pools at a reactor site may have sufficient capacity and not need an ISFSI. This results in a reactor without an ISFSI. Building two ISFSIs would be impractical for the TVM. If a reactor site built a separate ISFSI in addition to the already operating ISFSI, the TVM would not change. The ISFSIs at a reactor site are effectively degenerate because the shipments behave the same. Since canisters are the smallest quanta, it is not possible to mix assemblies between canisters. It is also probably reasonable to assume no reactor would deliver its SNF canisters to another reactor's ISFSI after the TVM is

running. This results in each reactor only storing its own SNF or inherited SNF from the Fuel Projection table on site.

Reactors sites comprise the largest scope in the TVM. They own the reactors, the pools in the reactors, the assemblies in pools, the ISFSI, and the canisters at the ISFSI. Using an object-oriented programming approach is particularly useful when using a hierarchy such as the one contained at reactor sites. The reactor site is an object in itself and holds an ISFSI object and an array of reactors, which holds an array of pool objects. The pool objects hold an array of assemblies, so each assembly belongs to a single reactor. The canisters at an ISFSI also belong to a single reactor site. In developing such a rigid hierarchy, it requires creativity to account for some odd operations at reactor sites.

In practice, not all reactors still possess every assembly discharged from their reactor. This creates a problem in tracking the assemblies that permeate through the entire method for optimization. A way to work around this was developed by allowing fictional pools to be created. For example, reactor 1, a PWR, has received SNF from reactor 2, a BWR. Reactor 1 has two real pools; pool 1 and pool 2. The SNF from reactor 2 is moved in pool 2 of reactor 1. Instead of attempting to sort the SNF within the pool, a separate pool can be created that contains only the SNF from reactor 2. This results in less confusion in deciding which canister should be utilized to remove SNF from the pool. Figure 8 provides a description of the previous scenario. The pools are located inside the reactor building, but for the purpose of this description, they exist outside of the reactor.

A list of attributes for the reactor object are provided in Table 9. The reactor object is a key component in minimizing the number of years that reactors keep SNF onsite after reactors

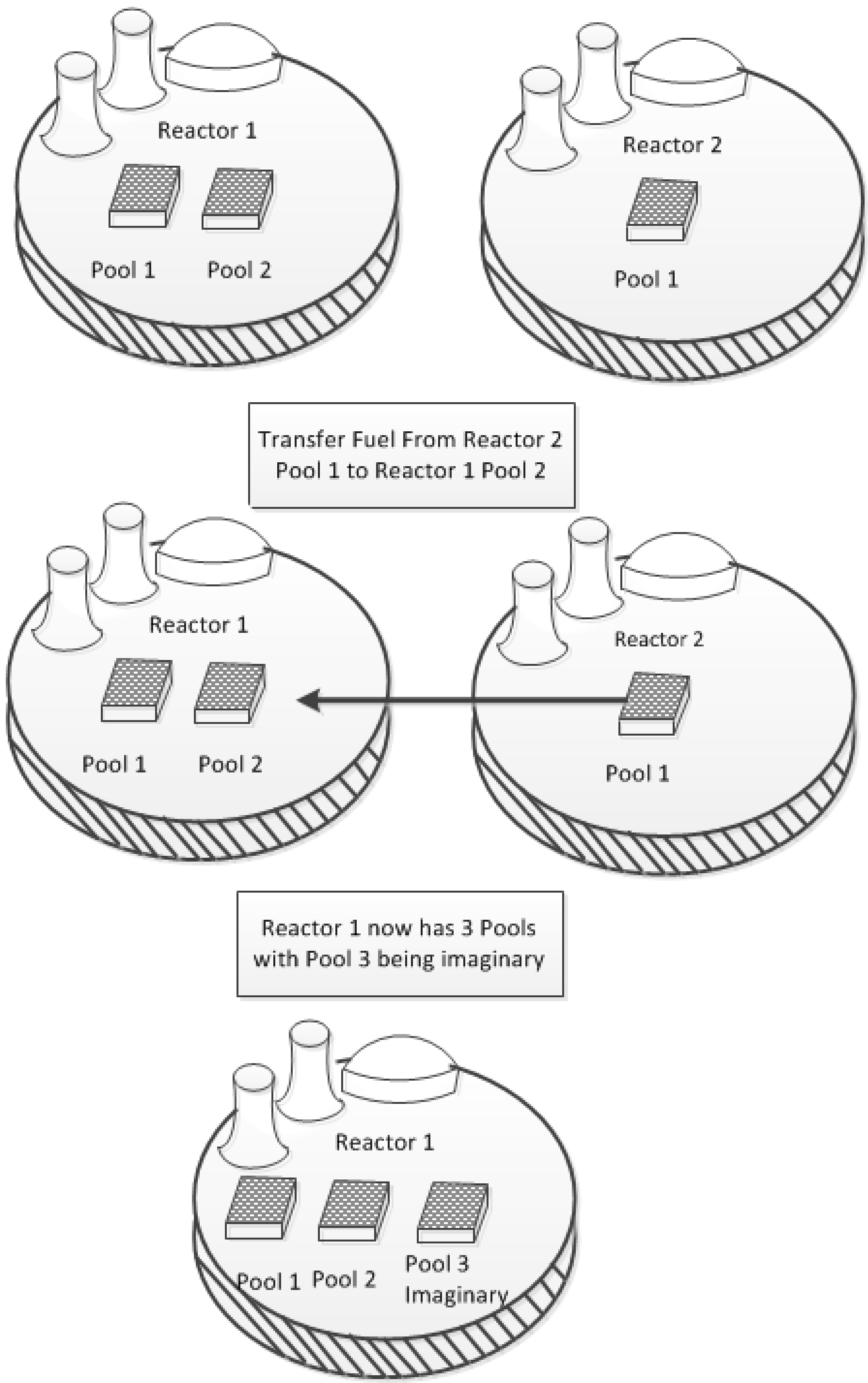

Figure 8: Example Adding Invisible Pool to Reactor Site

Table 9: Reactor Attributes

Attributes	Description
Reactor_ID	Reactor identification number
Shutdown_Date	Year reactor stops producing power
Operating_Limit	The maximum number of canisters a reactor can ship while operating
Shutdown_Limit	The maximum number of canisters a reactor can ship while shutdown
Cans_from_Reactor	Dynamic variable keeping track of how many canister have been shipped from a reactor in a year
Pools	An array of Pool objects
Interim Storage	An ISFSI object
Shippable_Cans	Dynamic variable calculating the maximum number of canisters a reactor can ship

discharge the last assembly into the pool. It keeps up with the shutdown date, the limits, and all the other objects in the model. The assumption a reactor will only possess its own SNF allows for the rigid hierarchy alluded to previously.

In the case where an imaginary pool is created, the reactor with the imaginary pool assumes full ownership of those assemblies. This prevents an assembly from a reactor in Pennsylvania being loaded in the same canister with an assembly from Georgia. The logistics of at reactor transportation do not allow loading from different reactors to happen in a real life scenario either. Some may contend that separating canisters by reactor is not always the case as some geologic repository concepts allow canisters filled with assemblies from different reactor units. These concepts all have a consolidation location, whether it is at a CISF or a repository. Scenarios that deal with fuel blending at a CISF or repository are outside the scope of this evaluation. This evaluation strictly deals with at reactor logistics and allocation strategies from the reactors.

The final object of the hierarchy is the reactor site. The reactor site was included in the TVM to articulate a clearer representation of a real world scenario. If two reactors are collated together, it does not make sense to start counting shutdown years when one shuts down. The cost of operating the reactor counteracts the cost of a shutdown reactor with SNF on site. Some example scenarios do not use the reactor site object, but in order to more accurately model a real time scenario the reactor site object was created. Table 10 gives the attributes to the reactor site object.

Table 10: Reactor Site Attributes

Attributes	Description
Reactor_Site_ID	The reactor site identification number
reactors	An array of reactors contained at the reactor site

The reactor site object acts as a placeholder for the reactors it contains. The only attributes the object has are an identification number and an array of the reactors.

After the objects are created, an allocation needs to be created to compare the different scenarios. The *Allocate_Year_ISFSI* object helps arrange allocation schedules into the output from the TVM. It gives the accepted allocation strategy by both reactor and by year. Table 11 displays a list of attributes for the *Allocate_Year_ISFSI* object. Using the *Allocate_Year_ISFSI* object creates results that give an appropriate amount of information. It is helpful to have the can size and the pool id in addition to the reactor id to get a more accurate representation of the model.

The *Pool_ID* will be zero if the canister was removed from the ISFSI. The results section provides a representation for how the *Allocate_Year_ISFSI* was used.

The reactor removal object represents the order the model uses to remove SNF from reactors. It uses the reactor identification number and the number of assemblies to be removed. Table 12 describes the list of attributes included in the Reactor Removal object.

Table 11: Allocate_Year_ISFSI Attributes

Attributes	Description
Year	The year the allocation occurs
Num_Cans	The number of cans the allocation removes
RX_ID	The reactor identification number
Pool_ID	The pool identification number
Can_Size	The size of the canister in number of assemblies removed for that year

Table 12: Reactor Removal Attributes

Attributes	Description
Pool_ID	The pool or reactor identification number. In many cases the attribute uses the reactor identification number instead of the pool identification number
Assemblies_Removed	The number of assemblies to remove

In most cases, the reactor removal object uses the *reactor_id* instead of the *pool_id*, but it has flexibility where it can be converted to allocate based on pool instead of reactor. The number of assemblies removed depends on the allocation schedule. In an OFF allocation schedule, the reactor removal objects directly mirror the fuel projection table. The fuel projection table gives the number of assemblies and the order in which they came out of the reactor. The reactor removal objects utilize this information to form an allocation strategy. Using different methods

for optimizing the allocation strategy provides different ways to calculate the reactor removal objects.

3.4 Methods of the TVM

A method is similar to a function in that the model calls the method and a task is performed. In many instances, there is an input and an output to the method, but both input and output may be void. In object-oriented programming, methods that are contained within an object's class are "encapsulated". About half of the methods in the TVM are classified as encapsulated methods. They interact with an object in order to change its state.

The setup class contains the first method used by the TVM. This method is not contained within an object's class, so it is not encapsulated. Having a stand–alone class responsible for setting up the model worked well in this instance, because the setup method inputs the tables and organizes them into arrays of objects. This significantly cut down on the time the model took to run to completion but also increased the amount of memory the TVM needed to run. The setup method reads the heat tables for PWR and BWR reactors, the Canister Info table, and the Canister Matching table. The TVM uses the stored variables to create the objects it needs to run the simulation.

The pre-calculated decay heat curves in the BWR and PWR heat tables help determine the thermal characteristics of each assembly based on its initial enrichment, burnup, and present cooling time. The thermal characteristics of a group of assemblies determine if a canister can be loaded or shipped. The limiting factor for the storage and transportation limit are user-determined values for the canister. In order for the assemblies to be loaded or shipped, the sum

of the assemblies' thermal output cannot exceed the canister heat limit. The TVM uses this information in constraining the number of canisters a site can ship in a year. It is possible that a canister will have a higher storage heat limit than transportation heat limit resulting in a canister at the ISFSI that cannot be shipped for several more years.

The first objects the TVM creates is an array of assemblies. It reads in the Fuel Projection table and creates an assembly object for every assembly. If the *DryStorageYear* (Table 5) is less than the current year, then the assembly assumes an *Interim_Storage* value of one. A zero alerts the TVM that the assembly is in the pool, while the one signifies the assembly is loaded into a canister and sitting on the ISFSI. Next, using the stored heat curves input in the setup function combined with the age of the assembly, the burnup of the assembly, and the enrichment of the assembly, the *getThermalOutput* method interpolates the thermal output of the assembly. The heat curves are divided into twelve burnup curves specifying a thermal output for a select number of initial enrichment and cooling time values, derived from the Unified Database [78]. The method then interpolates to find the correct value using the assembly burnup, enrichment, and age. Table 13 provides example data to use the heat tables using a linear interpolation equation. *y1-y18* are index variables representing the thermal output of an assembly and the *x*'s represent the burnup of an assembly. Performing a linear interpolation using burnup and thermal outputs for a fixed enrichment and cooling time yields the thermal output (*y*) of the assembly.

For an assembly located in with an age of 2 years, an enrichment of 2.0%, and a burnup of *x* between burnup 1 (*x1*) and burnup 2 (*x2*), thermal output (*y*) for the assembly (in watts) linearly interpolated from the burn curve data in Table 12 as:

$$\boldsymbol{y = y5 + \frac{(y11 - y5) \times (x - x1)}{x2 - x1}} \qquad \textbf{(3.4.1)}$$

Where *y5* and *y11* are the thermal output for a two-year-old assembly with an enrichment of 2.0% and respective burnups of *x1* and *x2* from Table 13.

Table 13: Example Data for Thermal Calculation

Age of Assembly	Enrichment	Burn 1 (x1)	Burn 2 (x2)	Burn 3 (x3)
1	**1.5**	y1	y7	y13
2	**1.5**	y2	y8	y14
3	**1.5**	y3	y9	y15
1	**2.0**	y4	y10	y16
2	**2.0**	y5	y11	y17
3	**2.0**	y6	y12	y18

After the TVM creates the assemblies, it separates them into their corresponding reactors and pools.

The TVM first separates the assemblies into reactors using the reactor identification tag in the assembly. The reactor then determines if the assembly is in a pool or in dry storage. If the assembly is in dry storage, it groups the assembly with other assemblies tagged for dry storage for that particular year. The method must grab the information from the Canister Matching and Canister Info data to make the correct choice of canister to use when loading the assemblies to be put at the ISFSI. If the assembly is not in dry storage, the method breaks the remaining assemblies up by pool. Some reactors have only one pool, but the TVM has no limit for the number of pools it can create.

The next method creates reactor removal objects in one of two ways. The first is for an OFF allocation strategy. The reactor removals are determined from the Fuel Projection table. Each new batch creates a reactor removal object. If a batch size is one, the number of assemblies to

remove for the reactor removal object will be one. If the batch size is ten-thousand, the number of assemblies to remove for the reactor removal object will be ten-thousand. This allows the TVM to remove canisters in a way that corresponds to the OFF allocation strategy.

The second way the reactor removal objects are determined is with the initial guess method. This method calculates all the assemblies in a reactor and populates the assemblies left with this number. The number of reactor removal objects equals the number of reactors in this method. The allocation strategy can change by reordering the reactor removal objects.

The TVM begins removing canisters in the specified order put forth by the reactor removal objects until it reaches a limit. The limit could be the number of canisters an operating reactor may ship, the number of canisters a shutdown reactor may ship, the number of canisters the system can ship in a year, or the reactor has no more shippable cans. After reaching a limit, it calculates the number of canisters removed from a site for that year. It stores this information using an *Allocate_Year_ISFSI* object. If no more canisters can be shipped for a particular year, the TVM model increases the year, recalculates the assembly thermal output for all assemblies still left at reactor sites and in pools and repeats the removal process. It continues to advance years (one at a time) and remove canisters until no assemblies remain at the reactor sites. The TVM recalculates the thermal output every year because the assembly thermal output decreases with increasing age. The sum of these thermal outputs for a group of assemblies matching the can size for a particular reactor determines how many canisters the reactor may ship each year.

After the model has completed the TVM calculates the number of shutdown reactor years from the array of *Allocate_Year_ISFSI* objects. The shutdown reactor years method takes the sum of the differences between each reactor's last discharge and the last canister removed from the

reactor. To better analyze the results, the method does not count shutdown reactor years prior to 2025, the year the federal government expects to start removing SNF from reactor sites for the example scenario. A reactor that shuts down in 2015 is treated as equivalent to a reactor that shuts down in 1980 in terms of reducing the number of shutdown reactor years. All years prior to the first removal of a canister are nominal values. By removing these years, the difference between changing the allocation strategy to reduce shutdown reactor years is more readily apparent.

3.5 TVM Variables

The TVM has a number of variables that operate as either *static* or *dynamic*. The *static* variables are limits used to curtail the number of canisters from a reactor sites or total number of canisters shipped in a year. The *dynamic* variables change by year or as a new scenario is complete. Table 14 lists the variables used in the TVM.

These variables allow the TVM to run through a varying number of scenarios. The user must change these variables within the model as they are not input in a table form.

Table 14: TVM Variables

Variable	Description
Year	Dynamic variable for the current year
ShutdownYears	Dynamic variable for the number of shutdown years for a scenario
OperatingLimit	Static variable for the maximum number of canisters that an operating reactor can ship
ShutdownLimit	Static variable for the maximum number of canister that a shutdown reactor can ship
YearlyLimit	Static variable for the maximum number of total canisters can be shipped in a year
Number_Assemblies	Dynamic variable for the total number of assemblies left to ship in the model

Chapter Four

Optimization Strategies in the TVM

The TVM utilizes forms of a genetic algorithm, a simulated annealing algorithm, and a mixed integer programming to determine the optimal allocation strategy to reduce the number of shutdown reactor years. The allocation strategies generated using these methods are verified further using a system modelling software package (TSL CALVIN [70]).

4.1 Combinatorial Algorithm

In order to verify that each tested optimization strategy correctly locates the global optimum, a combinatorial algorithm is employed, wherein each possible input permutation is examined within a tractable space. For this limited-scope model, the true optimum can thus be analytically known, allowing for validation of the implementation of each evaluated optimization strategy. Equation 4.1.1 gives the number of scenarios run by the combinatorial method where n is the number of reactors included.

$$\boldsymbol{Combinatorial\ Scenarios = n!} \qquad \textbf{(4.1.1)}$$

This factorial approach is generated by assuming a reactor will attempt to remove the maximum number of canisters from a reactor each year. It also assumes that the allocation order from reactors will remain consistent every year. This assumption removes a number of allocation

strategies that do not attempt to remove the maximum number of canisters in a year, and allocation strategies that do not target specific reactors such as an OFF allocation strategy. These allocation strategies cannot be better than allocation strategies targeting specific reactors with an intention to reduce the number of shutdown reactor years by common logic demonstrated by the following Proof 1.

Proof 1

$$\boldsymbol{Strategy\ 1 = Targeting}$$

$$\boldsymbol{Strategy\ 2 = Non - Targeting}$$

$$\boldsymbol{Minimize \sum_{i=1}^{n} R_i}$$

$$\boldsymbol{R_i = Reactor\ \in \{0, 1\}}$$

$$\boldsymbol{N_i = Canisters\ Left\ at\ Reactor}$$

$$\boldsymbol{R_i = 0\ if\ (N_i = 0)}$$

$$\boldsymbol{else\ R_i = 1}$$

$$\boldsymbol{X \in integer\ \{0 - maximum\ removal\}}$$

$$\boldsymbol{Strategy\ 1 \leq Strategy\ 2}$$

$$\boldsymbol{N_i - \max(X) \leq N_i - (\max(X) - (\max(X) - X))}$$

Strategy 2 cannot reduce the number of shutdown reactor years, because Strategy 2 cannot reduce N_i to zero faster than Strategy 1. Therefore, only allocation strategies that attempt to remove the maximum number of canisters per year are considered, given that they are the only strategies capable of achieving the maximum achievable reduction in the number of shutdown years.

