INL/JOU-22-65821-Revision-0

Towards an Agent-Based Blackboard System for Reactor Design Optimization

November 2021

Ryan Hunter Stewart, Samuel E Bays, Todd S Palmer

Changing the World's Energy Future

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/JOU-22-65821-Revision-0

Towards an Agent-Based Blackboard System for Reactor Design Optimization

Ryan Hunter Stewart, Samuel E Bays, Todd S Palmer

November 2021

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

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

Towards an Agent-Based Blackboard System for Reactor Design Optimization

Ryan Stewart,[*,a,b] Todd S. Palmer,[a] and Samuel Bays[b]

[a] *Idaho National Laboratory, 2525 N Freemont Ave., Idaho Falls, ID 83415*

[b] *Oregon State University, 1500 SW Jefferson St. Corvallis, OR 97331*

[*]Email: ryan.stewart@inl.gov

Number of pages: 38
Number of tables: 11
Number of figures: 18

Abstract

The field of reactor design is rich with opportunities for applications of computational optimization algorithms; these applications can range from preliminary core design to reactor shuffling patterns. Many of these schemes rely on sets of previously generated solutions (sometimes referred to as "generations") to inform future decisions. While it is important to build upon prior knowledge, this process requires a full generation of solutions to be formed before future solutions can be examined. Rather than relying on a generational scheme to perform an optimization, we propose using an agent-based approach in conjunction with a blackboard framework for performing reactor design optimizations. Utilizing an agent-based approach allows agents to perform tasks independently, while retaining the ability to build off of previous solutions. We develop an agent-based blackboard system (ABBS) for determining the Pareto front (PF) in sodium fast reactor design optimization problems and compared this with the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). Our goal is to evaluate the viability of the ABBS in producing a PF which is comparable with the NSGA-II algorithm. The design space consists of the fuel height, fuel smear, and plutonium fraction in the core, and we seek to minimize the reactivity swing and plutonium mass, while maximizing the burnup. The diversity, coverage, and spread of the PFs generated by the two methods are examined, and the ABBS is able to converge to the same PF as the NSGA-II algorithm. These results show that the ABBS is able to find optimal designs that are similar to those found by the NSGA-II algorithm. We conclude our study by applying the ABBS to the design of a sodium-cooled fast reactor to dispose of weapons-grade plutonium. The ABBS finds a core design which can burn upwards of 17.5 kg of weapons-grade plutonium per year, and degrade an additional 195 kg of weapons-grade plutonium per year into non weapons-grade material.

Keywords — Multi-Objective Optimization, Blackboard System, Agent-Based Modeling, Sodium Fast Reactor, Nuclear Non-Proliferation

I. INTRODUCTION

Multi-objective optimization algorithms are used across science and engineering disciplines to find ideal solutions satisfying a set of objectives and constraints. This process of determining optimal reactor designs has been difficult in the past, due to limitations in computational power, and tended to rely on expert knowledge to find ideal designs. The continual increase in computational power has allowed researchers in nuclear reactor design to create more efficient and robust designs with the aid of computational optimization algorithms [1, 2, 3]. O. Fabbris et al. [4] examined the design space for a low sodium void worth sodium-cooled fast reactor (SFR) core, using a genetic algorithm (GA) coupled with a surrogate model. 30 unique design variables (such as fuel pellet radius, inner/outer core enrichment, etc.) were adjusted, which created a large initial design space. Seven objectives (such as sodium feedback coefficient, maximum linear power, etc.) were selected to determine the solution quality and were used in addition to other constraints for the optimization algorithm. Surrogate modeling was accomplished by performing an initial design of experiment, which produced a database of high-fidelity solutions. This surrogate model allowed for a sensitivity analysis and an optimization problem to be solved in a shorter time frame, compared with utilizing full core simulations. Through the sensitivity analysis and optimization process, the authors were able to glean information about both the optimized core designs and global tradeoffs for the various parameters. This included finding reactor designs that were deemed "more" optimal than the reference design and found that adjusting the constraints placed on the core could yield additional optimal designs.

Recently, researchers have been exploring the use of multi-agent systems to help solve large and/or complicated problems. Huff et al. [5] utilize an agent-based system, Cyclus, to simulate the nuclear fuel cycle. Cyclus examines the effects of fuel cycle choices for individual nuclear plants or wider-scale fleets. This includes following nuclear material through its fuel cycle accounting for fuel enrichment, burnup, etc. Various metrics can be examined to determine the effects on economics, nuclear proliferation, and environmental concerns. Due to the customizability of agent based programming, users can examine custom applications rather than simulations that were original developed for Cyclus. Hulse et al. [6, 7, 8] proposed using a multi-agent system to perform a design-based optimization on complex systems, where expertise in multiple areas of study are required. Each agent was required to design an aspect of a drone with four propellers (i.e., a

quadrotor), rather than developing a full quadrotor design, which is typical of a general optimization algorithm. Agents solved subproblems concurrently by designing specific components of the quadrotor. This process involved generating a framework where agents learn and choose actions based on an action-value learning system, which keeps track of actions taken and their corresponding reward. Compared with a simulated annealing algorithm and a GA, the multi-agent system was able to outperform the simulated annealing algorithm but fell short of the GA in terms of producing an optimized final result.

Reactor design has benefited from the use of optimization algorithms to help designers generate optimal reactor configurations. The use of agent-based programming has helped researchers in various fields of engineering solve problems which can be broken into subcomponents. Prior research has proposed or used blackboard systems in conjunction with a multi-agent environment for real-time planning, environmental monitoring after a catastrophe, and large-scale structural assembly [9, 10, 11, 12]. We examine the implementation of a multi-objective optimization algorithm using an agent-based blackboard system (ABBS) [13]. The ABBS will be compared against the Non-dominated Sorting Genetic Algorithm II (NSGA-II) from the Python package `pymoo` — to evaluate and compare the ability of the ABBS to discover the Pareto front (PF) [14]. For this work, the ABBS is executed in serial to initially remove many of the complications that naturally arise when working with a multi-agent environment. Future iterations of the ABBS will allow for a fully implemented multi-agent environment and will create a highly parallelizable multi-objective optimization algorithm. This work serves as an intermediate step for developing a multi-agent blackboard system (MABS) to solve multi-objective optimization problems.