For an example of three reactors (R_1, R_2, R_3) there are six possible queuing combinations:

$$\mathbf{R_1, R_2, R_3}$$

$$\mathbf{R_1, R_3, R_2}$$

$$\mathbf{R_2, R_1, R_3}$$

$$\mathbf{R_2, R_3, R_1}$$

$$\mathbf{R_3, R_1, R_2}$$

$$\mathbf{R_3, R_2, R_1}$$

The Combinatorial Algorithm lists the permutations using a recursive function, which replaces the first element of a list. The TVM inputs the list and calculates the resulting allocation strategy and number of shutdown reactor years.

4.2 Genetic Mutation Algorithm

The TVM optimizes the allocation schedule to minimize the number of shutdown reactor years by using a form of a genetic mutation algorithm. This algorithm has stochastic properties, which allow it to look for a solution that satisfies the constraints of the problem. Similar to the combinatorial algorithm, it only tests allocation strategies that remove as many canisters as possible from a reactor site in a year. Eliminating weaker allocation strategies allow the algorithm to search fewer possible solutions.

The genetic mutation algorithm first creates a population of a size input by the user. The algorithm creates an initial queuing order. It then randomly generates a queuing order by selecting a random element from the population and placing it first and subsequently removes

this element from the initial queuing order. This repeats until no elements remain in the initial queue.

1. $initial\ queuing\ order = \{x_0, x_1, \dots, x_n\}$
2. $x_i = Reactor\ ID = Random\ \in \{initial\ queuing\ order\}$
3. $y_j = x_i$
4. $remove\ x_i\ from\ intial\ queuing\ order$
5. $j = j + 1$
6. $repeat\ steps\ 2 - 5\ until\ inital\ queuing\ order = \emptyset$

The algorithm repeats until the population specified by the user reaches capacity.

After finding an initial population, the algorithm finds the best possible parents by measuring the fitness of the population. The fitness function for this algorithm calculates the number of shutdown reactor years for each member of the population.

The best two performing members of the population take the title of mother and father. The mother is the best performing allocation strategy and the father is the second best performing allocation strategy. The father and mother allocation strategy come together to form a user-specified number of children. The children are allocation strategies formed by looking for differences in the two allocation queues. If a difference is found, there is a 50% chance the father's reactor identification number will be placed in the spot and a 50% chance the mother's reactor identification number will be placed in spot. The reactor identification cannot be reused in the queuing order, resulting in the removal of the reactor identification number from a future queue position. If the reactor identification number is the same for an element, that element will remain the same. Once every element of the queuing order is filled for a child, the process

repeats for the user specified number of children. If two children have the exact same queuing order, the algorithm eliminates one child.

The new set of children undergoes a fitness test where the algorithm picks two new parents. If the fitness does not meet the user's expectations, then the two new parents produce children. The algorithm eliminates the twins, performs the fitness test, and selects the new mother and father. This repeats until the fitness for the mother reaches the user specified goal for number of shutdown reactor years.

1. $intial\ population = \{x_0, x_1, \dots, x_n\}$
2. $fitness\ \{f_0, f_1, \dots, f_n\} = fitness\ test(initial\ population)$
3. $\text{mother} = \min\{f_0, f_1, \dots, f_n\}$
4. $\text{father} = 2\,\text{ndmin}\{f_0, f_1, \dots, f_n\}$
5. $child = mother \times father$
6. $repeat\ step\ 5\ until\ all\ children\ are\ found$
7. $eliminate\ twins$
8. $repeat\ steps\ 2 - 8\ with\ children\ becoming$

   $initial\ population\ until\ goal\ is\ reached$

4.3 Simulated Annealing

The TVM uses a simulated annealing algorithm to find the optimal allocation strategy to minimize the number of shutdown reactor years. This algorithm uses stochastics in order to speed up the process of searching for the optimal solution. It only tests algorithms that target reactor sites for removal. This is to ensure that only the best possible solutions are picked.

The algorithm starts with an initial guess as the queuing order. The initial guess puts the queues of the reactor IDs in numerical order from smallest to largest. This allows for a consistent start point. The algorithm calculates the number of shutdown years from the initial guess to use in terms of reference. Once the reference scenario is stored, a random element of the queuing order swaps with another random element that comes after the first. The random elements are stored as variables so they can swap back or undergo another swap.

The TVM performs the calculation to determine the number of shutdown reactor years and compares it to the stored value. If the value is better, it is accepted and takes the place as the stored value. If the value is equal or worse, it uses equation 4.3.1 to decide which value to store.

$$\boldsymbol{acceptance\ probability = e^{\left(-\left(\frac{100}{n}\right)\times\frac{(SDY-PSDY)}{T}\right)}} \qquad \textbf{(4.3.1)}$$

$$\boldsymbol{n = Total\ Number\ of\ Reactors}$$

$$\boldsymbol{SDY = Calclulated\ Shudown\ Years}$$

$$\boldsymbol{PSDY = Stored\ Value\ for\ Shudown\ Years}$$

$$\boldsymbol{T = Temperature}$$

The Temperature variable starts at a user-specified degree. Every subsequent run, the Temperature lowers based on a formula. As the Temperature lowers, the acceptance probability reduces. For this particular method, equation 4.3.2 defines the temperature.

$$\boldsymbol{T = initial\_T^{(.95\times k)}} \qquad \textbf{(4.3.2)}$$

The *k* is the number of runs so the temperature decreases logarithmically as *k* increases. The *initial_T* represents the initial temperature set forth by the user.

The algorithm either accepts or rejects the new value based on the previous two equations. If the value is accepted, the queuing allocation remains. The first element selected to swap maintains control of the first variable position. It then randomly selects a second element and swaps with it unless there are no elements after the first element. This way of perturbing the queuing order works because the algorithm continues toward an optimal value until it reaches a value that is suboptimal. After one element can move no further, either by reaching the end or failing to be accepted, a new first and second element are chosen. It then repeats itself until the Temperature is less than one degree, given that the Temperature asymptotically approaches zero. The simulated annealing algorithm is presented below.

1. $get\ initial\ guess$
2. calculate shutdown years (PSDY) $using\ TVM$
3. $randomly\ generate\ values\ E1\ and\ E2\ where\ E1\ position < E2\ position$
4. swap the elements [E1, E2] in the inital guess
5. $calculate\ shutdown\ years\ (SDY)\ using\ TVM$
6. $compare\ PSDY\ with\ SDY$
    a. $If\ SDY < PSDY\ then\ SDY\ is\ accepted\ and\ becomes\ the\ new\ PSDY$

    b. $If\ SDY \geq PSDY\ then\ undergoes\ acceptance\ probability\ test$

    $$acceptance\ probability = e^{\left(-\left(\frac{100}{n}\right)\times\frac{(SDY-PSDY)}{T}\right)}$$

        i. $if\ Rand.Num \leq acceptance\ probability\ PSDY = SDY$
        ii. $if\ Rand.Num > acceptance\ probability\ PSDY = PSDY$
7. $calculate\ new\ Temperature\quad T = initial_T{}^{(.95\times k)}$

a. $calculate\ new\ Temperature \quad T = initial_T{}^{(.95\times k)}$

b. $randomly\ generate\ value\ for\ E2\ where\ E1\ position < E2\ position$

c. $repeat\ steps\ 4-8$

8. $if\ PSDY = SDY \rightarrow E1 = E1$

9. $if\ PSDY = PSDY\ repeat\ steps\ 3-8$

10. When Temperature < 1 Stop

4.4 Greedy Algorithm

The greedy algorithm used in the TVM uses a best-fit approach to get a good quick solution. In comparison to the other algorithms, this algorithm does not use shutdown years as a fitness parameter. This approach attempts to find factors that affect the number of shutdown reactor years and use these as part of the fitness function. No stochastic variables are involved in this algorithm, so the solution will always be the same. The greedy algorithm does not use a constant queuing order as the previous algorithms employed. It computes a fitness function after each year to determine the best queuing order based on the shutdown reactor date and number of canisters left at a reactor site for that year.

The greedy algorithm first gets the number of canisters that are shippable from a reactor site and the reactor shutdown date. It then sorts the queuing order from least to greatest for shutdown reactor date and secondly shippable canisters. The queuing order is then converted into reactor removals and run through the TVM simulated SNF removal. This repeats each year until the reactors do not have any more assemblies. The greedy algorithm is presented below:

1. Find number of canisters shippable from reactor and reactor shutdown date

2. Order the reactors first by reactor shutdown date and then shippable canisters
3. Convert queuing order to Reactor Removal objects
4. Simulate SNF removal for one year
5. Increase year
6. Repeat steps 1-5 until no assemblies left at reactor sites

4.5 Integer Programming

The TVM utilizes Gurobi [77] to implement integer programming. Gurobi is a commercial optimization solver specializing in solving linear programs and integer programs. It requires an optimization, equation, bounding constraints, and variables. These inputs create a solution space for which possible solutions may exist. If the solver is unable to find a solution, it returns an infeasible solution. Within the TVM, this primarily means that the problem is not set up correctly.

The optimization equation utilizes canisters shipped from a reactor in a year and binary variables for whether the reactor is shutdown with SNF or not. Equation 4.5.1-4.5.7 represents the integer programming formulation in the TVM.

$$\boldsymbol{min} \qquad \sum_{r \in R} \sum_{i \in T} \boldsymbol{SRY_{ir}} \qquad (\mathbf{4.5.1})$$

$$\boldsymbol{subject\ to} \qquad \sum_{r \in R} \boldsymbol{cans_{ir}} \leq \boldsymbol{yearly\ limit_i} \quad \boldsymbol{for\ i} \in \boldsymbol{T} \qquad (\mathbf{4.5.2})$$

$$\boldsymbol{assem_r} \times \boldsymbol{SRY_{ir}} + \sum_{i \in i-1} (\boldsymbol{cs_{ir}} * \boldsymbol{cans_{ir}}) \geq \boldsymbol{SD_{ir}} \times \boldsymbol{assem_r} \qquad \boldsymbol{for\ i} \in \boldsymbol{T}\ \ \&\ \ \boldsymbol{r} \in \boldsymbol{R} \ \ (\mathbf{4.5.3})$$

$$\sum_{i \in T} \boldsymbol{cs_{ir} \times cans_{ir} \geq assem_r} \quad \boldsymbol{for\ r \in R} \quad (\mathbf{4.5.4})$$

$$\boldsymbol{cs_{ir} \times cans_{ir}} + \sum_{i \in i-1} (\boldsymbol{cs_{ir} \times cans_{ir}}) \leq \boldsymbol{reactor\ limit_{ir}} \quad \boldsymbol{for\ i \in T\ \ \&\ \ r \in R} \quad (\mathbf{4.5.5})$$

$$\mathbf{0} \leq \boldsymbol{canisters_{ir} \leq shutdownlimit_r} \quad \boldsymbol{integral} \quad (\mathbf{4.5.6})$$

$$\mathbf{0} \leq \boldsymbol{SRY_{ir}} \leq \mathbf{1} \qquad\qquad \boldsymbol{integral} \quad (\mathbf{4.5.7})$$

The naming convention is listed below.

- **SRY: Shutdown Reactor Years**
- **cans: number of canisters shipped**
- **cs: size of the canister shipped (number of assemblies inside the can)**
- **assem: total number of assemblies at a reactor**
- **SD: shutdown binary variable 0 if not shutdown 1 if shutdown**
- **reactor limit in assemblies**
- **yearly limit in canister**
- **r:reactor**
- **R: Reactors**
- **i:year**
- **T: Time Horizon**

The objective function in equation 4.5.1 works to minimize the number of years a reactor is shutdown with SNF onsite. Equation 4.5.2 is a constraint for the total number of canisters that

may be shipped in a year. Equation 4.5.3 is a constraint, which determines whether a reactor site is shutdown and has fuel on-site. Equation 4.5.4 is a constraint that ensures all assemblies are shipped from each reactor site. Equation 4.5.5 is a constraint preventing the reactor from shipping canisters that are not shippable.

The variables used by the integer programming solver are number of canisters from a specific reactor site in a year and the binary variable determining if a reactor is shutdown with fuel on-site.

$$\boldsymbol{canisters_{ri}}$$

$$\boldsymbol{SRY_{ri}}$$

The number of total variables to optimize around is the number of reactor sites in the simulation multiplied by the number of years in the simulation. The number of years in the simulation is a user input depending on the acceptance rates and number of reactors in a simulation. This is to provide the user with more flexibility when solving the problem. It also guarantees that enough variables will exist to solve the problem. Often times the reason for the solver to return an infeasible solution stems from not enough variables allocated to solve the problem.

The number of canisters shipped from reactor must be less than or equal to the reactor limit and the number of canisters shipped in a year must be less than or equal to the yearly limit. The estimated number of assemblies removed must be greater than or equal to the total number assemblies at a reactor. This is to ensure that every assembly is removed. The canisters removed must be less than or equal to the sum of all the canisters removed from the reactor in previous years subtracted from than the maximum available canisters to ship from a reactor in a year. A

canister is shippable if the total thermal output for a group of assemblies is less than the transportation limit for the canister.

The TVM finds the maximum available number of canisters to ship by running the TVM without removing any canisters. The result is an array of constants containing the shipping possibilities in a given year for a particular reactor.

4.6 Pareto Curve

The TVM calculates a Pareto curve by adding additional constraints to the integer programming formulation of the problem. The Pareto curve ensures no reactor or utility has more shutdown reactor years after optimizing the allocation strategy than with a traditional allocation strategy. The traditional allocation strategy is an OFF allocation strategy. To optimize using a Pareto Curve, the TVM first simulates the problem using an OFF allocation strategy. The shutdown reactor years are stored for each reactor or utility. Then the integer programming formulation provides the necessary variables and constraints with a few added constraints presented in the equation below.

$$\boldsymbol{SDY_g optimized \leq SDY_g OFF}$$

The number of constraints added by this equation is only the number of reactors (g). Some data may provide an infeasible solution by adding these constraints, but real data should give a feasible solution.

The problem may change to optimize using a Pareto curve on the utilities which own the reactors. The formulation for the constraints change slightly depicted in equation 4.6.1.

$$\sum_{g}^{reactors\ in\ utility} \boldsymbol{SDY_g optimized_u} \leq \sum_{g}^{reactors\ in\ utility} \boldsymbol{SDY_g OFF_u} \qquad \textbf{(4.6.1)}$$

$$\boldsymbol{u = utility}$$

This formulation of the problem allows each utility to perform equal to or better using an optimized allocation strategy than using an OFF allocation strategy.

Chapter Five

Method Validation

The TVM utilizes integer programming, a genetic mutation algorithm, a simulated annealing algorithm, a greedy algorithm, and a combinatorial algorithm to arrive at an optimal allocation strategy for minimizing the number of shutdown reactor years at a site. This chapter contains validation to ensure correct implementation of the methods. It analytically finds the true optimum using the combinatorial algorithm via exploration of all permutations. This particular scenario calculates shutdown reactor years by taking the difference of last shipment and the last discharge from the reactor into the pool.

5.1 Data Analysis

The scenario for the TVM must be small enough to simulate the entire solution space of the problem. The number of possible solutions increases with the number of reactors as a factorial. Including eight reactors in the scenario provided a solution space of 40,320 possible solutions. Some of these solutions may be degenerate depending on the shutdown date and assembly makeup of the reactor. Tables 15-18 give information for the sample validation scenario.

Table 15 shows the top-level breakdown of the sample scenario. It has eight different reactors comprised of three BWRs and five PWRs. The scenario has 10 total pools and 30,252 assemblies where 161 of them start out in dry storage.

Table 15: Sample Data Breakdown

Category (Total)	Quantity
Reactors	8
BWR Reactors	3
PWR Reactors	5
Pools	10
BWR Pools	5
PWR Pools	5
Assemblies	30,252
Batches	2,650
Assemblies in Dry Storage	161
BWR Assemblies	14,435
PWR Assemblies	15,817

Table 16 gives specific information for each reactor. The reactors are numbered in a non-sequential order. Two of the BWR reactors have two pools. The number of assemblies ranges from 526 at reactor six to 7,163 at reactor twelve. The shutdown dates range from 1997 for reactor six to 2046 for reactor seven.

Table 17 provides specific information pertaining to individual pools in the sample scenario. There are an equal number of PWR and BWR pools, and all pools have the same capacity. Pools eleven and fifteen may offload to the ISFSI to stay below the pool capacity of 4,000 assemblies in a pool.

Table 16: Reactor Information Table

Reactor ID	BWR/PWR	# of Pools	# of Assemblies	Canisters Used	Shutdown Date
1	PWR	1	2493	4 Assembly	2034
4	PWR	1	2633	4 Assembly	2036
6	BWR	2	526	9 Assembly	1997
7	PWR	1	3360	4 Assembly	2046
9	BWR	1	6746	9 Assembly	2033
12	BWR	2	7163	9 Assembly	2036
14	PWR	1	3438	4 Assembly	2044
16	PWR	1	3893	4 Assembly	2044

Table 17: Pool Information Table

Pool ID	BWR/PWR	Total # of Assemblies	Associated Reactor ID	Pool Capacity
2	PWR	2493	1	4000
5	PWR	2633	4	4000
9	BWR	441	6	4000
10	PWR	3360	7	4000
11	BWR	6746	9	4000
15	BWR	5715	12	4000
16	PWR	3438	14	4000
17	PWR	3893	16	4000
44	BWR	1448	12	4000
111	BWR	85	6	4000

Table 18 gives the canister information table. The validation scenario only uses canister one and two. The smaller canisters have a greater storage and transportation heat limit per assembly.

Figures 9-13 give a description of the PWR heat curves. The BWR heat curves are not pictured but show similar behavior. Figure 9 is the thermal output for an assembly with an enrichment of 3% and varying burnups over 100 years. For each burnup curve, there is a dramatic drop in the thermal output over the first five years. This is because the elements with the shortest half-lives in the assembly are decaying down to something more stable. Higher discharge burnups consistently produce higher thermal output over the decay cycle for an equivalent assembly enrichment and cooling time.

Figure 10 is the thermal output for an assembly with an enrichment of 3% and varying burnups for ten years. The thermal output decays exponentially after each year. The sharpest slope comes immediately after removal from the reactor. In that first year, an assembly’s thermal output can reduce by more than 3000 Watts.

Table 18: Canister Information Table

Canister ID	Canister Size	Storage Heat Limit (kW)	Transportation Heat Limit (kW)	BWR/PWR
1	4	8	6	1
2	9	8	6	0
3	32	24	24	1
4	68	24	24	0

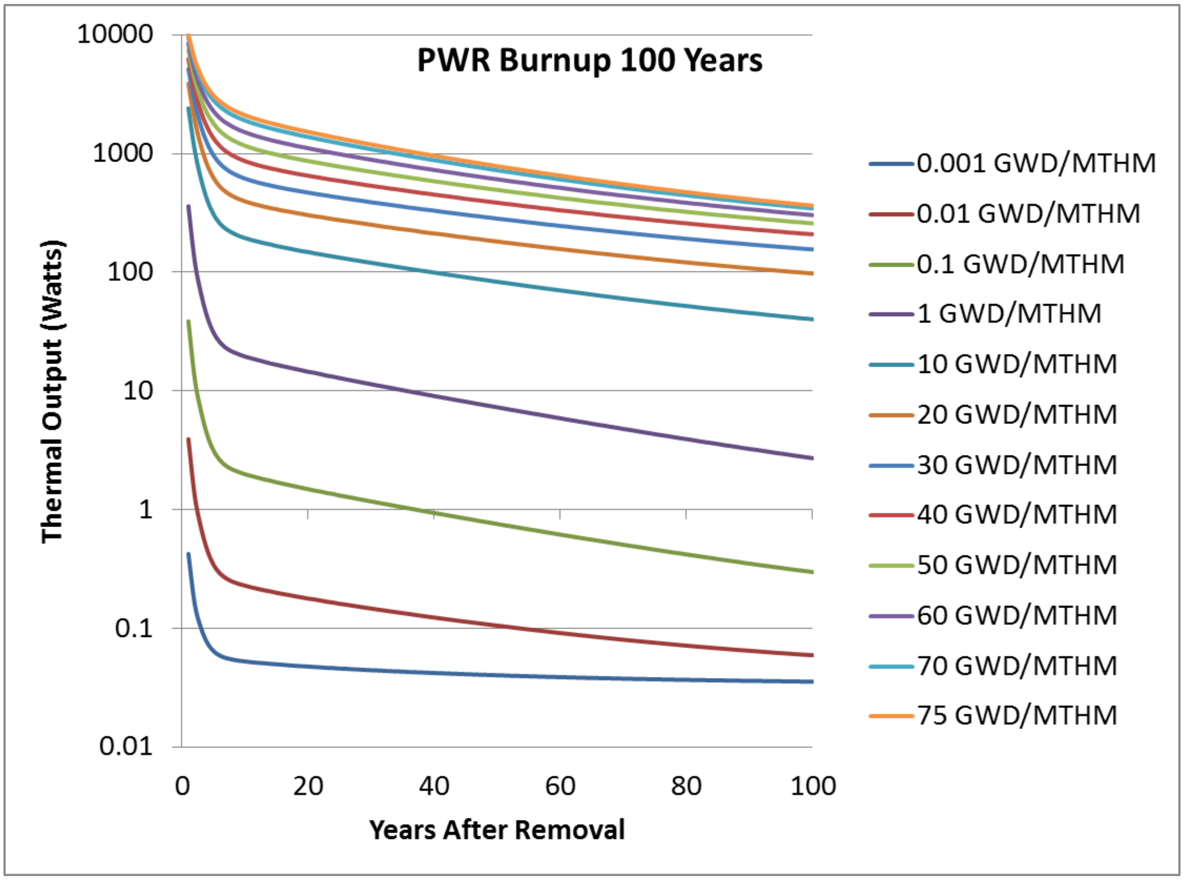

Figure 9: Thermal output as a function of time up to 100 years for a variety of different burnups for a fixed enrichment of 3%

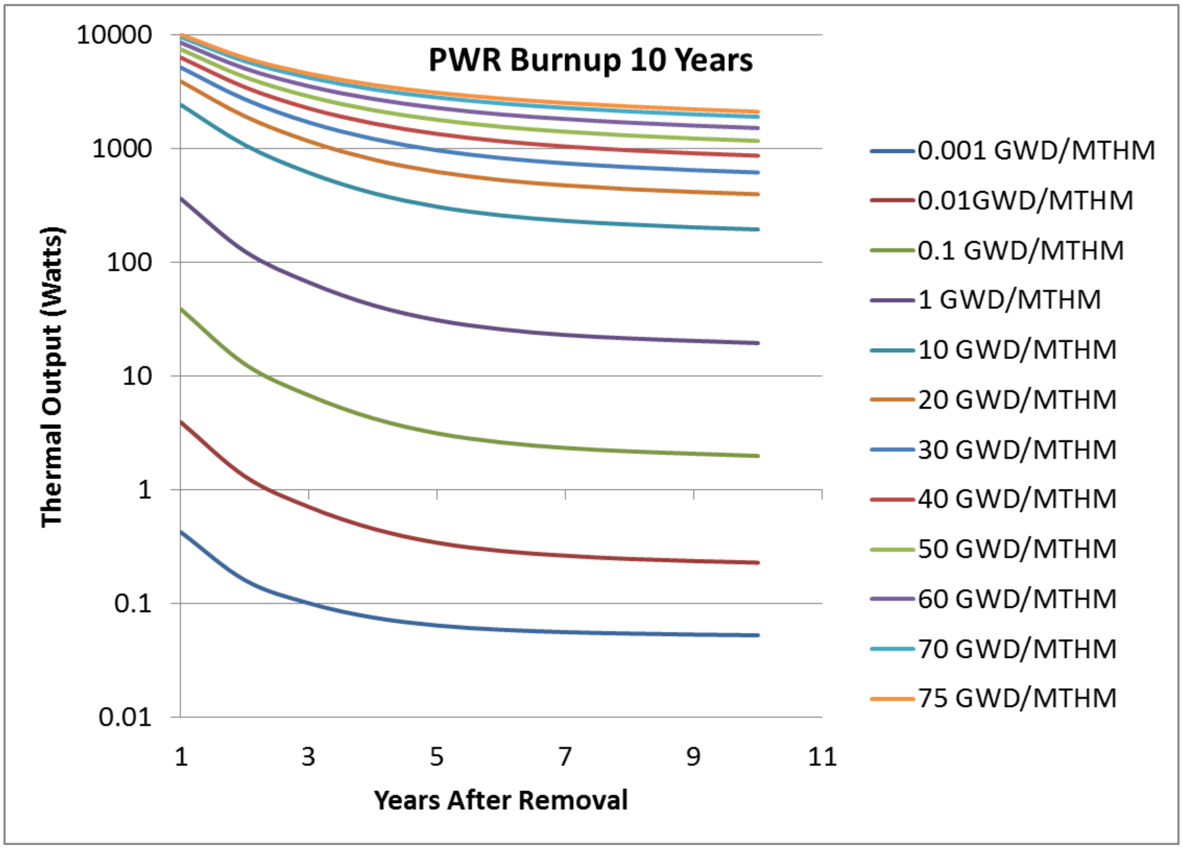

Figure 10: Thermal output as a function of time up to 10 years for a variety of different burnups for an enrichment of 3%

Figure 11 and Figure 12 give the thermal output for 100 years and 10 years for a fixed burnup of 50,000 GWD/MTHM for varying enrichments. The differences are not very significant. The thermal output is inversely proportional to the enrichment. As the enrichment of an assembly increases the thermal output decreases for similar burnup and age.

5.2 Combinatorial Optimization

The combinatorial algorithm analytically finds the true optimum via exploration of all permutations for reactor unloading queuing order. Using this subspace of the solution space (i.e., limiting the search solely to sequential reactor unloading, rather than considering all possible solutions, including non-sequential unloading strategies such as OFF) eliminates many degenerate solutions while also eliminating all solutions that do not specifically target reactors in order to reduce shutdown reactor years. The combinatorial algorithm validates the TVM to ensure the other algorithms and methods are implemented correctly. Figure 13 illustrates the solution space of the sample scenario. The histogram compares the number of times a particular solution exists in the solution space.

The minimum value for the solution space is 215 shutdown reactor years (SRY) while the maximum value is 290 SRY.

This does not model values resulting from a run using the OFF strategy, because this allocation strategy does not specifically target reactors to reduce shutdown reactor years. The combinatorial algorithm contains the complete solution for both the genetic mutation algorithm and the simulated annealing algorithm.

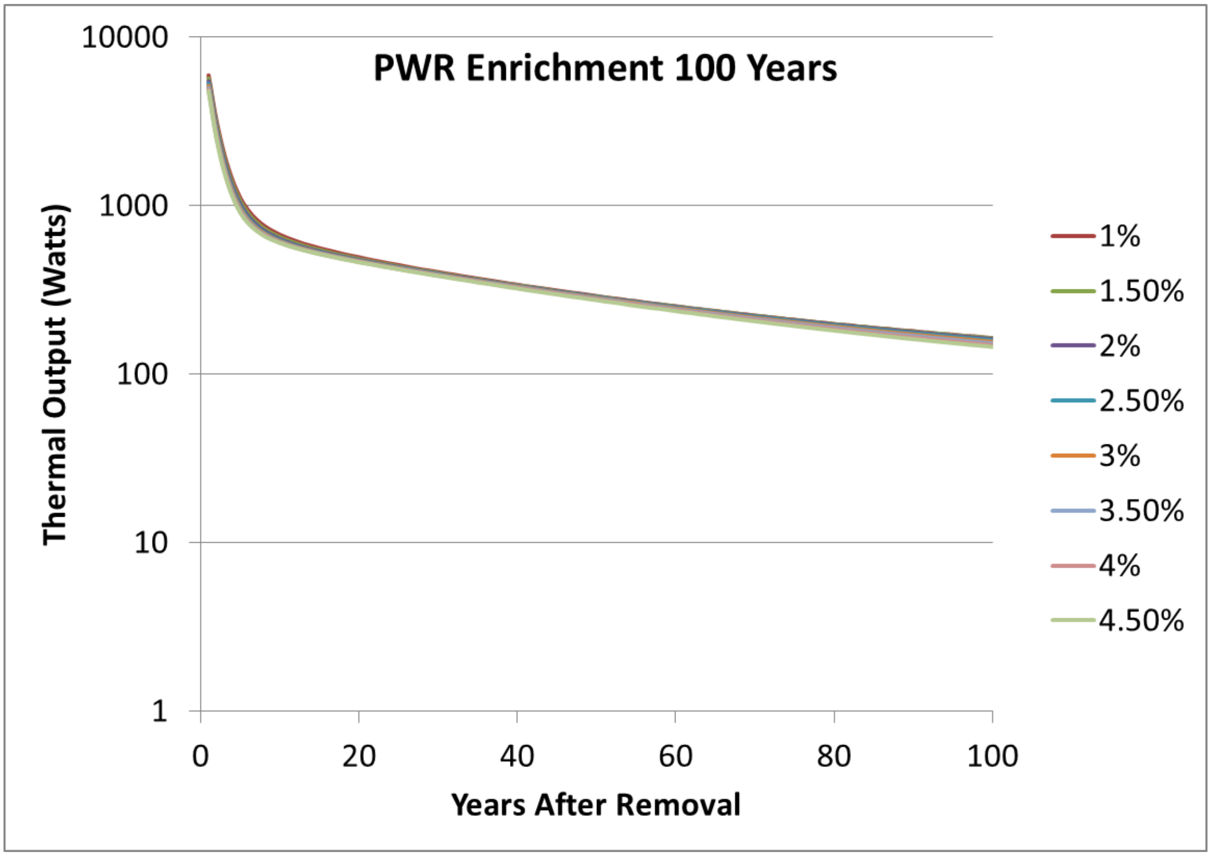

Figure 11: Thermal output as a function of time up to 100 years for a variety of different initial enrichments and a burnup of 50,000 GWd/MTHM

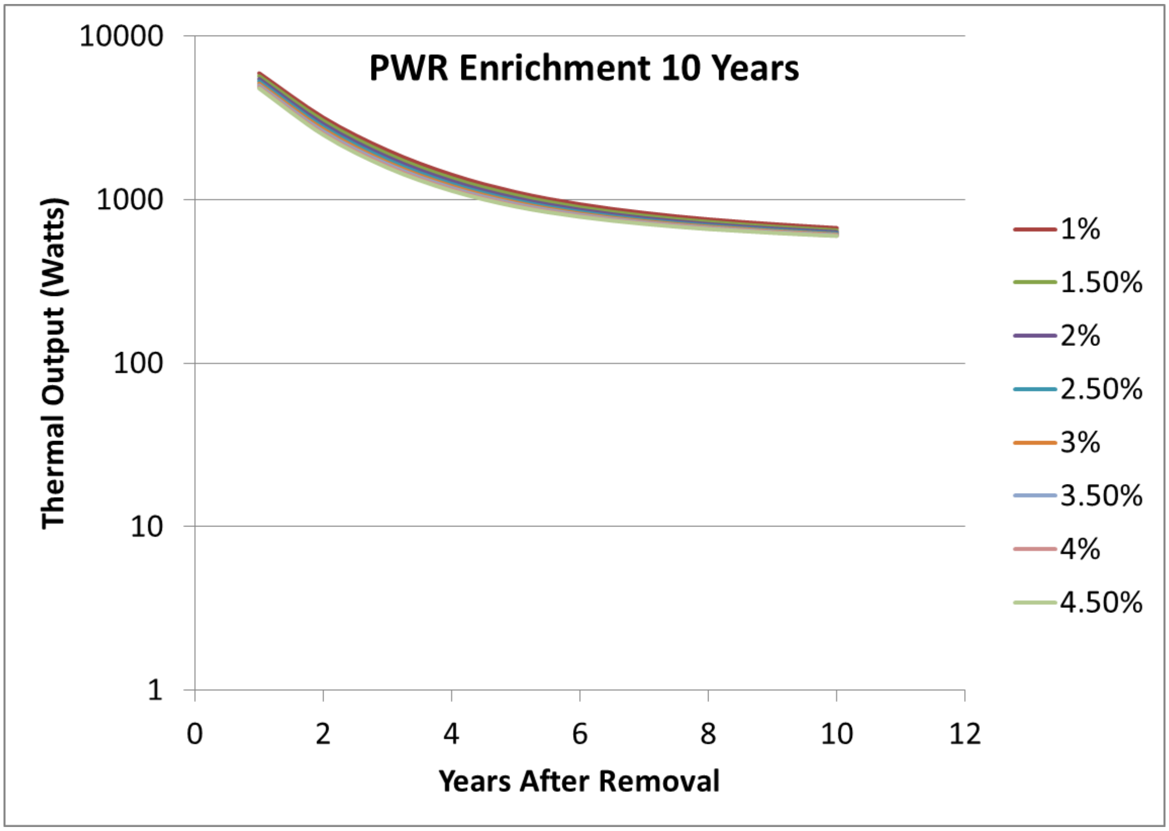

Figure 12: Thermal output as a function of time up to 100 years for a variety of different initial enrichments and a burnup of 50,000 GWd/MTHM

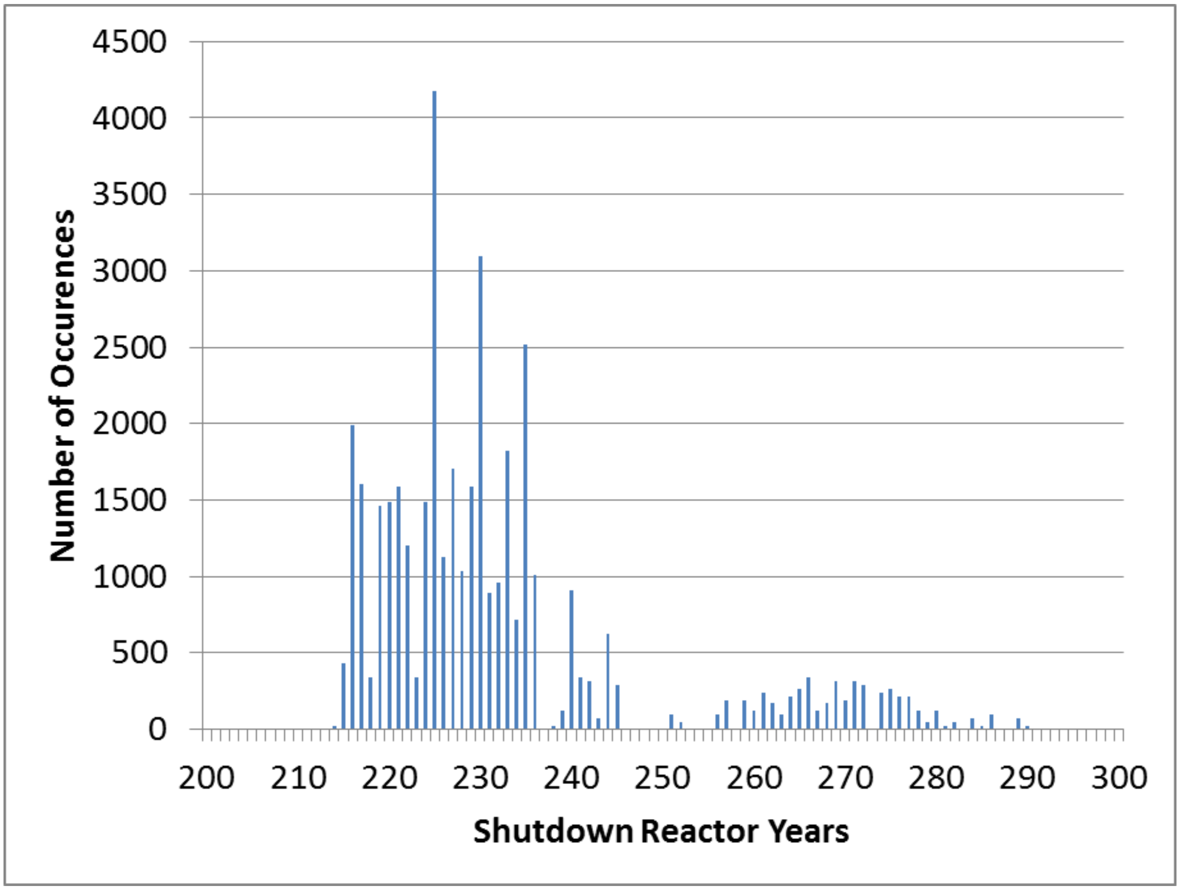

Figure 13: Shutdown Reactor Years for each possible scenario

Out of the 40,320 possible solutions, only 144 solutions managed to achieve the optimum value. This amounts to 0.36% of solution space that obtained the optimum value. The next best value of 216 SRY has 1728 solutions. The simulated annealing and genetic mutation algorithms attempt to find the optimal value in the narrowest part of the solution space. They are hindered by the degeneracy of the exponentially increasing number of "good" solutions or solutions that are near optimal, because an algorithm has a greater chance to be stuck in a local optimum.

The solution space defined by the combinatorial algorithm is queued loading. The maximum number of cans will be removed from a reactor before moving to another reactor. The queue will stay the same each year until all SNF from that reactor is removed. A reactor that has all SNF removed is deleted from the queue. SRY are not minimized by sharing allocation with other reactor sites. It takes a concerted effort to remove all remaining SNF at a reactor site systematically in order to reduce the number of SRY to an optimal value. Eliminating allocation schedules that do not make an effort to remove all SNF from sites significantly reduces the solution space. It also eliminates many degenerate solutions that remove near the limit from a reactor but overall do not affect minimize the SRY. Figure 14 categorizes the solution space by the first reactor chosen in the queue. The y-axis is the number of occurrences and the x-axis contains the number of SRY. These figures give a more in-depth look at where the optimal solution space exists. Table 19 analyzes the solution space represented by the various graphs in Figure 14.