I.A. Optimization

Optimization is a field of mathematics that focuses on finding the "best" solutions satisfying a set of given objectives and subject to a number of constraints. Typical optimization problems focus on a set of n independent design variables (denoted by the vector $\mathbf{x}$), along with a set of k objective functions ($f_k(\mathbf{x})$). The design variables and objective functions make up the design space and objective space. Objectives define the quality of solutions and are combined in a meaningful way to create an objective vector ($F(\mathbf{x})$). Each objective vector is a solution to the optimization problem. The goal of an optimization algorithm is to find the most optimal solution(s); the

standard form of a multi-objective optimization problem can be seen in Equation 1 [15]. In addition to the objective functions, we can apply up to m inequality (g_j) and e equality (h_l) constraints to help bound the problem.

$$\begin{aligned} min\ \mathbf{F}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x}))^T \\ \text{subject to } g_j(\mathbf{x}) &\leq 0, \quad j = 1, 2, ..., m \\ h_l(\mathbf{x}) &= 0, \quad l = 1, 2, ..., e \end{aligned} \tag{1}$$

For a multi-objective optimization problem there typically exists a set of ideal solutions, denoted as the Pareto front (PF), rather than a single solution. Solutions that reside on the PF are called non-dominated solutions and optimize the design objectives. A non-dominated solution can be described by looking at two separate designs, **a** and **b**. Objective vector $\mathbf{F}(\mathbf{a})$ represents design **a** and has i objective functions $(f_1(\mathbf{a}), f_2(\mathbf{a}), ..., f_i(\mathbf{a}))$, similarly $\mathbf{F}(\mathbf{b})$ represents design **b** and has i objective functions $(f_1(\mathbf{b}), f_2(\mathbf{b}), ..., f_i(\mathbf{b}))$. If all objective functions in $\mathbf{F}(\mathbf{a})$ are equal to or less than all the objective functions in $\mathbf{F}(\mathbf{b})$ (i.e., $f_1(\mathbf{a}) \leq f_1(\mathbf{b}), ..., f_i(\mathbf{a}) \leq f_i(\mathbf{b})$) and is strictly less than at least one objective function (i.e., for some objective function i, $f_i(\mathbf{a}) < f_i(\mathbf{b})$), **a** dominates **b**. The PF gives a set of optimal solutions with which to examine the trade-offs associated with multiple objectives. Any solution in the PF can be selected based on the user's preference.

Optimization algorithms are used to find the PF; in this research, we compare the ABBS (described in the next section) with the NSGA-II algorithm, a common genetic algorithm (GA) [14]. GAs leverage evolutionary characteristics to promote solutions that are superior and pass on their traits [16]. For a GA, a chromosome is used to denote the set of design variables ($\mathbf{x}$), and a gene denotes a specific design variable (x_i). GA's determine optimal solutions resulting from the probing of the design space over multiple generations. In each generation, some subset of chromosomes will cross over (exchange genes between superior chromosomes) or mutate (introduce point defects to change the value of a single gene) to further explore the design space. This subset of chromosomes are selected based on a fitness function that determines a chromosome's optimality, where solutions that have more optimal objective functions will have a higher fitness function. After a set number of generations, the chromosomes that are left are typically on the PF.

I.B. Blackboard Framework

A blackboard system is a problem-solving technique that utilizes a series of knowledge sources (knowledge agents in an agent-based system), each of which is able to contribute information to an overarching problem [17, 18]. The blackboard system consists of three components: the blackboard agent (BA), knowledge agents (KA), and a controller agent (CA).

The BA acts as a global database for storing problem-specific information (partial results, alternatives, input data, etc.) in the form of *abstract levels* for other KAs to utilize. Abstract levels can involve a construct stored in memory (i.e. a dictionary, array), partitioned construct (i.e. a nested dictionary, matrix), or a relational database. The design for abstract levels can allow for KAs to quickly scan for relevant information [9].

The KAs provide information required to solve various aspects of the problem. KAs can vary the approaches for solving the problem, examine different portions of the problem, or move information to and from abstract levels. No single KA can typically solve the problem independently. Instead, KAs wait until they can contribute to the problem, and, when their knowledge can contribute, they write this information to the blackboard. Once a KA contributes to the blackboard, the information that the KA provides may allow other KAs to act.

The CA organizes the problem, manages problem flow, and allocates resources for KAs. Figure 1 shows the problem flow for the blackboard system. Organizing and managing the problem flow requires the CA to assess how each KA can contribute to the problem. KAs read the blackboard to determine their ability to contribute to the problem, and then send a trigger value (TV) to the CA, based on their assessment. The CA receives each KA's TV and selects the KA most likely to progress the problem. Once a KA has been selected, it performs its action and writes any resulting information to the blackboard. This process repeats until the overarching problem has been solved. To communicate, the CA passes messages back and forth with the KAs to determine KA availability, inform KAs when to perform tasks, and allow KAs to update the blackboard. KAs are only able to communicate with the CA and not with other KAs; this leads to efficient knowledge extraction and acquisition.

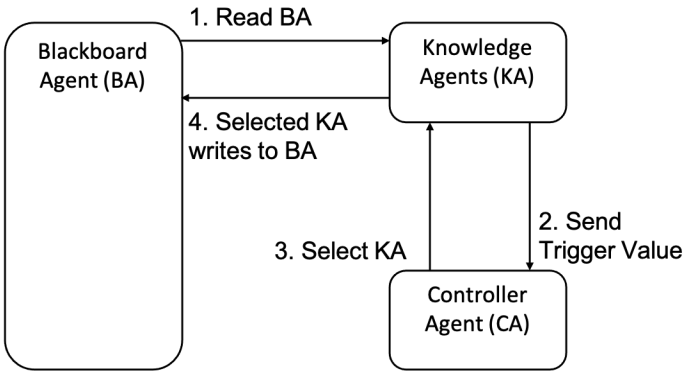


Fig. 1. Basic flow for blackboard framework to solve a problem.

II. METHODOLOGY

II.A. Sodium-Cooled Fast Reactor Design

The capabilities of the ABBS are tested on the optimization of a small SFR. The design and assembly aspects that are constant are described in Tables I and II. Core design and assembly parameters draw heavily from recent scoping studies for small SFRs [19]. The blank assemblies described in Table I were envisioned for control rod placement, additional fuel, or experimental assemblies.

TABLE I
Core Parameters.

Core Parameters	
Power (MWth)	300
Num. Fuel Assemblies	78
Num. Blank Assemblies	13
Rings in Core Region	6
Rings of Reflectors	3

TABLE II
Assembly Parameters.

Assembly Parameters	
Num. Pins	271
Assembly Pitch (cm)	12.0
Duct Thickness (cm)	0.3
Duct Inside Flat-to-Flat (cm)	11.1
Pin Pitch (cm)	0.661
Fuel Pin Diameter (cm)	0.53
Clad Thickness (cm)	0.037
Wire-Wrap Diameter (cm)	0.126

The fuel height, fuel smear, plutonium fraction, and cycle length were selected as design variables. The ranges for each design variable can be seen in Table III. The fuel smear, the ratio of the fuel slug to fuel pin internal volume, is used to describe the fuel slug volume. The fuel pin volume is held constant, while the volume of the U-Pu-Zr material is varied. In practice, the fuel pin volume may be reduced in step with a reduction in fuel slug volume; the effect on core neutronics is negligible. The fuel is U-Pu-Zr with 10 wt% Zr, and a ^{235}U + Pu weight of 27 wt%. An enrichment level of 27 wt% was selected based on previous design work for small SFRs [19]. The plutonium isotopic vector is weapons-grade plutonium with 94 wt% ^{239}Pu and 6 wt% ^{240}Pu.

The plutonium fraction describes the percent of 27 wt% that is plutonium. The remaining fissile material is made up of ^{235}U.

Following previous research into small SFR cores, we used a power level of 300 MWth [19]. Burnup calculations were performed out to 600 days to determine an approximation for the total burnup, reactivity swing, and fuel isotopics. We assumed that one fourth of the core will be replaced each cycle.

TABLE III
Design Variables for SFR Optimization.

Design Variable	Range
Fuel Height (cm)	50–100
Fuel Smear	0.5–0.8
Plutonium Fraction	0.0–1.0
Cycle Length (days)	120 or 240

High-fidelity simulations of various core configurations were performed using MCNP [20]. To generate each unique core configuration, we used the module FRIDGE [21]. To reduce the time requirements, we utilize a surrogate model to avoid running high-fidelity simulations. A surrogate model utilizes some type of regression to calculate the objective function and constraint values based on the design. The use of a surrogate model follows previous work performed in [4].

Data from the high-fidelity models was used to create a surrogate model via the Gaussian Process technique [22, 23]. Both optimization algorithms generate a design comprised of the three design variables and sends this information to the surrogate model. The surrogate model then returns the objective function and constraint values associated with the design back to the optimization algorithm. Both optimization algorithms used the same surrogate model to maintain consistency.

Table IV shows a list of constraints and objectives that can be placed on the problem. Each of the three core optimization problems will use a different set of constraints and objectives to explore the design space and presents the variety of ways in which a small SFR can be operated. For objective functions referencing changes in reactivity ($\Delta\rho$), reactivity is defined in terms of percent milli (pcm). Equation 2 defines an arbitrary change in reactivity, where k_1 is the eigenvalue at state 1 and k_2 is the eigenvalue at state 2.

$$\Delta\rho = \frac{k_2 - k_1}{k_1 k_2} * 10^5 \tag{2}$$

The reactivity swing is defined as the change in reactivity over a given time period and is used to gauge the necessary requirements for the control rod worth. Average burnup is the time an assembly spends in the core at a given power level for a given heavy metal loading, and will typically be bound by adhering to conventional metallic fuel burnup limits [24]. Plutonium mass is the mass of plutonium required each cycle to maintain the critical core for the desired cycle length. Excess reactivity measures the amount of reactivity left in the core at the end of each cycle. Some excess reactivity is needed to offset the negative reactivity associated with experiments placed in the core and is important for transitioning from a fresh core (as presented) to an equilibrium core. Limiting the maximum linear heat generation rate (LHGR) ensures that the fuel does not melt during normal operations or accident scenarios [19]. The Doppler and void coefficients both describe negative reactivity feedback mechanisms in the core. The Doppler coefficient describes the change in reactivity due to an increase in the fuel temperature. The void coefficient describes the change in reactivity due to a decrease in the coolant density, which is described as a change in the percent of sodium voided from the core. The ^{239}Pu/Pu ratio is measured at the end of cycle and describes the attractiveness of the plutonium, as a proliferation concern. The ^{240}Pu/^{239}Pu ratio is also a measure of the plutonium attractiveness and is commonly referred to in many nonproliferation treaties to ensure non-weapons-grade material [25]. Plutonium disposed is the mass of plutonium that has been consumed in the fuel during its time in the core.

TABLE IV
Objectives and Constraints for SFR Optimization Problems.

Objective/Constraint	
Reactivity Swing (pcm/month)	Doppler Coeff. (pcm/K)
Burnup (GWd/MT)	Void Coeff. (pcm/%void)
Pu Mass (kg/cycle)	^{239}Pu/Pu Ratio
Excess Reactivity (pcm)	^{240}Pu/^{239}Pu Ratio
Max LHGR (kW/m)	Pu Disposed (kg/cycle)

The objectives and constraints will be combined in various forms to generate three optimization problems. The first problem examines the relationship between minimizing the reactivity swing and maximizing the average burnup acquired for a fixed cycle length. The second problem minimizes the reactivity swing and plutonium mass required each cycle, while maximizing the

average burnup. The third optimization problem describes how an SFR could help reduce the ^{239}Pu/Pu ratio and the mass of weapons-grade plutonium. Additional constraints, such as the Doppler coefficient, void coefficient, excess reactivity, and peak LHGR, are placed on the last two optimization problems to provide additional design realism.