Each solution space looks relatively similar except for solutions that begin the queue with Reactor 6. The similar solution spaces contain a very large number of solutions from 215-245 SRY followed by very few solutions in the range of 245-255 SRY followed by a medium

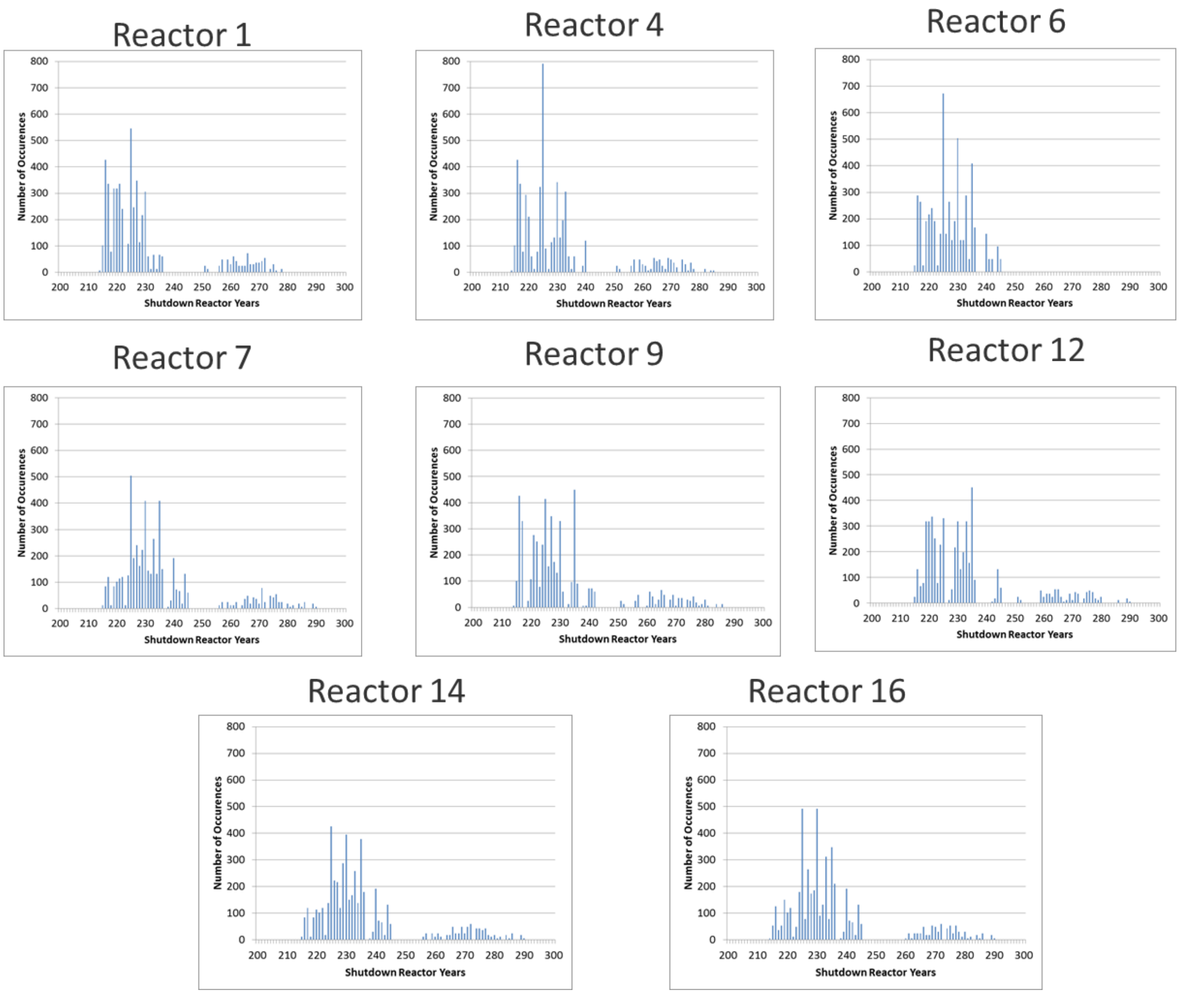

Figure 14: Shutdown Reactor Years for each scenario beginning with a particular reactor in the queue in the queue

Table 19: Minimum and maximum number of SRY for a queueing order and the number of times it occurs for a queuing order beginning with a specific Reactor ID.

Starting Reactor ID for Queue	Minimum Number of SRY	Number of Occurrences at Minimum	Maximum Number of SRY	Number of Occurrences at Maximum
1	215	30	278	12
4	215	30	285	6
6	215	24	245	48
7	216	60	290	6
9	215	30	286	12
12	216	120	290	6
14	216	60	290	6
16	215	30	290	6

.

number of solutions in the range of 255-290 SRY. The solution space beginning with Reactor 6 has no solutions greater than 245 SRY.

This indicates Reactor 6 is the most important reactor in determining the solution space. This will be examined further following an in-depth look into the solution space of the optimal allocation strategies.

Table 19 gives the minimum and maximum number of SRY for a scenario beginning with a specific Reactor ID. It also shows the number of times the maximum and minimum SRY values are realized. For example, for a queueing order beginning with Reactor 1, Reactor 1 always occupies the first position in the queue.

Five of the eight reactors (1, 4, 6, 9, and 16) are capable of starting the queue and still reaching the optimal value of 215 SRY. Starting the queue with three reactors (7, 12, and 14) can only achieve a minimum of 216 SRY. The solution space for four reactors leading the queue (7, 12, 14, and 16) also includes the maximum possible SRY, 290. All other queues have lower maximum possible SRY solutions.

One takeaway from the number of occurrences of a given number of SRY in each queueing scenario is that the value is always a factorial of a number or a multiple of the factorial. For example *factorial(3)=6*, *factorial(4)=24*, *factorial(5)=120*. This stems from the number of degrees of freedom each degenerate solution may have. This gives an insight in how degenerate solutions are formed for the solutions space

Degenerate solutions occur when two or more queue positions can be swapped with the same resulting SRY. Increasing the number of queue positions that can be swapped with each other increases the number of degenerate solutions. Each solution in the whole solution space has at

least six degenerate solutions. This means that five reactors must stay in a constant position, but the three other reactors can occupy any other position.

As previously stated the most interesting and important queue position is for Reactor 6. As seen in Figure 14, this solution space differs from the solution spaces of the other reactors. It has none of the worst solutions over 245 SRY. Reactor 6 is different from all the other reactors in that it shuts down before the any SNF is removed in the simulation. Every year Reactor 6 has SNF on site another SRY is tallied. In all "bad" solutions (over 245 SRY), the TVM cannot unload SNF from Reactor 6. This increases the SRY by at least four as demonstrated by gap the smallest gap between the two humps of solutions seen in all other solutions spaces in Figure 19.

If Reactor 6 cannot remove all its SNF in the first three years of the simulation, it is impossible for an optimal solution to be found. Reactor 6 starts the simulation with a total of 59 shippable canisters which means that 59 canisters can be filled with assemblies and still be under the thermal limit. At Reactor 6, these 59 canisters account for all the SNF. These canisters can be shipped in the first year, but the shutdown reactor limit prevents it from shipping the 59 canisters. A shutdown reactor may ship a maximum number of 25 canisters in a year. So the minimum number of SRY will only be obtained if Reactor 6 is cleared in the first three years.

Since no other reactor is shutdown at this time, each other reactor can only ship the operating limit of 15 canisters in a year. Assuming each reactor can fill the limit in the first three years, an optimal value is not possible with Reactor 6 in the seventh or eighth position. The latest possible position for Reactor 6 must be the sixth position for an optimal. The five previous reactors may remove 15 canisters totaling at 75 canisters, and Reactor 6 can still remove 25 canisters from its

site. In the seventh position, Reactor 6 would ship a maximum of 10 canisters each year resulting in four more SRY. This accounts for the gap in solution spaces between 245 SRY and 255 SRY.

The most important positions in order to obtain the optimal value of SRY are the last two positions. The seventh position must be Reactor 12 and the eighth position must be Reactor 7. Reactor 14 must be in the fifth position when Reactor 6 is in the sixth position otherwise Reactor 14 must be in the sixth position. The other four reactors may occupy positions 1-5 in any order to get an optimal solution.

The total number of optimal solutions can be calculated by the taking the factorial for Reactors 1, 4, 6, 9, and 16 that may be in any of the five positions *factorial(5)=120*. The other 24 solutions require Reactor 6 to be in the seventh position and Reactor 14 to be in the sixth position. This means there are Reactors that can fit in four different positions *factorial (4)=24*. The sum of these two values gives the 144 degenerate solutions making up the optimal solution.

Table 20 shows the shutdown date of each reactor and the total canisters that the reactor must ship.

Combining the data from Table 20 with the optimal allocation strategy provides these insights in achieving an optimal allocation with the sample data:

- The last reactor to shutdown must be the last reactor in the queue
- The first reactor to shutdown must have the maximum number of canisters removed every year
- If the reactor is not the first or last reactor to shutdown, other factors play a large role in determining the optimal value

Table 20: Comparison of reactor shutdown date and the total number of canisters a reactor must ship

Reactor	Shutdown Date	Total Number of Canisters to Ship
1	2034	624
4	2036	659
6	1997	59
7	2046	840
9	2033	750
12	2036	796
14	2044	860
16	2044	974

These insights may not be true for every scenario, but they provide an interesting observation. Queuing reactors by shutdown date or total number of canisters to ship will not produce the optimal allocation strategy in every case.

5.3 Simulated Annealing Validation

This section validates the implementation of the simulated annealing algorithm using the optimum value found in the combinatorial algorithm of 215 SRY. The simulated annealing algorithm uses stochastic variables to find the optimum solution. This means not every run may generate the same solution. The simulated annealing algorithm started with an *initial_Temperature* of 100 degrees. A 100 degrees temperature produces just over ninety iterations with the temperature function the TVM model uses described by equation 4.3.2. The 100 degrees start temperature also allowed the acceptance probability to start over 50% (equation 4.3.1) for a difference in SRY of five years and 30% for a difference of 10 SRY. This allowed a chance for the simulated annealing algorithm to break out of a local minimum to find the optimal value. Figure 15 charts the acceptance probabilities for a difference of 1, 2, 3, 4, 5, and 10 years are charted for a starting temperature of 100 degrees. A higher temperature could make the simulation achieve the optimum value a greater percentage of the time, while a lower temperature could allow the optimum value to be selected at a lower percentage. The higher temperature allows more time to find the optimal solution and a larger initial acceptance probability, but it requires more CPU time. Figures 16 and 17 show SRY as a function of number of iterations and temperature respectively.

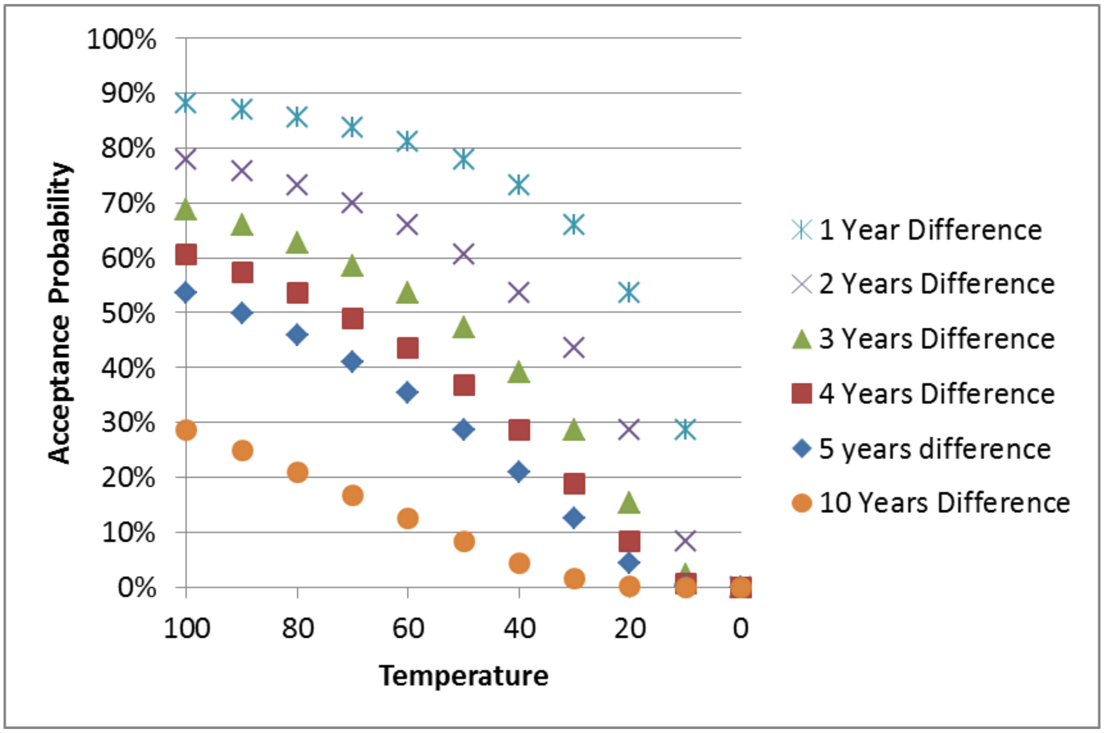

Figure 15: Acceptance probabilities for 1, 2, 3, 4, 5, and 10 years using a simulated annealing algorithm

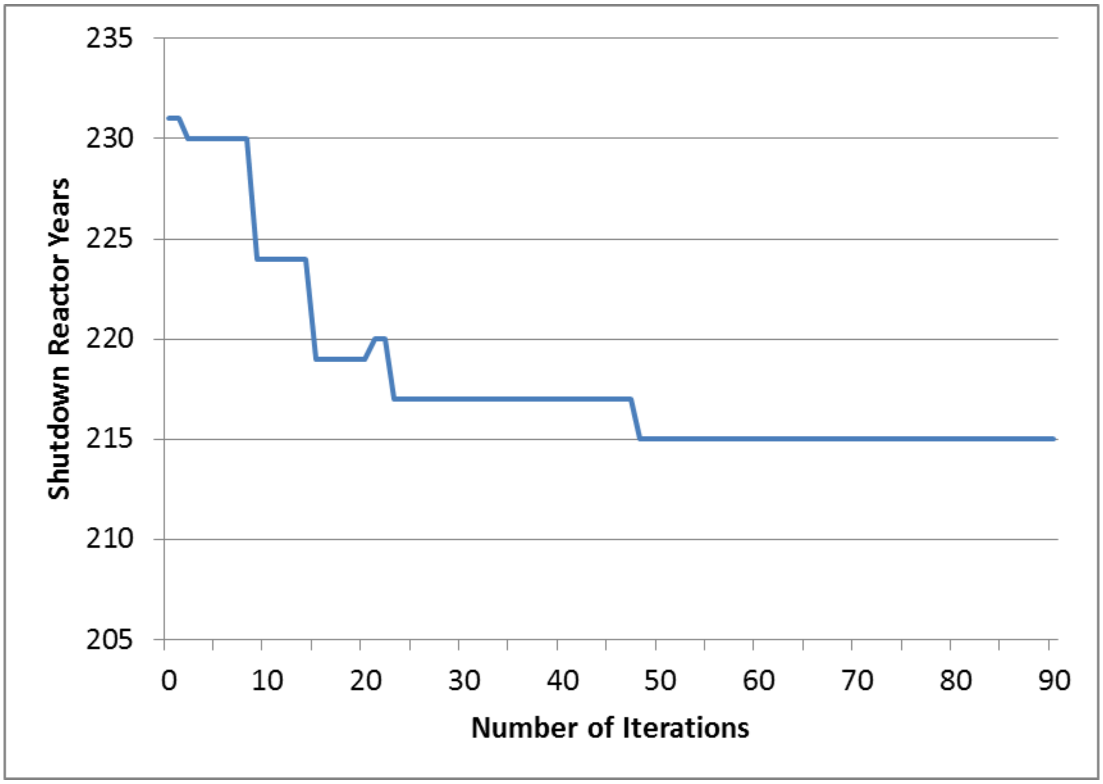

Figure 16: Walk of Simulated Annealing Algorithm by number of iterations

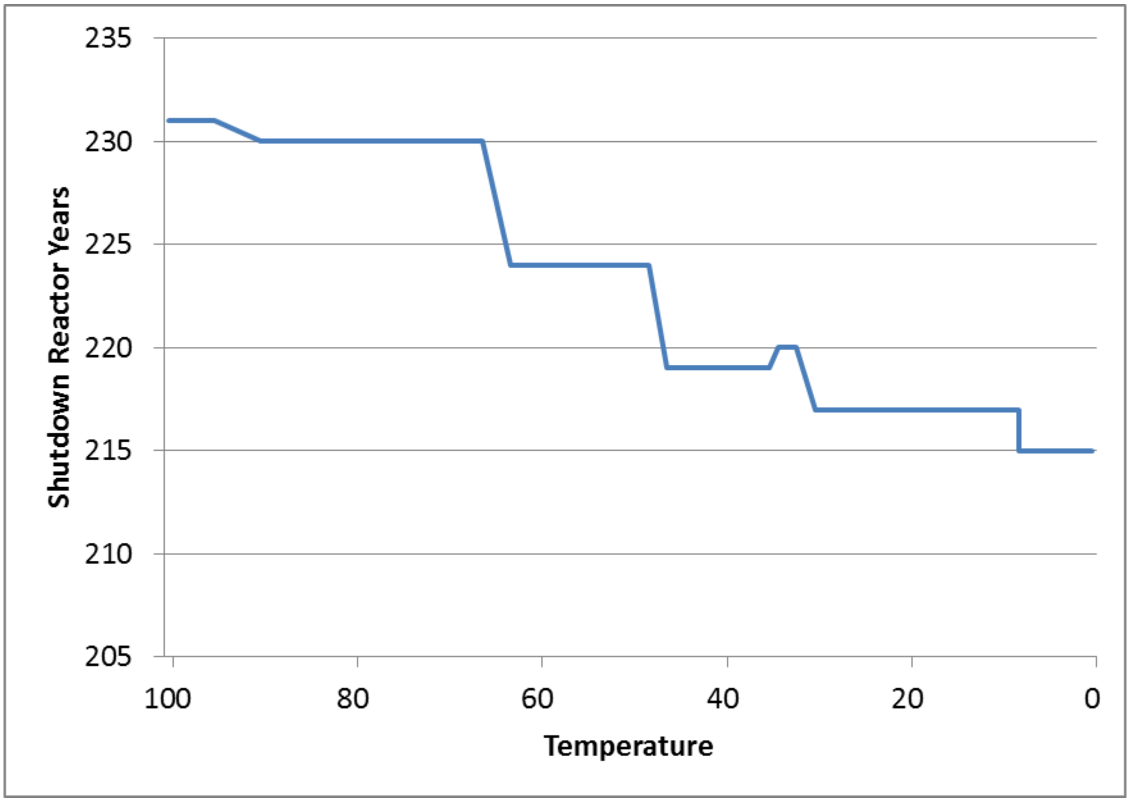

Figure 17: Walk of Simulated Annealing Algorithm by Temperature

The shutdown reactor years start at 218 before dipping down to 216 and back up to 228 maxing out at 229. It then gradually decreases until it settles on 216 shutdown reactor years past the fiftieth iteration.

The temperature distribution in Figure 17 gives a better demonstration of exactly what the algorithm does. At the simulation outset, the temperature decreases rapidly with each iteration, while it decreases very slowly after each iteration near the end. A comparison of the Figure 16 and Figure 17 reveals an increase in SRY at iteration twenty-one and temperature thirty-four. This is the same increase, but it symbolized the final thirty-four degrees has seventy iterations. The first twenty-one iterations reduce the temperature by sixty-six degrees.

The TVM ran the simulated annealing algorithm one-hundred times with an initial temperature of one-hundred degrees. The results are in Figure 18.

The shutdown reactor years for the scenario ranged from 215 to 217 SRY for the one hundred iterations. The simulated annealing algorithm managed to achieve the optimum value of 215 SRY 36% of the time. It achieves a value of 217 or less 100% of the time. These percentages may get better with a higher initial temperature.

5.1 Genetic Mutation Validation

This section validates the implementation of genetic mutation algorithm using the optimum value found in the combinatorial algorithm of 215 shutdown reactor years.

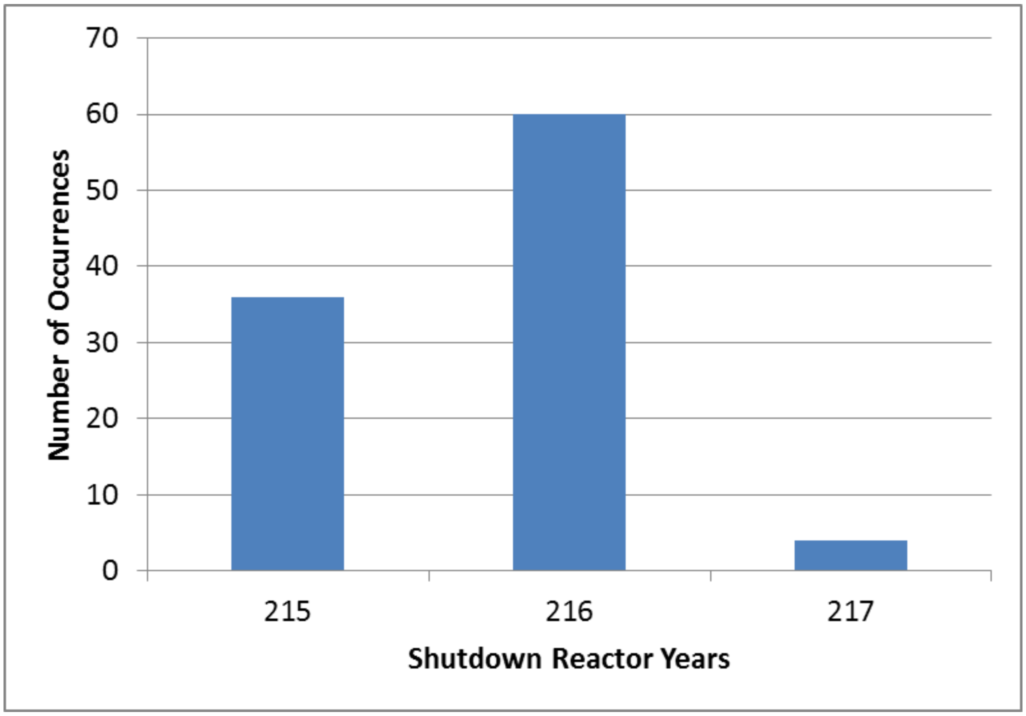

Figure 18: Distribution of shutdown reactor years for the simulated annealing algorithm

The genetic mutation algorithm uses stochastic variables so not every simulation may generate the same solution. Figure 19 shows the distribution for 100 simulations running the genetic mutation algorithm. The initial population was 100 and 10 children were generated from the initial population.

The genetic mutation algorithm succeeds in attaining the optimal value 215 SRY 41% of the time and attains 216 SRY in 58% of the simulations. It achieves a value at or less than 217 SRY 100% of the time. 99% of the simulations are at or below 216 SRY. Increasing the initial population and number of children increases the chances of obtaining an optimal value.

5.2 Integer Programming Validation

This section validates the integer programming formulation using the optimum value found in the combinatorial algorithm of 215 shutdown reactor years. The integer programming formulation is deterministic and does not use the same solution space as the combinatorial algorithm. This solution space is much bigger as it does not follow a queue. This formulation of integer programming will arrive at the same solution for every simulation. The integer programming solution achieved 215 shutdown reactor years. While the simulated annealing and the genetic algorithm both achieve this optimal solution, they both utilize stochastic variables so they do not attain the optimal solution each time. The Integer Programming formulation will achieve this solution 100% of the time. Table 21 compares the dates the reactor empties for an integer programming formulation and a simulated annealing simulation that obtained the optimal value.

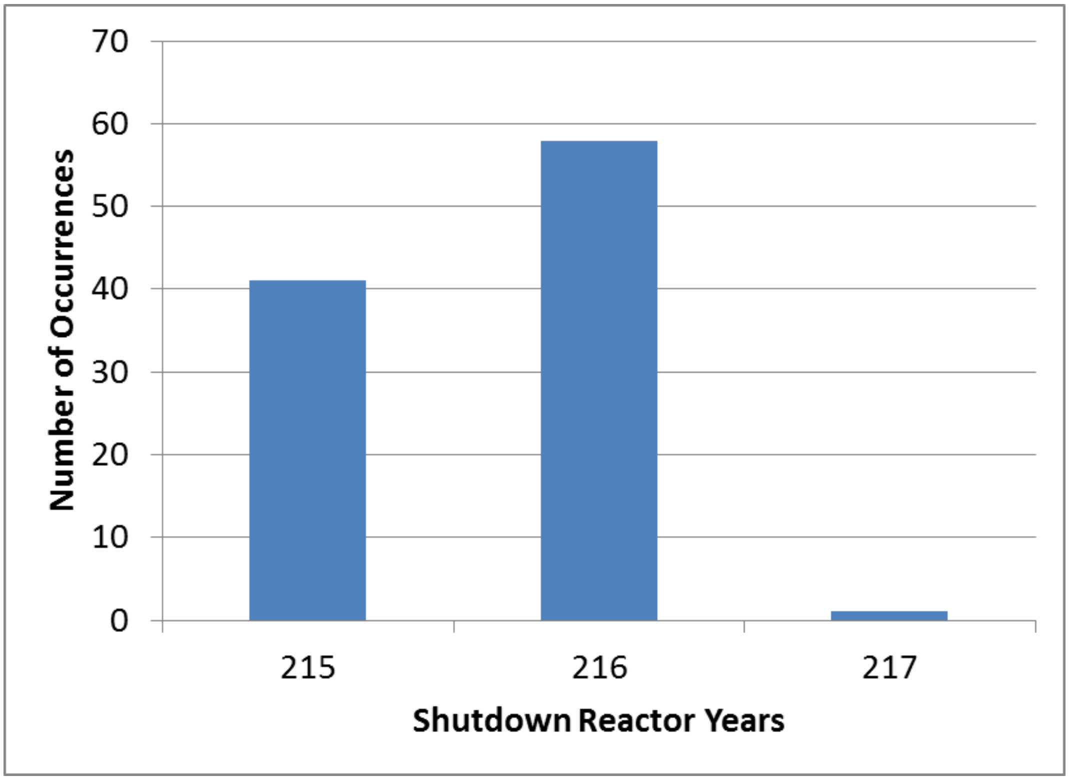

Figure 19: Shutdown Reactor Years generated by the genetic mutation algorithm for an initial population of 100

Table 21: Comparison of reactor shutdown date between integer programming and simulated annealing allocation strategies

Reactor	Integer Programming	Simulated Annealing
1	2053	2053
4	2056	2056
6	2027	2027
7	2080	2089
9	2058	2058
12	2073	2082
14	2080	2077
16	2086	2071
Total SRY	**215**	**215**

5.3 Greedy Algorithm Validation

The greedy algorithm is a heuristic solution to the problem that could be outside of the combinatorial algorithm's solution space. The greedy algorithm determines a new queue every year based on the shutdown year and the number of canisters that can be shipped. The queue could be the same as a combinatorial solution depending on the input data. The greedy algorithm attained a solution of 216 SRY. This technique is by far the least sophisticated and easiest to use. It performs almost as well as the simulated annealing and genetic mutation algorithm, which average just under 216 SRY.

Chapter Six

Results

The TVM validates the removal simulation against previous software “TSL-CALVIN” [72] designed to analyze the entire waste removal system in section 6.1. The TVM compares an OFF allocation strategy to an OFF allocation strategy in TSL-CALVIN. Section 6.2 compares the different optimization techniques and breaks down the differences. Section 6.3 shows results a Pareto formulation tacked onto the Integer Programming formulation. Section 6.4 shows analysis on a full scale scenario with seventy-four reactor sites.

6.1 Comparison to CALVIN

The Used Fuel Disposition Campaign developed a Transportation-Logistics Simulation (TSL) tool that uses the legacy Civilian Radioactive Waste Management System (CRWMS) Analysis and Logistics Visually Interactive model (CALVIN). TSL-CALVIN simulates the logistics and costs of managing SNF across reactors, storage facilities, and disposal facilities. It has the capability to track discharges from a reactor site to a disposal facility and calculate the various costs associated with onsite storage, transportation, interim storage (offsite), and emplacement. The model also provides logistic information relative to the waste stream movement and system resources required to accomplish that movement.

A reference scenario was required to test the compatibility between CALVIN and the TVM. The reference scenario contained the eight reactors and ten pools as described in the previous section. The BWR reactors only used a canister with a nine-assembly capacity and the PWR reactors only used a canister with a four-assembly capacity. The reference scenario only used one canister per reactor type to allow an easier transition between the two models. The two canisters were the smallest for each canister's respective reactor type. Since the TVM implements a limit for the number of canisters leaving a reactor, small canisters require the model to run the simulation longer. The limits for the reference scenario for the TVM were 100 canisters per year, with a maximum of 25 canisters removed annually from a single shutdown reactor and 15 canisters removed annually from an operating reactor. CALVIN had a yearly CISF acceptance limit of 162 MTHM in order to match the 100 canisters shipped per year. CALVIN does not use reactor limits. Table 22 shows the model input comparison for CALVIN and the TVM.

Table 22: Model input comparison

Model	PWR Canister ID	BWR Canister ID	CISF Acceptance Limit	Operating Reactor Limit	Shutdown Reactor Limit
TVM	1	2	100 Canister	15 Canisters	25 Canisters
CALVIN	1	2	162 MTHM	N/A	N/A

CALVIN differs from the TVM in that CALVIN uses metric tons of heavy metal (MTHM) as the unit for establishing acceptance rates and throughput while the TVM uses number of canisters as the baseline unit for acceptance rates and throughput. Canisters contain assemblies, which have masses in MTHM, but not all assemblies have the same mass; similarly, the canister capacity will vary by fuel type. This is illustrated in Table 23. The more realistic limit for throughput will be number of canisters instead of mass, because the number fixed time to load

and unload a canister is a heavier burden on the system than the variable cost of loading assemblies into a canister.

Table 23: Comparing the can size, average assembly weight, and average canister weight used in the example scenario for different reactor types

BWR/PWR	Ref. Can Size	Avg. Weight/Assembly	Avg. Weight/Canister
BWR	9	0.0106 MTHM	0.0957 MTHM
PWR	4	0.0491 MTHM	0.1963 MTHM

Since the assembly weights are not uniform between BWR and PWR types (resulting in possible different canister weights), the comparison to analyze the yearly limit for CALVIN used trial and error. The goal was to allow CALVIN to remove 100 canisters in a year in the same way as the TVM. Allocating 162 MTHM per year allowed Calvin to ship around 100 canisters in a year.

A comparison of the different models for the reference scenario is in the Tables 24 and 25 below. Table 24 removes 100 canisters in a year, while Table 25 removes 45 canisters in a year. Removing 100 canisters a year in this OFF allocation strategy totals one more shutdown reactor year for the TVM than CALVIN. The biggest discrepancy is in reactor 1 where the TVM unloads all the SNF six years earlier than CALVIN. The difference in the CALVIN and the TVM most likely stems from the way canisters are loaded from the allocation strategy. The TVM does not allow semi-loaded canisters to be removed. Instead of shipping a semi-loaded canister, the TVM loads it completely and ships it. The next allocation for that reactor is not affected by the previous reactor removal. Since CALVIN is using MTHM, the next allocation for a reactor site could be affected by a previous removal. Another difference in the TVM and CALVIN is which assemblies get loaded. CALVIN attempts to load the youngest fuel first (hottest) and the TVM

Table 24: Comparing the dates of reactor shutdown between the TVM and CALVIN using OFF and a limit of 100 canisters per year

Reactor	TVM	CALVIN
1	2068	2074
4	2076	2076
6	2036	2037
7	2081	2080
9	2076	2073
12	2077	2076
14	2081	2079
16	2081	2080
Total	**278**	**277**

Table 25: Comparing the dates of reactor shutdown between the TVM and CALVIN using OFF and a limit of 45 canisters per year

Reactor	TVM	CALVIN
1	2133	2133
4	2139	2137
6	2052	2053
7	2148	2147
9	2131	2133
12	2137	2138
14	2145	2145
16	2146	2145
Total	**733**	**733**

loads the coldest fuel first at each respective reactor. Utilities may attempt to remove the YFF, but for the purpose of this study, a coldest fuel first loading strategy is a conservative view.

Removing 45 canisters per year in this OFF allocation strategy totals an equal number of shutdown reactor year for the TVM and CALVIN. At most, the last pickup date for a reactor differs by two years.

Constraining the number of canisters that can be removed in a year greatly increases the number of shutdown reactor years. It also provides more of an opportunity to optimize the allocation strategy. The less SNF removed in a year, the more sensitive the allocation strategy becomes. As the yearly limit increases to the sum of the reactor limits, the optimization impact gets smaller until it reaches zero. In order for an optimized allocation strategy exist, the inequality expressed in equation 6.1.1 must be true, i.e., yearly limit must be less than the minimal sum of all reactor limits in a year.

$$\boldsymbol{YearlyLimit < min \sum Reactor\_Limits} \tag{6.1.1}$$

6.2 Comparison of Different Optimization Techniques

As seen in the previous section, an OFF allocation strategy for the eight-reactor test case with an annual limit of 100 canisters results in 281 shutdown reactor years. The combinatorial algorithm (assuming sequentially queued reactor unloading) gives a solution space ranging 215 to 290 shutdown reactor years, with the optimized allocation strategies ranging from 215 to 217 shutdown reactor years. The OFF allocation strategy at 278 shutdown reactor years falls in the top 3% worst performing allocation strategies within the combinatorial algorithm (2.7%). Almost

any alternative allocation strategy that follows a queue will outperform an OFF allocation strategy in terms of reducing the number of shutdown years at reactor sites. The OFF allocation strategy was decided on in the Standard Contract to maintain a fair and predictable way to remove SNF. In section 6.3, the scenario is formulated into a Pareto optimization problem so that no utility will have more total SRY employing a different allocation strategy than using an OFF allocation strategy.

The optimization strategies performed on the scenario significantly reduced the number of shutdown reactor years. The simulated annealing and genetic mutation algorithms employed stochastic variables to develop a queueing strategy to determine the optimal allocation strategy. The queue that these strategies developed was contained within the solution space generated by the combinatorial algorithm. The integer programming formulation does not use the same solution space that the combinatorial algorithm utilizes. It utilizes a solution space not limited to a queuing strategy. It also does not have to send the maximum canisters shippable from a reactor site in a given year. The greedy algorithm does not use the same queuing solution space, but it is highly likely it will develop into a queue. The greedy algorithm sorts by shutdown years and number of canisters to ship each year. The shutdown years are constant and if SNF comes online in similar capacity at different reactor sites, the queue will not change from year to year.

Table 26 compares the results of the different optimization strategies for SRY of 215, 216, and 217. The Integer Programming formulation performs the best followed by the genetic mutation algorithm, simulated annealing, greedy and combinatorial algorithm. The simulated annealing and genetic mutation algorithms may have performed better in determining the optimal value, if the number of degenerate solutions was not as large.

Table 26: Comparison of different optimization methods

SRY	Simulated Annealing	Genetic Mutation	Integer Programming	Greedy	Combinatorial
215	36%	41%	100%	0%	0.4%
216	60%	58%	0%	100%	4.2%
217	4%	1%	0%	0%	3.9%

6.3 Pareto Formulation

The TVM calculates a Pareto curve by adding additional constraints to the integer programming formulation of the problem. The Pareto curve ensures no reactor or utility has more shutdown reactor years after optimizing the allocation strategy than with a traditional OFF allocation strategy. Table 27 breaks down the reactor sites into three utilities. The different colors symbolize different utilities. The table gives the date of shutdown and the OFF last reactor removal date.

The reactors are sorted into three groups consisting of the first three PWRs in purple (Utility A), the two BWRs in Green (Utility B), and the last three PWRs in Orange (Utility C). Utility A has a total of 109 SRY, Utility B has a total of 50 SRY, and Utility C has a total of 117 SRY. Table 28 shows the year of shutdown for a Pareto formulation of the scenario. Table 29 shows a utility comparison between the Pareto formulation and OFF allocation strategy.

Table 27: Reactor shutdown date and last canister removal using OFF allocation strategy

Reactor	Shutdown Date	OFF Removal Date	SRY
1	2034	2068	34
4	2036	2076	40
6	1997 (2025)	2036	11
7	2046	2081	35
9	2033	2076	43
12	2036	2077	41
14	2044	2081	37
16	2044	2081	37

Table 28: Reactor shutdown date and last canister removal using Pareto formulation

Reactor	Shutdown Date	Pareto Removal Date	SRY
1	2034	2053	19
4	2036	2056	20
6	1997 (2025)	2027	2
7	2046	2080	34
9	2033	2058	25
12	2036	2073	37
14	2044	2080	36
16	2044	2086	42

Table 29: Comparison between OFF and Pareto formulation for SRY

Utility	OFF (SRY)	Pareto (SRY)
A	109	73
B	52	39
C	117	103
Total	278	215

The Pareto value was able to achieve an optimal value as well. This shows that the best strategy may work for stakeholders. Utilizing the Pareto curve with different scenarios allows users to determine strategies that do not make anyone worse; however, one could alternatively find strategies that evenly distribute the number of SRY between the reactors.

Table 30 gives an alternative utility plan with utility A highlighted by purple, utility B highlighted by green, utility C highlighted in blue, and utility D highlighted in orange. Once again, the two BWR reactors are grouped together.

Table 31 gives the shutdown date and last canister removal for the Pareto formulation of the problem and Table 32 gives a comparison of SRY for the four separate utilities.

Once again, the Pareto value was able to achieve an optimal value. Even though the formulation may change, it is still possible to achieve an optimal value for SRY.