The construction of a small SFR core could be used to perform experimental irradiations, provide electricity or process heat, or gain operational experience for future commercial ventures. In addition to these uses, a small SFR core could be used to dispose of weapons-grade plutonium. A small SFR core would not provide a plutonium throughput comparable with large nuclear reactor, however, this task could be performed in-tandem with any other uses. The objectives for plutonium disposition are to reduce the mass of plutonium present and to dilute the plutonium isotopic vector. Weapons-grade plutonium is typically defined as the ratio of ^{239}Pu to the total plutonium content being greater than 0.94 (^{239}Pu/Pu > 0.94) [25]. To dilute the plutonium vector, we require a ratio of ^{239}Pu/Pu < 0.90 and a ^{240}Pu/^{239}Pu > 0.1 [25]. These limits are derived from the Plutonium Management and Disposition Agreement (PMDA) between the United States and Russia [26]. Each party agreed to reduce their weapons-grade plutonium stockpiles by 34 MT, at a rate of 1.3 MT per year. This plutonium vector is still designated as weapons-usable; however, these constraints have been used in previous weapons-reductions treaties [26]. While both parties are no longer in agreement with the PMDA, the reduction and dilution of weapons-grade plutonium is still an admirable goal [27].

II.B. Blackboard Architecture

The method by which a GA determines the PF was described in Section I.A; however, it is important to explain the method by which the ABBS will find optimal solutions. The ABBS utilizes the blackboard environment with various agents to explore the objective space and discover the PF. Each component of the blackboard is a unique agent running on a separate thread, where the multi-agent environment is achieved with the Python module *osbrain* [28]. The problem flow follows in a manner similar to the previous blackboard description, however the BA and CA in the ABBS differ slightly from their previous descriptions. The difference is due to constraints placed on a multi-agent environment for communication and action execution.

The BA contains two primary functions: storing the blackboard and communicating with

the KAs. Three abstract levels are present on the blackboard; these levels are denoted *Data Level*, *Viable Design Level*, and *Pareto Front Level*. The *Data Level* consists of specific core design information, including values for each design variable, objective function, and constraint. All core designs that are examined are placed on the *Data Level*, regardless of if they are viable in accordance with the objectives and constraints. The *Viable Design Level* is an intermediate abstract level, containing core names that fulfill all constraints and objective functions placed on the problem. The *Pareto Front Level* consists of the PF and contains optimal reactor designs. The abstract levels along with the data found in each level can be seen in Figure 2. The BA stores the abstract levels in memory; these abstract levels are also written to an HDF5 file [29]. Storing the blackboard in an HDF5 file allow for restarting problems, updating the optimization scheme, or preforming post-process analysis.

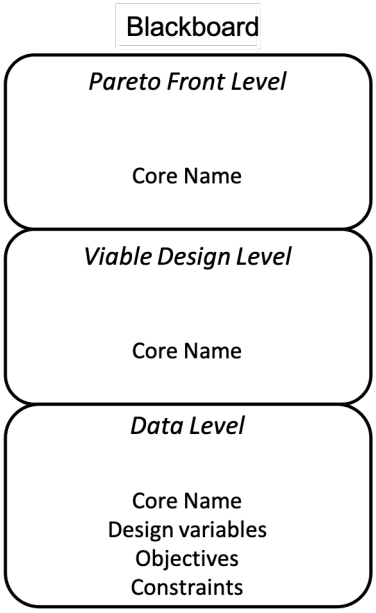


Fig. 2. Structure for Blackboard Abstract Levels.

The BA also sends and receives messages to and from the KAs. There are four types of messages that can be sent between the BA and KAs: trigger, executor, writer, and action complete. When a message is sent, there is a corresponding *handler* function that performs a set of actions. For a trigger message, the BA requests a TV from each agent, and each KA trigger handler replies with their TV. The BA assesses these TVs and sends an executor message to the KA with the highest TV. The executor handler tells the KA to perform its action. While a KA is performing

its action, it will occasionally be able to contribute to the blackboard. To contribute to the blackboard, a KA sends a writer message to the BA. If the BA is available, the KA can then write information has been gathered to one of the abstract levels. Upon completing its action, the KA sends an action complete message. Message passing also allows the BA to send data contained on the blackboard to KAs.

The ABBS consists of two KA types: blackboard reader agents (KA-BR) and search agents (KA-S). KA-BR agents move solutions on the blackboard to find optimal solutions and remove non-optimal solutions. Three different KA-BR agents are used to move solutions from the *Data Level* up to the *Pareto Front Level*. KA-BR-3 examines solutions from the *Data Level* and promotes solutions that meet all objectives and constraints of the optimization problem to the *Viable Design Level*. KA-BR-2 examines solutions from the *Viable Design Level* and determines if solutions are optimal when compared with solutions in the *Pareto Front Level*. If solutions are optimal, they are then placed into the *Pareto Front Level*. KA-BR-1 maintains the *Pareto Front Level* using a grid of hyperboxes from the Diversity Comparison Indicator (DCI). The DCI compares the diversity of the PF by placing solutions in hyperboxes [30]. The hypergrid environment is user defined, and in our implementation we create hyperboxes which represent 1% of objective range. For example, the range for the burnup objective is 50–150 GWD/MTU, so each hyperbox has a size of 1 GWD/MTU. If multiple solutions are present in a hyperbox, one solution is selected to remain. This process is used to promote a diverse set of solutions, while maintaining the size of the PF.

KA-S agents search the objective space using different methods, relying on the surrogate model to generate objective function values based on a set of design variables. Reliance on a surrogate model is not necessary for the ABBS, and agents could instead run a more costly high-fidelity simulation to obtain objective function values. The ABBS can utilize multiple types of KA-S agents for both global and local searches of the objective space. The benefit of the ABBS is the ability to utilize multiple agent types to solve the problem. This allows for optimization modules from various sources to be incorporated into the ABBS with relative ease.