Table 30: Reactor shutdown date and last canister removal using OFF allocation strategy

Reactor	Shutdown Date	OFF Removal Date	SRY
1	2034	2068	34
4	2036	2076	40
6	1997 (2025)	2036	11
7	2046	2081	35
9	2033	2076	43
12	2036	2077	41
14	2044	2081	37
16	2044	2081	37

Table 31: Reactor shutdown date and last canister removal using Pareto formulation

Reactor	Shutdown Date	Pareto Removal Date	SRY
1	2034	2053	19
4	2036	2056	20
6	1997 (2025)	2027	2
7	2046	2081	35
9	2033	2058	25
12	2036	2079	43
14	2044	2081	37
16	2044	2078	34

Table 32: Comparison between OFF and Pareto formulation for SRY

Utility	OFF (SRY)	IP (SRY)	Pareto (SRY)
A	74	39	39
B	52	39	45
C	78	59	60
D	74	78	71
Total	278	215	215

6.4 Full Scale Analysis

The TVM is capable of determining the optimal allocation for an entire reactor sized fleet. The data used in the full scale analysis is in Table 33.

Because the Integer Programming formulation gave the optimal value with no variation, it was used to determine the optimal allocation strategy for a full scenario. The full scenario used canisters with a capacity of 32 PWR assemblies and 68 BWR assemblies contained in Table 18 to more accurately model the canisters used at current reactor sites. The yearly limit was 3,000 MTHM for CALVIN equating to 225 canisters in the TVM. Variable assumptions for the full scale scenario are in Table 34.

Table 33: Full-scale data breakdown

Category (Total)	Quantity
Reactors Sites	74
BWR Reactors	44
PWR Reactors	82
Pools	126
Assemblies	459,508
Batches	253,737
Assemblies in Dry Storage	441

Table 34: Assumptions for full reactor scenario

	OFF	IP
PWR Canister	32 Assembly Size	43 Assembly Size
BWR Canister	68 Assembly Size	68 Assembly Size
Yearly Limit	3,000 MTHM	225 canisters
Shutdown Reactor Limit	N/A	25 canisters
Operating Reactor Limit	N/A	15 canisters

The full scenario using an OFF allocation strategy resulted in 1554 SRY. The optimized allocation strategy resulted in 532 SRY (within 1.02% of the minimum LP solution). The optimized allocation strategy accounted for an almost 300% decrease.

Table 35 gives the seventy-four reactors removal dates for the OFF and optimized allocation strategy. Figure 20 illustrates the comparison of number of SRY for each reactor site between OFF and the optimized solution for each.

Four reactors had more SRY in the optimized allocation strategy than the OFF allocation strategy. The optimized allocation strategy most negatively affected Reactor 45 adding 7 SRY to the OFF allocation strategy. Although the total number of SRY significantly decreased using the optimal allocation strategy, the worst case reactor was only marginally affected.

Table 36 compares the full scale scenario with the sample scenario. The ratio between the maximum number of canisters that can ship from all reactors and the throughput limit at the CISF is an indicator for how well an optimized allocation strategy will perform. If this ratio

Table 35: Reactor shutdown dates for OFF and the optimized allocation strategy

Reactor Site	OFF	IP	Reactor Site	OFF	IP
Reactor 1	2061	2043	**Reactor 38**	2068	2050
Reactor 2	2071	2052	**Reactor 39**	2056	2039
Reactor 3	2032	2021	**Reactor 40**	2065	2058
Reactor 4	2065	2053	**Reactor 41**	2062	2045
Reactor 5	2060	2044	**Reactor 42**	2059	2039
Reactor 6	2060	2045	**Reactor 43**	2049	2027
Reactor 7	2065	2058	**Reactor 44**	2057	2036
Reactor 8	2064	2049	**Reactor 45**	2065	2072
Reactor 9	2060	2041	**Reactor 46**	2059	2048
Reactor 10	2066	2048	**Reactor 47**	2064	2057
Reactor 11	2047	2026	**Reactor 48**	2048	2031
Reactor 12	2069	2064	**Reactor 49**	2058	2038
Reactor 13	2061	2042	**Reactor 50**	2059	2039
Reactor 14	2058	2047	**Reactor 51**	2048	2027
Reactor 15	2041	2022	**Reactor 52**	2027	2021
Reactor 16	2061	2042	**Reactor 53**	2056	2035
Reactor 17	2053	2030	**Reactor 54**	2064	2053
Reactor 18	2056	2040	**Reactor 55**	2062	2045
Reactor 19	2059	2041	**Reactor 56**	2043	2026
Reactor 20	2064	2061	**Reactor 57**	2067	2055
Reactor 21	2062	2046	**Reactor 58**	2062	2046
Reactor 22	2047	2027	**Reactor 59**	2083	2057
Reactor 23	2046	2022	**Reactor 60**	2063	2048
Reactor 24	2055	2034	**Reactor 61**	2063	2047
Reactor 25	2064	2055	**Reactor 62**	2057	2038
Reactor 26	2032	2022	**Reactor 63**	2064	2069
Reactor 27	2065	2051	**Reactor 64**	2029	2021
Reactor 28	2061	2049	**Reactor 65**	2058	2038
Reactor 29	2064	2061	**Reactor 66**	2066	2064
Reactor 30	2021	2021	**Reactor 67**	2045	2024
Reactor 31	2060	2040	**Reactor 68**	2063	2056
Reactor 32	2044	2022	**Reactor 69**	2064	2049
Reactor 33	2025	2021	**Reactor 70**	2073	2060
Reactor 34	2063	2065	**Reactor 71**	2064	2050
Reactor 35	2065	2066	**Reactor 72**	2028	2021
Reactor 36	2032	2022	**Reactor 73**	2032	2023
Reactor 37	2068	2048	**Reactor 74**	2059	2039
Total SRY				**1554**	**532**

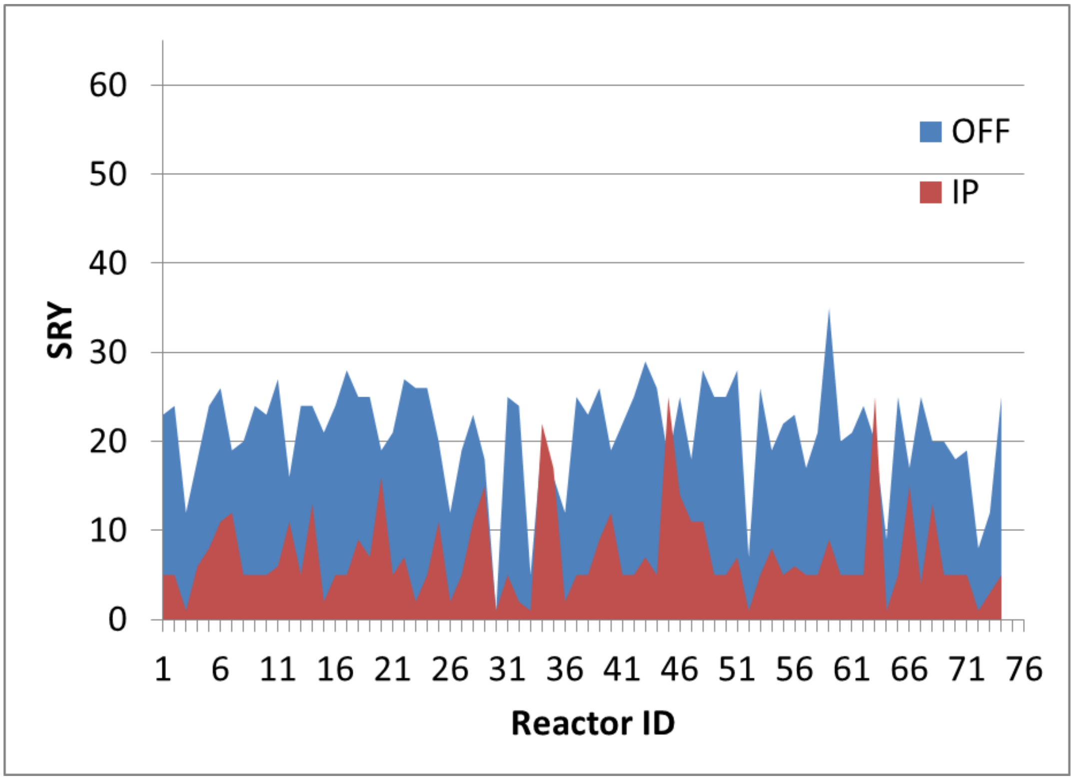

Figure 20: Comparison of SRY between optimized and OFF allocation strategy for each reactor site in the full scenario

Table 36: Comparison of sample scenario with full scenario

Parameter	Full Scale	Sample
CISF Acceptance Rate	225 canisters	100 canisters
Maximum Canisters in a Year	74x25=1850 canisters	8x25=200 canisters
Acceptance Rate/ Max Canisters	225/1850=0.12	100/200=0.50
Optimal Value (SRY)	584	215
OFF (SRY)	1554	277
Optimal Value/OFF	0.38	0.78

reaches is greater than or equal to one, then there will not be an optimized allocation strategy better than OFF.

This results from the OFF strategy and the optimized strategy shipping all available canisters from reactor sites, therefore both strategies will end up with the same allocation strategy.

The ratio of CISF throughput to the maximum number of shippable canisters at all reactors was less for the full scale scenario, which allowed the optimal allocation strategy on the full scale scenario to have more of an impact than on the eight reactor sample scenario. As the ratio of CISF throughput to maximum number of shippable canisters decreases the allocation, strategy is more sensitive to changing the number of SRY.

Chapter Seven

Conclusions

7.1 Summary

Commercial nuclear power plants produce long-lasting nuclear waste, primarily in the form of SNF assemblies. SFP and canisters or casks that sit at an ISFSI at the reactor site store the fuel assemblies that are removed from operating reactors. The federal government has developed a plan to move the SNF from reactor sites to a CISF or a geological repository. In order to develop a predictable pick-up schedule and give utilities notice of an impending pickup from a reactor site, the federal government developed a queuing strategy based on OFF. The OFF allocation strategy allows the federal government to remove SNF from reactor sites in the same order the assemblies came out of the reactor. While this approach may result in a fair approach, it is far from the most cost-effective approach.

The problem with accepting SNF using an OFF algorithm is that a handful of sites are no longer producing power and exist only to store the SNF they produced. This is an expensive process, which results in an annual cost of ~$8M [22]. Utilizing different algorithms to reduce the amount of time these shutdown reactors keep SNF on site may reduce the total system costs for the federal government.

The TVM simulates removing SNF from reactor sites to demonstrate the effectiveness of different algorithms in reducing the total number of shutdown years incurred by the system. The

goal of the TVM is to validate the implementation of the optimization algorithms on a problem space small enough such that the true optimum is analytically known via exploration of all permutations (via a combinatorial algorithm). By validating the optimization algorithms against a space where the solution can be analytically known, they can then be applied to larger, more representative systems where the number of permutations is too large for a combinatorial algorithm to effectively process. This provides a true optimal solution as a baseline for the other algorithms to achieve.

The TVM utilizes integer programming, a genetic mutation algorithm, a simulated annealing algorithm, a greedy algorithm, and a combinatorial algorithm to arrive at an optimal allocation strategy for minimizing the number of shutdown reactor years at a site. The TVM calculates SRY by taking the difference of last shipment and the last discharge from the reactor into the pool.

7.2 Key Points

The combinatorial algorithm provides the solution space of a scenario, which can lead to some generalization concerning all optimal allocation strategies. This particular scenario showed that the oldest shutdown reactor must remove as much SNF as possible to reach an optimal value. It also had the reactor with the shutdown date farthest in the future positioned last in the queue for every optimal allocation strategy. In between the first and the last queue position, no particular pattern stood out. The variables for reactor shutdown date and total canisters to ship were not helpful in determining rest of the queue. The combinatorial solution space also provided a

backdrop to view the OFF allocation strategy. The OFF allocation strategy performed in the bottom 10% of all queued solutions in reducing the number of shutdown years.

The optimization algorithms worked very well to get a “good” solution, but did not find always find the optimal solution with the exception of integer programming. They would routinely break into the top percentage of all solutions, but getting an optimal solution was difficult. The genetic mutation algorithm performed a bit better than the simulated annealing algorithm, reaching the optimal value 41% compared to 38%. The Integer programming formulation calculates the optimal solution each time. The greedy algorithm returns a value one SRY more than the optimal value. This is still a good solution considering the lack of complexity built into the algorithm scoring in the top 4.6% of the available solution set.

The Pareto formulation proved that an optimal solution could reduce each stakeholder’s costs as well. This is a key point because it allows a clear incentive for all parties to change the allocation strategy from OFF to one that will benefit everyone. When SNF is ready to move, more possibilities for site removal will exist than just OFF.

The allocation is significantly more sensitive for a smaller throughput to the CISF. If the yearly limit is greater than the sum of all reactor limits, the OFF allocation strategy will be equivalent to an optimized allocation strategy. In this case, all shippable canisters would be able to ship every year leaving no change between allocation strategies.

Analyzing a realistic scenario with 74 reactor sites provided more clarity into the benefits of optimizing the allocation strategy. The OFF allocation strategy resulted in 1554 SYR, while the optimized allocation strategy resulted in 532 SRY. The optimized allocation strategy reduced the

number of SRY by almost 300%. Assuming a SRY costs the utility $8M this could reduce the total cost of the waste removal system by $8.18B.

In general, a “good” allocation strategy to reduce the number of SRY would have the following rules.

1. Prioritize removal of SNF from shutdown sites
2. Prioritize projected shutdown sites by year
3. If two sites have same projected shutdown sites, prioritize the reactor with the least SNF.

7.3 Future Work

The TVM was developed to ensure the correct implementation of optimization strategies in regards to reducing the number of shutdown reactor years. Adding on to the model could provide opportunities to examine how different parts fit together. The TVM could incorporate a smart loading strategy, which works to optimize the way assemblies are loaded into casks at the utility level. This could be a useful tool for utilities provided they know the allocation order. A utility may be able to load the canister to maximize the removal of the thermal source term from the SFP. A utility could also optimize the loading strategy to form a system wide coupling between the allocation strategy and the loading algorithm.

More work can be done in determining the optimal allocation strategies at the macro level. A Pareto formulation could be used to determine the best way to remove SNF from reactor sites to ensure fairness for all the utilities involved. It can provide a quantitative analysis to a qualitative

topic. Another formulation of the problem could attempt to level the number of SRY. This could create an equal distribution of SRY between all the utilities.

Finally, additional formulations of the integer programming could be found to solve the problem faster. The objective solution could be reformulated and some constraints may be combined. Some extraneous constraints may be eliminated.

Bibliography

1. Nuclear Waste Policy Act of 1982, 42 U.S.C. 10101 note. (2015) Retrieved from *http://epw.senate.gov/nwpa82.pdf*

2. DOE. (2004). "Acceptance Priority Ranking and Annual Capacity Report. "Participant#:HQO.20041004.0041, DOE/RW-0567, Department of Energy, Office of Civilian Radioactive Waste Management, Washington DC.

3. BRC (Blue Ribbon Commission on America's Nuclear Future). 2012. Report to the Secretary of Energy. Available online at *http://brc.gov/sites/default/files/documents/brc_finalreport_jan2012.pdf*

4. "US Nuclear Fuel Cycle." *US Nuclear Fuel Cycle*. World Nuclear News, Sept. 2015. Web. 04 Nov.2015 *http://www.world-nuclear.org/information-library/country-profiles/countries-t-z/usa-nuclear-fuel-cycle.aspx*

5. Reitenbach, Gail. "Dry Cask Storage Booming for Spent Nuclear Fuel. "POWER Magazine. 02 Feb. 2015. Web. 04 Nov. 2015. *http://www.powermag.com/dry-cask-storage-booming-for-spent-nuclear-fuel/*

6. "Spent Fuel Storage in Pools and Dry Casks Key Points and Questions & Answers." NRC: Spent Fuel Storage in Pools and Dry Casks. N.p., n.d. Web. 08 Dec. 2015. *http://www.nrc.gov/waste/spent-fuel-storage/faqs.html*

7. Nesbit, Steven P., and Lara S. Nichols. A Proposed Acceptance Queue for Shutdown Nuclear Power Reactors. Proc. of IHLRWM 2015, Charleston, SC. Charlotte, NC: Duke Energy Corporation, 2015. Print.

8. Ballou, L. B., D. N. Montan, and M. A. Revelli. Spent Fuel Receipt Scenario Study. Rep. no. UCID--21530. N.p.: Lawrence Livermore National Lab, 1990. Print.

9. "UPS: 'planning the UPS air network.'." Management Science 50.6 (2004): 855. Academic OneFile. Web. 22 Mar. 2016.

10. Murty, Katta G., Wan, Yat-wah, Liu, Jiyin, Tseng, Mitchell M., Leung, Edmond, Lai, Kam-Keung, & Chiu, Herman W.C. (2005). Hongkong International Terminals gains elastic capacity using a data-intensive decision-support system. Interfaces, 35(1), 61

11. Sahoo, Surya, Seongbae Kim, Byung-In Kim, Bob Kraas, and Alexander Popov, Jr. "Interfaces." *Routing Optimization for Waste Management*, January-February 35.1 (2005): 24-36. Print.

12. Karabakal N, Gunal A, Ritchie W. Supply-Chain Analysis at Volkswagen of America. Interfaces [serial online]. July 2000;30(4):46-55. Available from: Business Source Complete, Ipswich, MA. Accessed March 22, 2016

13. Maged Georgy, Sameh Y. Basily, (2008),"Using genetic algorithms in optimizing construction material delivery schedules", Construction Innovation, Vol. 8 Iss 1 pp. 23 – 45

14. Heinz, Stefan (Zuse Institute Berlin, Berlin, Germany); Christopher Beck, J. Source:COPLAS 2011 - Proceedings of the Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems, p 23-30, 2011, COPLAS 2011 - Proceedings of the Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems

15. Jarrell. Joseph. Howard. Hale. Petersen. Wilkerson. Fortner. Kalinina. Initial Standardized Canister System Evaluation. September, 2015 .

16. Chen, Der-San, Batson, Robert G., and Dang, Yu. Applied Integer Programming. Hoboken, US: Wiley, 2011. ProQuest ebrary. Web. 24 March 2016.

17. Alexander Schrijver. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA.

18. Bertsimas, Dimitris, and John Tsitsiklis. “Simulated Annealing”. Statistical Science 8.1 (1993): 10–15. Web..

19. Vose, M., & NetLibrary, Inc. (1999). *The simple genetic algorithm : Foundations and theory* (Complex adaptive systems). ISBN 0470031379. Cambridge, Mass.: MIT Press.

20. Fahad Syed Muhammad. Various resource allocation and optimization strategies for high bit rate communications on power lines. Signal and Image processing. 2010. English.

21. U.S. Nuclear Regulatory Commission. (1989). *Standard format and content guide for financial assurance mechanisms required for decommissioning under 10 CFR parts 30, 40, 70, and 72*. Washington, DC: Division of Low-Level Waste Management and Decommissioning, Office of Nuclear Material Safety and Safeguards, U.S. Nuclear Regulatory Commission.