Two approaches are used as a global search method: stochastic and latin hypercube. The stochastic method can also be used by randomly selecting each design variable value based on a uniform distribution [15]. The latin hypercube method generates a set of designs based on a hypergrid, which adequately samples the entirety of the design space [31]. The Python module

`pydoe` is used to generate the hypercube structure [32].

For local searches, the steepest-ascent hill-climbing algorithm, neighborhood search algorithm, and a modified Nondominated Sorting Genetic Algorithm II (NSGA-II) are used. The hill-climbing algorithm selects an approximate optimal point and then makes incremental steps towards design variables that further optimizes the objective functions [33]. The neighborhood search algorithm selects an approximate optimal point, applies a small step change to a design variable, and determines the objectives for the new design [33]. A modified NSGA-II algorithm from `pymoo` is also implemented with a fixed population size (set to 40) and the algorithm is terminated after a set number of function evaluations (set to 800) [14]. As the NSGA-II algorithm examines designs, it writes these designs to the Blackboard. Upon implementation of the full MABS, other agents will be able to utilize, in tandem, the designs that various KA-S agents produce. The local search methods select solutions present in the *Pareto Front Level* and perform a local search to further explore the PF. Utilizing multiple local and global search methods enable the rapid establishment of the PF, with subsequent examination to ensure the PF is thoroughly described.

Figure 3 shows how each of the KAs interact with the blackboard abstract levels from Figure 2. Dashed lines indicate a KA reading information from an abstract level to preform its actions. A solid line shows where a KA can write information gained through an action to the blackboard. The type of KA-S agents used for both local and global searches are shown in the respective blocks.

The CA generates the environment for the ABBS, initiates the BA, and begins the iterative process to discover the PF. This process is outlined in Fig. 4, with two representative KAs, where the KA-S is selected to execute. For a serial instance of the ABBS, once the BA receives the action complete message, the process is repeated. In this work, a convergence criteria is not used; instead, at the conclusion of a set number of function evaluations, the CA shuts down the problem. The HDF5 file will then consist of the three abstract levels, where we can easily examine the PF, viable solutions, and reactor data.

The ABBS is a framework for solving multi-objective optimization problems using multiple agents and a central knowledge database. The blackboard system provides a convenient archiving tool for storing and examining the optimal core designs, viable core designs, and can be used to determine correlations among the design variables, objectives, and constraints. This is enabled because the blackboard is storing data on the various abstract levels in an HDF5 file. Post-

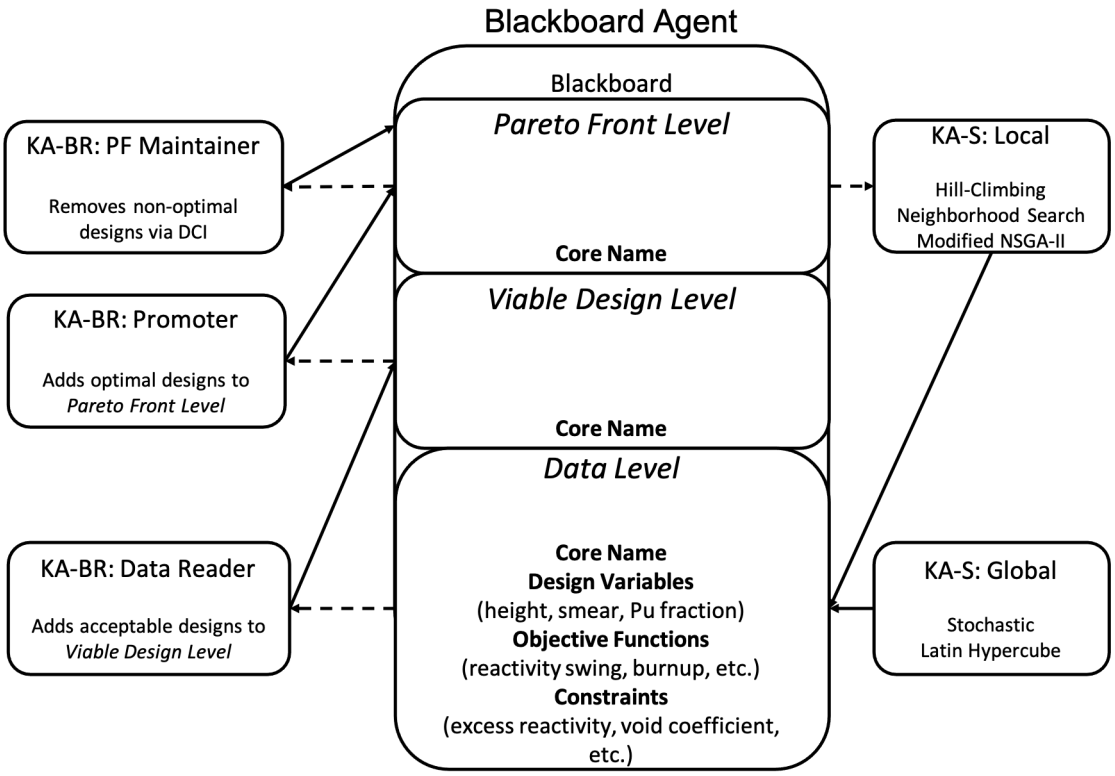


Fig. 3. Interaction of KAs with the Blackboard Abstract Levels.