22. Kadak, A.C, and K. Yost. 2010. Key Issues Associated with Interim Storage of Used Nuclear Fuel. MIT-NFC-TR-123. December. Cambridge, Mass.: Massachusetts Institute of Technology.

23. "Country Statistics – United States of America". IAEA—International Atomic Energy Agency. Retrieved April 18, 2015 *https://www.iaea.org/PRIS/CountryStatistics/CountryDetails.aspx?current=US*

24. "Nuclear Energy Overview". EIA. Retrieved April 2015 *http://www.eia.gov/*

25. "Who pays for Yucca Mountain?". Yucca Mountain Repository, U.S. Dept. of Energy. Archived from the original on August 22, 2008. *http://www.nrc.gov/docs/ML1006/ML100600704.pdf*

26. United States. (2014). Oversight of DOE's strategy for the management and disposal of used nuclear fuel and high-level radioactive waste: Hearing before the Subcommittee on Environment and the Economy of the Committee on Energy and Commerce, House of Representatives, One Hundred Thirteenth Congress, first session, July 31, 2013.

27. Nutt, Mark, Casey Trail, Thomas Cotton, Robert Howard, and Bret Van Den Akker. "Waste Management System Architecture Evaluations." Proc. of 2015 International High Level Radioactive Waste Management, Charleston, SC. 742-48. Print

28. U. S. Department of Energy, “Preliminary Evaluation of Removing Used Nuclear Fuel from Shutdown Sites,” FCRD-NFST-2013-000238. September 2013. *http://www.pnl.gov/main/publications/external/technical_reports/PNNL-22676rev1.pdf*

29. Department of Energy. "Office of Standard Contract Management." Department of Energy. Office of the General Council, n.d. Web. 18 Apr. 2016. *http://energy.gov/gc/services/environment-and-nuclear-programs/office-standard-contract-management*

30. NEI. "Repository for Spent Nuclear Fuel." Repository for Spent Nuclear Fuel. N.p., 15 Aug. 2001. Web. 18 Apr. 2016. http://lobby.la.psu.edu/066_Nuclear_Repository/orgstatements_nuke.html.

31. Gurobi. "Mixed-Integer Programming (MIP) - A Primer on the Basics."*Mixed-Integer Programming (MIP) Basics*. N.p., n.d. Web. 27 Apr. 2016. *http://www.gurobi.com/resources/getting-started/mip-basics*

32. Conforti, Michele, Gerard Cornuejols, and Giacomo Zambelli. Integer Programming. Switzerland: Springer International, 2014. PDF.

33. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. ad Teller, E. (1953). Equation of state calcuations by fast computing machines. *J. Chem*. Phys. 21 1087-1092.

34. Hajek, B. (1988). Cooling schedules for optimal annealing. *Math. Oper. Res*. **13** 311-329.

35. Sarker, R., Mohammadian, Masoud, Yao, Xin, & SpringerLink. (2002). *Evolutionary Optimization* (International Series in Operations Research & Management Science, 48).

36. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley.

37. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor, MI: The University of Michigan Press.

38. Jong, K. A. D. (1975). An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan, Ann Arbor.

39. Fraser, A. S. (1957). Simulation of genetic systems by automatic digital computers. I. Introduction. Australian Journal of Biological Sciences, 10, 484-491.

40. Koza, J. R. (1989). Evolving programs using symbolic expressions in Proc. of the 11th Int'l Joint Conf. on Artificial Intelligence (N. S. Sridharan,ed.), (San Mateo, CA), Morgan Kaufmann, 768-774.

41. Koza, J. R. (1990). Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems. Tech. Rep. STAN-CS-90-1314, Department of Computer Science, Stanford University.

42. Khuri S., Back T., Heitkoter J., "The zero/one multiple knapsack problem and genetic algorithm", *Proceedings of the Symposium of Applied Computation*, ACM, Phoenix, AZ, USA, 1994.

43. Flockton S.J., White M.S., "Pole-zero system identification using genetic algorithms", *Proceedings of the 5th International Conference on Genetic Algorithms*, ICGA, 1993.

44. Homaifar A., Shanguchuan G., Liepins G.A., "A new approach on the travelling salesman problem by genetic algorithms", *Proceedings of the 5th International Conference on Genetic Algorithms*, ICGA, Urbana-Champaign, IL, USA, 1993.

45. Yin X., Germany N., "Investigations on solving the load flow problem by genetic algorithms", *Electric Power Systems Research*, vol. 22, pp. 151-163, 1991.

46. Watabe H., Okino N., "A study on genetic shape design", *Proceedings of the 5th International Conference on Genetic Algorithms,* ICGA, Urbana-Champaign, IL, USA, 1993.

47. Schoenauer M., Ronald E., Damour S., "Evolving nets for control*", Proceedings of the 6th International Conference on Neural Networks and their Industrial and Cognitive Applications,* AFIA, Nimes, France, 1993.

48. Unger R., Moult J., "Agenetic algorithm for 3D protein folding simulations", *Proceedings of the 5th International Conference on Genetic Algorithm,* ICGA, Urbana-Champaign, IL, USA, 1993.

49. Schaffer J.D., Eshelman J.E., "Designing multiplierless digital filter using genetic algorithms", *Proceedings of the 5th International Conference on Genetic Algorithms,* ICGA, Urbana-Champaign, IL, USA, 1993.

50. Forrest S., Smith R.E., Jakornik B., *et al*., "Using genetic algorithms to explore pattern recognition in the immune system", *Evolutionary Computation*, vol. 1, no. 3, pp. 191-211, 1993.

51. Delahaye D., Alliot J.M., Schoenauer M., *et al*., "Genetic algorithms for air traffic assignment", *Proceedings of the European Conference on Artificial Intelligence*, ECAI, 1994.

52. Eshelman L.J., Caruana R.A., Shaffer J.D., "Biases in the crossover landscape", *Proceedings of the 3rd International Conference on Genetic Algorithms*, ICGA, Fairfax, Virginia, USA, 1989.

53. Syswerda G., "Uniform crossover in genetic algorithms", *Proceedings of the 3rd International Conference on Genetic Algorithms*, ICGA, Fairfax, Virginia, USA, 1989.

54. Goldberg D.E., *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley, Reading MA, 1989.

55. Back T., Hoffmeister F., "Extended seltion mechanisms in genetic algorithms", *Proceedings of the 4th International Conference on Genetic Algorithms*, ICGA, San Diego, CA, USA, 1991.

56. Goldberg D.E., Deb K., "Adaptive selection methods for genetic algorithms", *Proceedings of the Foundation of Genetic Algorithms*, FOGA, 1991.

57. Delahaye, D., Puechmorel, Stéphane, Puechmorel, Stéphane, & Ebrary, Inc. (2013).*Modeling and optimization of air traffic* (Computer engineering series (London, England)). London : Hoboken, N.J.: ISTE ; Wiley.

58. Kreher, D., & Stinson, Douglas R. (1999). *Combinatorial algorithms : Generation, enumeration, and search* (CRC Press series on discrete mathematics and its applications). Boca Raton, Fla.: CRC Press.

59. Eric Angel, Evripidis Bampis, Laurent Gourvès, Approximating the Pareto curve with local search for the bicriteria TSP(1,2) problem, Theoretical Computer Science, Volume 310, Issues 1–3, 1 January 2004, Pages 135-146, ISSN 0304-3975, http://dx.doi.org/10.1016/S0304-3975(03)00376-1. (http://www.sciencedirect.com/science/article/pii/S0304397503003761)

60. Tutum C.., Deb K., "A Multimodal Approach for Evolutionary Multi-objective Optimization (MEMO): Proof-of-Principle Results", *Proceedings of the 8th International Conference Evolutionary Multi-Criterion Optimization*, EMO, Guimaraes, Portugal, March29-April1, 2015.

61. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416–423, Morgan Kaufmann, San Mateo (1993)

62. Horn, J., Nafploitis, N., Goldberg, D.E.: A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 82–87 (1994)

63. Srinivas, N., Deb, K.: Multi-objective function optimization using non-dominated sorting genetic algorithms. Evolutionary Computation Journal 2(3), 221–248 (1994)

64. Chankong, V., Haimes, Y.Y.: Multiobjective Decision Making Theory and Methodology. North-Holland, New York (1983)

65. Miettinen, K.: Nonlinear Multiobjective Optimization. ISBN 978-1-4615-5563-6 Kluwer, Boston (1999)

66. Lewis, John; Loftus, William (2008). Java Software Solutions Foundations of Programming Design 6th ed. Pearson Education Inc. ISBN 0-321-53205-8., section 1.6 "Object-Oriented Programming"

67. Borror J. *Java Principles Of Object-Oriented Programming* [e-book]. [N.p.]: R & D Books; 1999. Available from: eBook Collection (EBSCOhost), Ipswich, MA. Accessed September 25, 2016.

68. United States Regulatory Commission. "Backgrounder on High Burnup Spent Fuel." NRC:. N.p., 30 Sept. 2015. Web. 26 Sept. 2016. <http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/bg-high-burnup-spent-fuel.html>.

69. United States Regulatory Commission. "Backgrounder on High Burnup Spent Fuel." NRC:. N.p., 30 Sept. 2015. Web. 26 Sept. 2016. <http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/bg-high-burnup-spent-fuel.html>.

70. W. M. Nutt, E. Morris, F. Puig, E. Kalinina, and S. Gillespie, Transportation Storage Logistics Model - CALVIN (TSL-CALVIN), FCRD-NFST-2012-000424, prepared for the US Department of Energy, October 2012

71. W. M. Nutt, E. Morris, F. Puig, J. Carter, P. Rodwell, A. Delley, R. L. Howard, and D. Giuliano, Used Fuel Management System Architecture Evaluation, Fiscal Year 2012, FCRD-NFST-2013- 000020 Rev. 0, prepared for the US Department of Energy, October 2012

72. W. M. Nutt, F. Puig, E. Morris, Y.S. Park, R. Joseph III, G. Petersen, D. Giuliano, R. Howard, Used Fuel Management System Architecture Evaluation, Fiscal Year 2013, FCRD-NFST-2013- 000377 Rev. 1, prepared for the US Department of Energy, October 2013.

73. BNFL Fuel Solutions Corporation. FuelSolutions Storage System Final Safety Analysis Report. Rep. no. WSNF-220. Rev 2. Campbell, California: BNFL, 2005. Print.

74. Miller, L., D. Basu, K. Das, T. Mintz, R. Pabalan, and G. Walter. Overview of Vauum Drying Methods and Factors Affecting the Quantity of Residual Water After Drying-Public Version. Rep. no. NRC-02-07-006. San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses, 2013. Print.

75. Joseph, R., III, R. Hale, G. Petersen, R. Howard, and M. Nutt. Process Flow Diagrams and Node Descriptions. Proc. of International High Level Radioactive Waste Management, Charleston Marriot, Charleston, SC. Print.

76. Joseph, R., R. Howard, R. Hale, G. Petersen, and M. Nutt. Process Flow Diagrams and Node Descriptions (PFDND) for the WMS. Proc. of Institute of Nuclear Materials Management, Atlanta Marriot, Atlanta, Georgia. Print.

77. Gurobi Optimizer Version 6.5.2. Houston, Texas: Gurobi Optimization, Inc., June 2016. (software program)

78. Kaushik Banerjee. “Streamlining Analysis Capabilities for SNF Management” Waste Management Symposium, March 2015, Phoenix AZ. Conference Presentation.

Appendix

Appendix A
Node Descriptions

1. Preparation of a canister for fuel loading

The inspection and any repair of a canister is to be performed in accordance with written procedures. Upon receipt of the cask verify that safety related items pertaining to the canister and cask are in accordance with FSAR commitments. The certification should specifically identify equipment by number and identify specific met and failed requirements. The certification should be attested by a person responsible for the QA function. The certification system should be described in the purchaser's QA program. Means should be provided by the COL to ensure validity of certificates. User must be able to demonstrate product was manufactured under a process of control. (NRC Inspection Manual No: 35752 Issue Date 10-03-07)

The following tasks may be performed in a suitable staging area or inside the plant's cask receiving bay with the canister in a horizontal or vertical orientation, as practical. First examine the empty canister for any physical damage that might have occurred since the receipt inspection was performed. The reception of any empty cask and the shipping of the casks with spent fuel for reprocessing are controlled by radiation protection specialists that check the fixed and non-fixed contamination and the gamma and neutron radiation according to the procedures in force. These inspections concern the irradiation of the load and the irradiation and contamination on the cask and rail wagons. The points to be checked compulsory were indicated in the "Transport Documentation of Radioactive Material" publication. The maximum permissible non-fixed external surface contamination was checked according to the applicable transport regulation requirements.

The canister should be clean and any packaging material or loose debris removed. Inspect the quick-connect fittings on the vent and drain ports for any physical damage, and repair or replace the fittings, as necessary. If repair is needed, the repair of any canister damage shall be performed and documented in accordance with an established procedure. Trial fit the top end shield plug, inner closure plate, and outer closure plate to reconfirm acceptable fit-up. Trial fit the AW/OS shield plate to the inner and outer closure plates. Trial fit the canister vertical lift

fixture lift adapter to the outer closure plate, if vertical canister transfer is to be performed. Remove the outer closure plate, inner closure plate, and top end shield plug. Move the empty canister into the cask receiving bay within the plant's fuel building or to another suitable staging area where it can be installed in the transfer cask. This can be done in a variety of ways, including movement with a trailer, movement on air pallets (on or off the empty canister shipping skid), etc.

To stage the transfer cask, connect the cask lifting yoke to the hook of the fuel building crane. Position the crane and the lifting yoke in the plant's cask receiving bay with the empty transfer cask. Then engage the lifting yoke with the transfer cask lifting trunnions and visually inspect the yoke lifting arms to assure that they are properly positioned and engaged on the cask lifting trunnions. Upend the transfer cask on the skid, if not already upended and place the empty cask in the cask decontamination area. In addition, horizontal movement of the cask should always be in a direction perpendicular to the plane of the trunnions. In this way, an inadvertent impact with an object will cause the cask to remain engaged with the lifting yoke and rotate on the trunnions. However, if a vertical canister transfer is to be used, a cask support pad is to be prestaged in the decontamination area. The cask support pad holds the transfer cask high enough to allow removal and installation of the bottom cover bolts on the cask bottom end.

2. Insert Canister into Transfer Cask

In order to insert the canister into the transfer cask, remove the cask top cover. Then using a crane and the empty canister vertical lift fixture, lower the empty canister into the transfer cask cavity and position the canister circumferentially to match the cask and canister alignment marks. There should be an approximately even canister/cask annular gap all around. The gap must be sufficient to permit installation and inflation of an annular seal. This operation may be performed in the cask decontamination area, the plant's cask receiving bay, or a suitable staging area depending on plant-specific conditions and rigging and handling operations must comply with the plant's NUREG-0612/ANSI N14.6 commitments.

If required for the fuel type to be loaded, install SNF assembly spacers into the canister guide tubes, if not already installed. If a canister is to be "short-loaded" (e.g., 20 SFAs for a W21 canister), install guide tube fuel stop(s) as shown on the applicable canister field assembly drawing (see Section 1.5.1 of the applicable Canister Storage FSAR). Next install the shield plug retainers on the transfer cask top flange and rotate the shield plug retainers to the cask exterior to permit unobstructed access for canister fuel loading. Alternatively, the shield plug retainers may be installed following canister fuel loading as the cask breaks the water surface, depending on plant conditions.

3. Place canister and transfer cask into fuel pool

To place the canister and transfer cask into the pool, first connect the cask lifting yoke to the hook of the fuel building crane, if not already in place and hang the top shield plug from the lifting yoke using the associated yoke rigging cables. Adjust the rigging cables to provide a level shield plug orientation and verify that the shield plug can be installed into the canister without binding. Remove the shield plug from the canister and lifting yoke and set it aside. Make sure that the Rigging and handling operations comply with the plant's NUREG-0612/ANSIN14.6 commitments. The proper seating of the top shield plug should be assured to avoid potentially high radiation exposure of cask operating and plant personnel. For the W74 fuel solutions canister, the upper basket assembly is removed from the canister at this point as discussed in Section 8.1.3 of the FuelSolutions™W74 Canister Storage FSAR.

Next evaluate any plant-specific crane limitations and, if necessary, drain the liquid from the cask neutron shield to assure that the crane limits are not exceeded. Once these checks are made, fill the cask/canister annulus with clean demineralized water. Place the inflatable cask/canister annulus seal into the upper cask liner recess and seal the cask/canister annulus by pressurizing the seal with compressed air. The use of clean demineralized water, an inflatable annulus seal, and the overflow/pressurization bottles assure that the interior surfaces of the transfer cask and the exterior surfaces of the canister will not become contaminated during submersion in the fuel pool. Visually analyze the cask bottom cover-to-flange joint for any visible leakage. If leakage occurs, drain the cask/canister annulus, remove the canister from the cask, and repeat the cask preparation sequence described above in Node A1.

If no leakage occurs fill the canister cavity with water from the spent fuel pool, or an equivalent source and connect the overflow/pressurization bottles to the fittings of the cask/canister annulus and the liquid neutron shield, in order to maintain a positive head during pool immersion. Likewise connect a quick-connect fitting to the canister vent port fitting to vent the area below the top shield plug. Position the cask lifting yoke and engage the transfer cask lifting trunnions and visually inspect the yoke lifting arms to assure that they are properly positioned and engaged on the cask lifting trunnions.

4. Load fuel into canister

Before loading the fuel verify that the spent fuel pool water level is at or above the minimum required for fuel transfer operations, including compensation for the water volume displaced by the cask. Then lift the cask/canister and position it over the cask loading area of the spent fuel pool in accordance with the plant's 10CFR50 cask handling procedures. As mentioned before

horizontal movement of the cask should always be in a direction perpendicular to the plane of the trunnions. In this way, any inadvertent impact with an object will cause the cask to remain engaged with the lifting yoke and to rotate on the trunnions.

Lower the cask into the spent fuel pool until the bottom of the cask is at the height of the pool water surface. As the cask is lowered into the fuel pool, spray the exterior surface of the cask and lifting yoke with clean demineralized water to wet the surface and ease decontamination when the cask is removed from the pool. Place the cask in the location of the spent fuel pool designated as the cask loading area. If the plant's spent fuel pool has a cask shelf or platform below the water level designed to keep the fuel building crane hook dry, the cask can then be set on this shelf or platform. At this time, the yoke can be disengaged and a yoke extension can be installed between the yoke and the crane hook to prevent immersion of the crane hook. The yoke should be rinsed with clean demineralized water as it is removed, and the extended yoke should be similarly rinsed as it is immersed in the pool water. The extended yoke should then be re-engaged with the cask trunnions. Visual confirmation of proper trunnion engagement should be made. The cask can then be lowered into the designated cask loading area. Next disengage the lifting yoke from the cask lifting trunnions, move the yoke clear of the cask, and remove the lifting yoke from the spent fuel pool. Spray the lifting yoke with clean demineralized water as it is raised out of the pool to reduce dose to the workers.