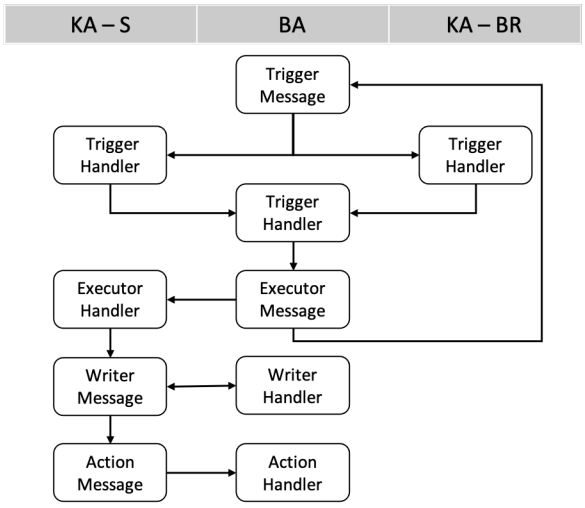


Fig. 4. ABBS Iteration for Communication between BA and KAs.

processing with the blackboard can reveal important relationships for future design studies and can be used to generate a surrogate model for additional studies.

To compare two PFs we use the the hypervolume (HV), DCI, and the Pareto front spread for each objective (PFS–obj). The HV gives the volume of the objective space that is dominated by the set of solutions on the Pareto front, and is used to measure the distribution of the PF; a larger HV indicates a more diverse PF [34]. Each objective function is scaled to ensure each equal contribution to the HV. In addition to our use of the DCI in the previous sections, the DCI can also be used to compare the diversity of multiple PFs by combining and placing them in the hyperbox grid environment [30]. The DCI ranges from 0 to 1, where a larger value denotes a more diverse PF with respect to the PFs examined. The maximum PFS examines the range of each objective function [35]. The ABBS and NSGA-II were executed 20 separate times (initialized with a unique random number seed each time). The PFs from each execution were examined to determine the average HVI, DCI, and PFS. These metrics will be compared to examine the abilities of the ABBS. Both algorithms were executed until 5,000 function evaluations were completed (i.e. 5,000 designs were analyzed). A full study on the convergence properties is not included in this paper, but will instead be left for a more detailed analysis of the MABS.

III. RESULTS

III.A. Benchmark Problems

To verify that the ABBS is able to solve an optimization problem, three multi-variable, multi-objective benchmark problems are used. The benchmarks are the first three problems in the ZDT benchmark suite [35]. ZDT1 has a convex Pareto front, ZDT2 has a concave Pareto front, and ZDT3 has a discontinuous Pareto front. Each problem has three design variables (x_n) ranging from 0 to 1. The benchmark objectives are constructed using Eq. 3.

$$\begin{aligned} &\min f_1(x) \\ &\min f_2(x) = g(x)h(f_1(x), g(x)) \end{aligned} \tag{3}$$

For ZDT1-3, the functions for $f_1(x)$ and $g(x)$ are constant, as seen in Eq. 4 and Eq. 5. The h function varies and can be seen in Eq. 6–8.

$$f_1(x) = x_1 \tag{4}$$

$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{5}$$

$$\text{ZDT1} \quad h(f_1, g) = 1 - \sqrt{f_1/g} \tag{6}$$

$$\text{ZDT2} \quad h(f_1, g) = 1 - (f_1/g)^2 \tag{7}$$

$$\text{ZDT3} \quad h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g)sin(10\pi f_1) \tag{8}$$

Figure 5 shows the Pareto fronts obtained by the ABBS for each of the benchmark problems. The ABBS is able to converge the known PF for each of the benchmark problems. For ZDT1 and ZDT2, ABBS converges to the true PF. For ZDT3, the ABBS is able to identify each of the discontinuous portions of the PF but does not fully converge to the true PF.

Table V provides quantitative evidence of the ABBS ability to converge to the PF by examining the mean and standard deviation (SD) for the generational distance (GD), inverse generational distance (IGD), and the HVI. The GD examines the Euclidean distance between solutions on the

TABLE V
Mean (SD) of GD, IGD, and HV for ABBS and NSGA-II

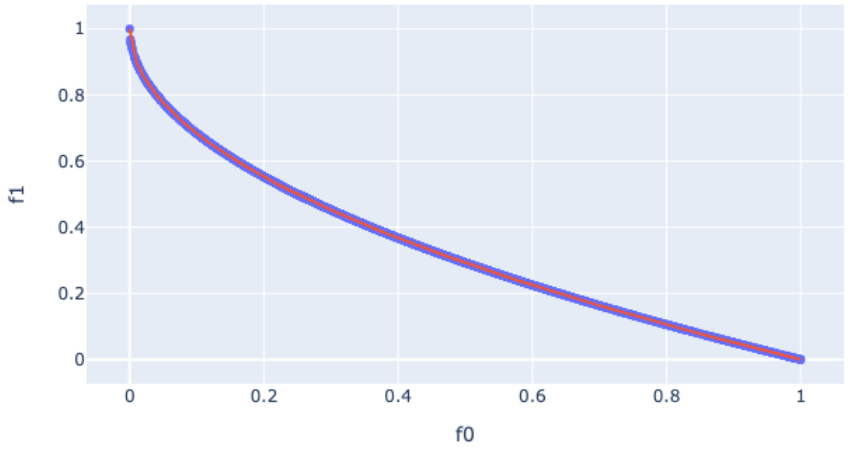
	ABBS			NSGA-II		
	GD	IGD	HVI	GD	IGD	HVI
ZDT1	5.225e-03	4.500e-03	6.609e-01	4.534e-03	5.973e-04	6.658e-01
	(5.435e-04)	(2.601e-04)	(1.594e-04)	(2.095e-04)	(4.559e-05)	(7.886e-05)
ZDT2	4.001e-03	4.610e-03	3.277e-01	4.183e-03	5.088e-04	3.326e-01
	(3.144e-04)	(2.337e-04)	(1.858e-04)	(1.285e-04)	(8.549e-05)	(8.762e-05)
ZDT3	5.516e-03	5.242e-03	1.041e+00	6.568e-03	1.080e-03	1.044e+00
	(3.670e-04)	(3.053e-04)	(1.415e-04)	(5.158e-04)	(1.809e-04)	(6.328e-05)

PF and the nearest point on the optimal PF [36]. The GD is used to examine how far solutions presented by the algorithm are from the true PF, where a smaller value indicates that the problem has more closely converged to the true PF. The IGD is similar to the GD however, we instead examine the Euclidean distance between solutions on the optimal PF and the algorithms PF [37]. The IGD measures an algorithms convergence and diversity of solutions on the PF, as we are ensuring that there are adequate points surrounding the optimal solutions on the true PF. A smaller value for the IGD indicates a more converged and diverse solution. The ABBS and NSGA-II algorithm were both run 20 times to generate an average value for each of the three metrics. For all three benchmark problems, the ABBS and NSGA-II algorithms yield similar values for each of the three performance metrics.

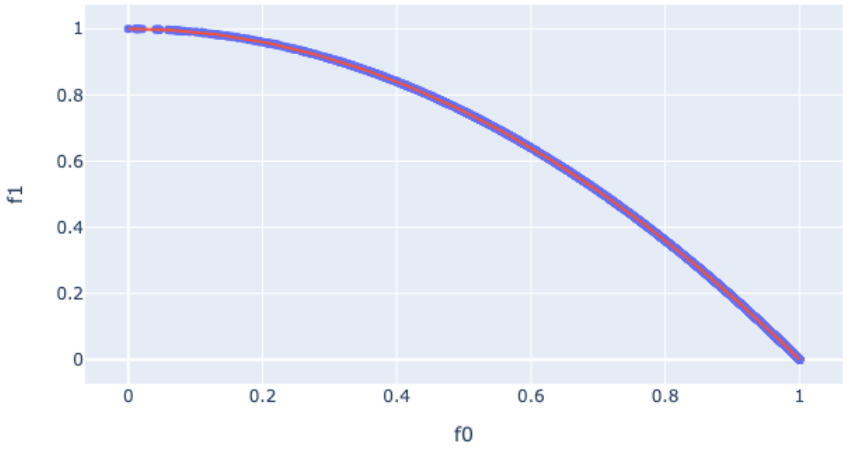
The ability of the ABBS to locate and explore the PF for the benchmark problems indicates that it can be applied to an SFR core design problem. For SFR core design, objective functions are likely to be smooth and continuous, similar to ZDT1 and ZDT2, and the ABBS should be able to quickly converge on the Pareto front.

III.B. SFR Optimization I

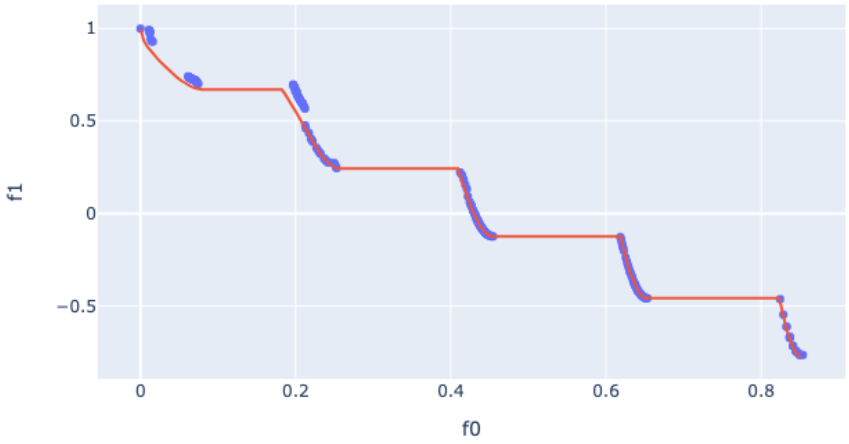
The ABBS is first used to examine a two-objective optimization problem for an SFR core. The objectives are the burnup and reactivity loss, given a fixed cycle length of 120 days. At 120 days, the core must have a positive excess reactivity. The constraints and objectives, along with their ranges, are shown in Table VI. The ranges for burnup and the reactivity swing adhere closely to both conventional limits for fuels and previous optimization studies for small SFR cores [24, 38]. This provides a pedagogical problem for reactor design where burnup increases the fuel consumption, which increases the reactivity loss.



(a) PF for ZDT1.



(b) PF for ZDT2.



(c) PF for ZDT3.

Fig. 5. PF obtained by the ABBS for ZDT Benchmarks 1-3. The ABBS PF is seen in blue. The True PF is seen in red.

TABLE VI
Constraints and Objectives for SFR Optimization Problem I.

Constraint	Range	Objective	Range	Goal
Excess Reactivity (pcm)	>0	Reactivity Swing (pcm/month)	0–1500	Min.
		Burnup (GWd/MT)	0–150	Max.

Table VII presents the performance metrics for the NSGA-II and ABBS algorithms when searching the SFR objective space using the surrogate model. Values are presented with one standard deviation in parenthesis. For each performance metric, we find that the ABBS and NSGA-II algorithms are statistically similar. Both algorithms are able to search the design space and find designs that span the objective space. The ABBS is not able to take advantage of agents working in tandem because of its serial execution. Introducing a multi-agent environment would help further explore the objective space.

TABLE VII
Performance Metrics for SFR Optimization I.

Approach	HVI	DCI	PFS–RS	PFS–BU
ABBS	0.3732 (0.0154)	0.847 (0.051)	748 (54)	57.9 (3.2)
NSGA-II	0.3763 (0.0086)	0.798 (0.028)	760 (34)	58.9 (2.0)

Fig. 6 shows the PF for the two objectives, where the excess reactivity constraint can be seen in the legend. As expected, we observe that increasing the burnup causes a corresponding increase in the reactivity swing, leading to a large PF. Fig. 7 shows the distribution of optimal cores in the design space. For each design variable, there is a region of solutions under which a "majority" of the solutions lie. Each design variable distribution also has a large tail which extends across the entire design space. Both the tail and high probability regions can be explained when looking at Fig. 8. We observe that the PF creates two regions of interest in the design space. The first region has little to no plutonium in the fuel, and has a fuel smear of nearly 0.80. This region corresponds to solutions on the PF with lower reactivity swing and burnup. The second region includes designs with a fuel height of 50 cm and with lower values of the fuel smear, causing a corresponding increase in the plutonium fraction. These cores tend to have a shorter core height, reduced fuel smear, and increased plutonium fraction to yield a higher burnup while maintaining a critical core.