Then move a SNF assembly that meets the *technical specification* requirements contained in Section 12.3 of the respective FuelSolutions™ Canister Storage FSAR from the fuel pool storage rack position, in accordance with the plant's 10CFR50 fuel handling procedures and place the SNF assembly into a visual inspection area to record the identification number. Prior to insertion of the SNF assembly into the canister, the identification of the SNF assembly is to be independently verified by two individuals using an underwater video camera or other means, which is read and recorded and check this identification number against the site-specific canister loading plan prepared by the licensee. Also check the plant records to verify that the *technical specification* requirements contained in Section 12.3 of the respective FuelSolutions™ Canister Storage FSAR are met, which indicates that the SNF assembly is acceptable for dry storage. Position the SNF assembly for insertion into the selected canister guide tube and load the SNF assembly. Prior to release of the SNF assembly, record the location of the SNF assembly in the canister and verify its location against the canister loading plan. Repeat the process for each SNF assembly to be loaded into the canister.

If there are not enough SNF assemblies to fully load the canister, install dummy fuel assemblies in the empty guide tube openings that do not have mechanical blocks. The dummy fuel assemblies should have approximately the same external dimensions, total weight, and weight

per unit length as the fuel type being loaded to maintain the overall weight of a fully loaded canister.

5. Remove the loaded canister/transfer cask from the fuel pool

To remove the canister/cask from the pool suspend the top shield plug from the lifting yoke using the associated yoke rigging cables. Make sure rigging and handling operations comply with the plant's NUREG-0612/ANSI N14.6 commitments. After spraying the top shield plug, rigging cables, and yoke with clean demineralized water as they enter the fuel pool, position the lifting yoke and the top shield plug over the cask/canister and lower the shield plug into the canister and visually verify that the top shield plug is properly seated in the canister. The proper seating of the top shield plug should be assured to avoid potentially high radiation exposure of cask operating and plant personnel.

After verification position the lifting yoke and engage the cask lifting trunnions. Verify that the lifting yoke is properly engaged and lift the cask just far enough to allow the weight of the cask to be distributed onto the yoke lifting arms. Once the cask is lifted, re-inspect the lifting arms to assure that they are properly positioned on the cask trunnions. Raise the cask to near the pool surface, spraying the lifting yoke with clean demineralized water as it becomes exposed to air, but prior to raising the top of the cask above the water surface, stop vertical movement.

In plants where yoke extensions have been added to preclude immersion of the crane hook and where underwater cask shelves or platforms exist, the cask should be placed on that shelf or platform. The yoke should be disengaged and removed from the cask and raised out of the pool water. The yoke should be rinsed with clean demineralized water as it is being removed. The yoke extension should be removed and the yoke should be sprayed with clean demineralized water as it is re-immersed in the pool and re-engaged with the cask trunnions. Visual confirmation of proper trunnion engagement should be made.

With the cask near the pool surface, inspect the top shield plug to verify that it is properly seated in the canister. If not, lower the cask and reposition the top shield plug and repeat the previous steps as necessary. The proper seating of the top shield plug should be assured prior to lifting the cask above the pool surface to avoid potentially high radiation exposure of cask operating and plant personnel. Next rotate the temporary shield plug retainers into place; alternatively, if the shield plug retainers have not yet been installed due to plant conditions, they may be installed as the cask breaks the water surface. In addition, temporary shielding may be used to lower personnel radiation exposures. The shielding should be installed in accordance with plant-specific procedures. Continue to raise the cask from the pool and spray the exposed portion of

the cask and lifting yoke with clean demineralized water, until the top region of the cask is accessible in order to perform a radiation analysis. Check the radiation levels near the center of the top shield plug, in accordance with plant specific procedures and ALARA requirements (discussed in Section 10.1.3.2 of this FSAR). If the radiation levels exceed these requirements, return the cask to the cask loading area in the spent fuel pool and notify the cognizant management representative and await further instructions before proceeding.

If radiation levels are acceptable, proceed. Remove sufficient water from the top of the cask/canister back into the pool to expose the surface of the shield plug. Then lift the cask from the spent fuel pool. As the cask is raised from the pool, continue to spray the cask with clean demineralized water while recording the time of removal of the transfer cask from the fuel pool (i.e., the time the cask bottom end breaks the pool water surface). After recording the time, move the transfer cask with the loaded canister to the cask decontamination area. As previously mentioned, horizontal movement of the cask should always be in a direction perpendicular to the plane of the trunnions. In this way, any inadvertent impact with an object will cause the cask to remain engaged with the lifting yoke and to rotate on the trunnions. If vertical canister transfer is to be used, a cask support pad is to be prestaged in the decontamination area. The cask support pad holds the transfer cask high enough to allow removal and installation of the bottom cover bolts on the cask bottom end.

6. Decontaminate cask exterior

Once the cask/canister is in the decontamination area, disconnect the lifting yoke rigging cables from the top shield plug. After confirming that the lifting cables have been disconnected from the shield plug, disengage the lifting yoke from the trunnions and move it clear of the cask. Make sure that the top shield plug is not lifted during disengagement of the lifting yoke from the trunnions and removal from cask to avoid potentially high radiation exposure of cask operating and plant personnel. Next disconnect the overflow/pressurization bottles from cask/canister annulus and neutron shield fittings. If empty, fill the transfer cask neutron shield with liquid.

Reattach the neutron shield overflow/pressurization bottle. If required by site-specific seismic criteria, install the cask seismic restraint members. Then check the radiation levels near the mid-plane (mid-point) of the transfer cask to assure that dose rates are below maximum expected values, in accordance with site-specific procedures and ALARA requirements (discussed in Section 10.1.3.2 of Fuel Solutions FSAR Final Report). As previously mentioned temporary shielding may be used to lower personnel radiation exposures. The shielding should be installed

in accordance with plant-specific procedures. If the radiation levels exceed these requirements, notify the cognizant management representative and await further instructions before proceeding.

If radiation levels are acceptable, proceed. Decontaminate the accessible cask exterior surface and take swipes of the accessible surfaces to check for smearable contamination, in accordance with the *technical specification* requirements contained in Section 12.3 of Fuel Solutions FSAR Final Report. Remove the temporary shield plug retainer and deflate and remove the inflatable cask/canister annulus seal. Decontaminate the exposed surfaces of the canister shell perimeter adjacent to the shield plug, the top interior surface of the cask, top exterior surface of the canister above, and adjacent to the annulus seal location (Fuel Solutions FSAR April 2005).

In order to fully perform the decontamination procedure, the cask will then be spot decontaminated as necessary with high pressure water, commercial cleaners (Formula 409; Tri-Sodium Phosphate; and Blaze Off Emulsifier Degreaser Cleaner), high pressure steam, brushing and scouring, and a demineralized water rinse (Ref 3   V.10; APP IX).

7. Drain small amount of water from the canister/cask cavity then weld and inspect inner lid (vacuum or forced helium drying system)

Before beginning to install the inner plate verify that the neutron shield cavity is full. Then connect a drain line to the cask cavity drain port and allow water from the annulus to drain out until the water level is approximately 12 inches below the top edge of the canister shell. Take swipes around the outer surface of the canister shell and check for smearable contamination, in accordance with the technical specification requirements contained in Section 12.3 of Fuel Solutions FSAR Final Report. If the exterior of the canister has unacceptable contamination, the transfer cask/canister annulus may be drained and flooded as many times as necessary with clean demineralized water or plant-approved decontamination fluid to flush the canister's exterior of any unacceptable contamination. If the unacceptable contamination persists, return the loaded transfer cask to the fuel pool, remove SNF assemblies from the canister, remove the empty transfer cask and canister from the fuel pool, and remove the empty canister from the transfer cask for unrestricted access to the canister's exterior for decontamination following Sections 8.2.3.4 through 8.2.3.7.in Fuel Solutions FSAR.

Next cover the cask/canister annulus to prevent debris and weld splatter from entering the annulus (a lead "snake" can be used for this purpose). On a plant/canister-specific basis, the

canister vent port quick-connect fitting may be removed and a temperature measuring device with a quick-connect fitting installed to monitor the canister's water temperature while continuing to vent the canister. If the canister water temperature reaches 180°F, reinstall the cask/canister annulus seal and begin circulating cooling water through the cask canister annulus to cool the canister and prevent boiling of the canister water. In order to find the maximum temperature a specific canister's SNF decay heat, the prevailing ambient conditions, available annulus cooling water temperature and flow rate, as well as the lead time needed to initiate annulus cooling operations and prevent canister water boiling will determine the temperature below which a canister's water should be maintained. Prevention of canister water boiling is recommended to assure worker safety, but is not required for nuclear safety. Then connect the vacuum drying system dewatering pump to the canister drain port and remove approximately 15 gallons of water from the canister to lower the water level below the bottom of the shield plug. Return the water to the spent fuel pool. As previously mentioned temporary shielding may be used to lower personnel radiation exposures. The shielding should be installed in accordance with plant-specific procedures.

Check the radiation levels at the center of the top shield plug and perform radiation surveys in accordance with the site-specific procedures and ALARA requirements (discussed in Section 10.1.3.2 of this Fuel Solutions FSAR). If the radiation levels exceed these requirements, notify the cognizant management representative. Await further instructions before proceeding. If radiation levels are acceptable, proceed. Install the AW/OS onto the inner closure plate and place the inner closure plate with the AW/OS onto the canister. Verify proper positioning and fit-up of the inner closure plate with the canister shell prior to welding. Rigging and handling operations must comply with the plant's NUREG-0612/ANSI N14.6 commitments.

Prior to the initiation of welding, begin monitoring the perimeter of the inner closure plate and vent and drain port weld regions for the presence of hydrogen using a calibrated device capable of measuring concentrations of hydrogen to 0.4 % by volume. If hydrogen concentrations of 0.4 % by volume or more are detected, connect a "welding grade" argon source to the canister vent port. Purge the canister with argon gas prior to and as required during inner closure plate welding operations until the root pass of the weld is completed. If inner closure plate, vent and drain port body tack, and root pass welding begins without an argon purge through the canister's vent port, the vent port should remain vented and the perimeter of the inner closure plate and vent and drain port weld regions should continue to be monitored for the presence of hydrogen Next Tack weld the inner closure plate to the canister shell and tack weld the vent and drain port bodies to the inner closure plate. Place the inner closure plate and the vent and drain port body root pass welds. Just prior to completion of the second vent or drain port body root passweld, disconnect the argon gas source from the vent port and connect a hose to the canister vent port and route the

hose to the spent fuel pool (or other suitable water receiving vessel or location). Vent the canister to assure that internal pressure remains atmospheric during welding operations. Complete the root pass of this last inner closure plate weld.

Perform a dye penetrant inspection of the inner closure plate and the vent and drain port adapter root pass welds in accordance with ASME BPVC4 Subsubarticle NB-5350. With the canister vented through its vent port, complete the inner closure plate and the vent and drain port body welds. The canister should remain vented through its vent port at all times (except when used for purging) until the immediate start of the draining process. Perform a dye penetrant weld examination of the completed inner closure plate and vent and drain port body welds in accordance with NB-5350.

8. Install canister outer closure plate

Before draining the water, re-verify that the cask/canister annulus and neutron shield cavities are full before removing additional water from the canister. Then install the inner closure plate strongback, isolate the vacuum drying system, and open the compressed gas supply valve to allow the compressed inert gas (e.g., argon, helium, or nitrogen) to force the water from the canister cavity through the drain port to a maximum pressure of 30 psig. Throughout the draining, Monitor the canister pressure using the gauge on the vacuum drying system. Once water stops flowing from the canister, continue to purge with compressed inert gas for 30 minutes minimum. Isolate the compressed gas supply and disconnect the canister drain port hose.

Check the radiation levels near the center of the canister top end and near the mid-plane (mid-point) of the cask to assure that dose rates are below maximum expected values, in accordance with site-specific procedures and ALARA requirements (discussed in Section 10.1.3.2 of this FSAR). If the radiation levels exceed these requirements, notify the cognizant management representative. Await further instructions before proceeding. If radiation levels are acceptable, proceed to open the valve on the suction side of the pump and start the vacuum drying system to draw a vacuum on the canister cavity. The cavity pressure should be reduced in a step-wise progression (for example, 100 torr, 50 torr, 25 torr, 15 torr, 5 torr, and 3 torr). After pumping down to each level, the pump is valved off and stopped, and the cavity pressure monitored. The cavity pressure will rise as water and other volatiles in the cavity evaporate. When the cavity pressure stabilizes, the vacuum pump is reactivated and the pressure reduced to the next step. It may be necessary to repeat some steps, depending on the rate and extent of the pressure increase. Maintain the vacuum until a stable vacuum pressure has been achieved in accordance with the technical specification requirements contained in Section 12.3 of Fuel Solutions FSAR. Vacuum drying times are controlled by the Vacuum Drying Program established in accordance with the technical specification requirements contained in Section 12.3 of the respective canister FSAR.

The vacuum drying system may be connected to both the vent and the drain ports to expedite the drying process, but During vacuum drying the cask/canister annulus water level should be maintained at approximately 12 inches below the top edge of the canister shell.

Next isolate the vacuum drying system from the canister and connect a supply of compressed helium (if not already connected) to the canister vent port via the vacuum drying system. Allow compressed helium to flow into the canister cavity and pressurize the canister with 99.995% pure helium gas to a minimum of 12.5 psig in accordance with the requirements of Article NB-6000. Perform the helium leak rate test of the inner top closure plate and the vent and drain port body welds, in accordance with the technical specification requirement contained in Section 12.3 of Fuel Solutions FSAR and Subarticle NB-6300 in order to satisfy both pneumatic pressure testing and helium leak testing requirements. Once the system is demonstrating compliance with the technical specification requirement, isolate the source of compressed helium and lower the canister pressure by connecting a hose to the canister drain port which is routed into the spent fuel pool (or other suitable receiving vessel or location). Re-evacuate the canister, by repeating the progressive decrease of pressure in steps as described earlier, until a stable vacuum pressure has been achieved and held in accordance with the technical specification requirements contained in Section 12.3 of this FSAR.

Isolate the vacuum drying system from the canister and connect a supply of 99.995% pure compressed helium gas to the canister vent port via the vacuum drying system (if not already connected) with a calibrated in-line (temperature and pressure compensating) mass flow meter with an integrated read-out. Re-pressurize the canister, allowing a specified mass of helium to flow into the canister cavity, in accordance with the technical specification requirement contained in Section 12.3 of the respective FuelSolutions™ Canister Storage FSAR. Isolate the source of compressed helium and disconnect the vacuum drying system from the canister. The amount of helium allowed to flow into a canister is dependent on the canister and/or fuel assembly types. If dummy fuel assemblies are loaded in place of actual SNF assemblies, the quantity of helium backfill gas may need to be adjusted to compensate for the differential volume between the dummy assemblies and the assumed SNF assembly volumes. Once the helium backfill is complete, place the prefabricated port covers over the vent and drain ports. Tack the covers in place, as required, and place the root pass weld to the vent and drain port bodies. Complete the vent and drain port cover welds and Perform a dye penetrant examination of the completed vent and drain port cover welds, in accordance with NB-5350. Remove the AW/OS from the canister. The inner closure plate strongback may be removed at any time after connecting a supply of pure compressed helium to the canister vent port using the vacuum drying system.

9. Transfer canister from transfer cask to storage cask

First install the AW/OS onto the canister outer top closure plate, and place the outer top closure plate with the AW/OS onto the canister. Verify proper positioning and fit-up of the outer top closure plate with the canister shell, prior to welding. *Rigging and handling operations must comply with the plant's NUREG-0612/ANSI N14.6 commitments.* Place the outer top closure plate root pass weld. Perform a dye penetrant examination of the outer top closure plate root pass weld, in accordance with NB-5350 and place additional outer top closure plate weld passes until approximately ½ of the outer top closure weld preparation depth is filled. Perform a dye penetrant examination of the outer top closure plate intermediate level weld surface, in accordance with NB-5350. Complete the outer closure plate weld and perform a dye penetrant examination on the completed outer closure plate weld, in accordance with NB-5350. Remove the AW/OS from the canister, enter the date on the canister nameplate located on the outer closure plate, and record the canister serial number. Then Connect a drain line to the cask cavity drain port and remove the remaining water from the cask/canister annulus.

Appendix B
Fuel Projection

Example Fuel Projection

Batch_ID	CALVIN_RX_ID	MTU	NUM_ASS	Burnup	Enrich	Discharge_Date	Pool_ID	CALVIN_ID	Dry_Year
36	40	0.179674	1	324	2.133	6/5/1970	6601	65	0
37	40	0.183384	1	354	2.132	6/5/1970	6601	65	0
38	40	0.191193	1	177	2.131	6/5/1970	6601	65	0
39	40	1.340816	7	332.2802	2.13243	6/5/1970	6601	65	0
40	40	0.574137	3	353.334	2.131667	6/5/1970	6601	65	0
41	40	0.192687	1	232	2.135	6/5/1970	6601	65	0
42	40	0.577022	3	342.6555	2.134999	6/5/1970	6601	65	0
43	101	17.57433	48	18075.75	3.157001	10/2/1970	6605	148	0
44	41	0.76435	2	8614	3.413	2/4/1971	6402	112	0
45	41	1.5287	4	5856.25	3.473	2/4/1971	6402	112	0
46	41	2.29305	6	8652.834	3.473	2/4/1971	6402	112	0
47	6	0.138	1	5502	3.62	2/12/1971	6401	111	0
48	6	0.278	2	9780.296	3.62	2/12/1971	6401	111	0
49	6	0.278	2	10660.29	3.62	2/12/1971	6401	111	0

Vita

Gordon Matthew Petersen was born in 1991 to Joe and Teresa Petersen in Knoxville, TN. He attended high school at the Webb School of Knoxville graduating in 2010. He enrolled at the University of Tennessee in Fall of 2010 and earned his Bachelor of Science in Nuclear Engineering. During his time there he re-inaugurated a chapter of the Health Physics Society in 2013 serving as president for two years. He also joined the club water polo team earning an MVP award in 2012.

Gordon helped the community by serving as an assistant swim coach to Gulf Park Swim Team from the summers of 2010-2012. Additionally he began helping the Knoxville Central High School Swim Team in 2011 and was named the head coach in 2011. He continued as head coach through 2016.

During the summer of 2013, Gordon began interning at Oak Ridge National Lab (ORNL) working on the development of Process Flow Diagrams and Node Descriptions. This work led the Used Fuel Systems Group in the Reactor and Nuclear Systems Division to sponsor his post graduate work at the University of Tennessee. At ORNL he worked on system analysis of the waste management system. He received his Master of Science in Nuclear Engineering for his work concerning the similarities between standardizing containers in the transportation and distribution industry and utilizing standardized canisters in the nuclear industry.

This work propelled him into his PhD studying the effects of shutdown reactor years on the waste management system. The study evolved into discovering ways to optimize the allocation strategy to minimize the number of shutdown reactor years. Gordon completed his PhD in 2016 and is getting married to his fiancé, Betsy Pack, in 2017.