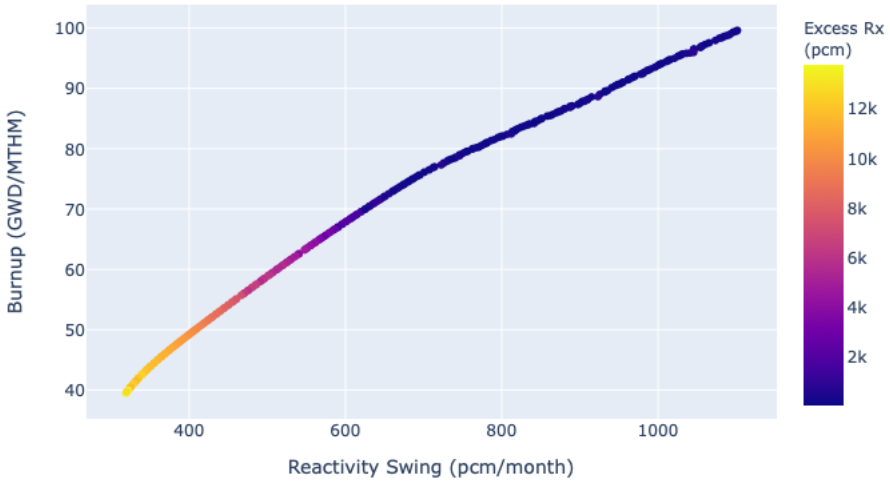
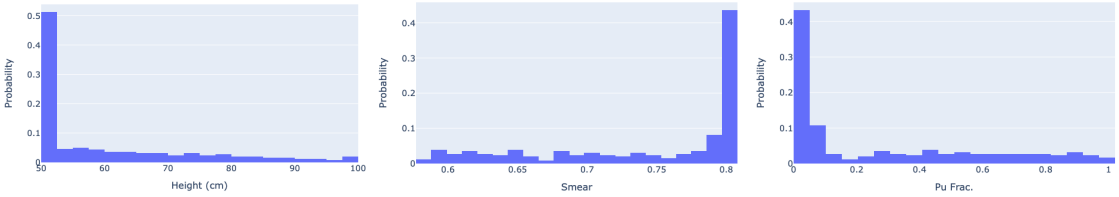


Fig. 6. Pareto Front for SFR Optimization I.



(a) Distribution for Height (b) Distribution for Smear (c) Distribution for Pu Frac

Fig. 7. Probability Distribution of Design Variables for SFR Optimization I.

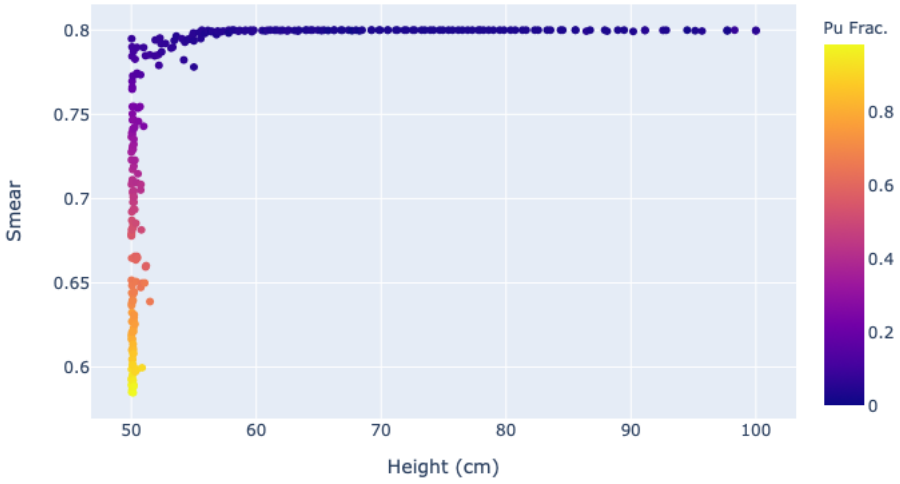


Fig. 8. Design Space for SFR Optimization I.

III.C. SFR Optimization II

We now examine a reactor design optimization problem with three objectives: the burnup, reactivity loss, and plutonium mass. For SFR Optimization II, we minimize the reactivity swing, maximize the burnup, and minimize the plutonium mass required per year. It is assumed that using plutonium as a fuel source is expensive and minimizing the plutonium mass is meant to reduce the cost of refueling. To provide a more realistic core design, we include four constraints: the Doppler coefficient, void coefficient, excess reactivity, and maximum LHGR. For the Doppler and void coefficient, we want to ensure a negative feedback coefficient, so we force both of these values to be negative. The objectives and constraints can be seen in Table VIII, along with their respective ranges. Limits placed on the reactivity swing, burnup, max linear heat generation rate, and plutonium mass were based on previous work examining the design and optimization of small SFR cores [24, 38, 39, 40]. The constraint on excess reactivity is large, as the core has all fresh fuel and must transition to a core with irradiated fuel in future cycles. Allowing for a large amount of excess reactivity will also allow assemblies to be removed if necessary. The cycle length was maintained at 120 days.

TABLE VIII
Constraints and Objectives for SFR Optimization Problem II.

Name	Range	Name	Range	Goal
Doppler Coeff. (pcm/K)	<0.0	Reactivity Swing (pcm/month)	0–1500	Min.
Void Coeff. (pcm/%void)	<0.0	Burnup (GWd/MT)	50–150	Max.
Excess Reactivity (pcm)	1,000–10,000	Pu Mass (kg/cycle)	0–1000	Min.
Max LHGR (kW/m)	<45.0			

Table IX presents the performance metrics for the GA and ABBS. Similar to the two-dimensional problem, we find that the ABBS and NSGA-II algorithms are similar in each metric. Despite the increase in objectives and constraints placed on the problem, both algorithms discover the PF. We expect that introducing a full MABS will increase the fidelity of the PF, when compared with other optimization algorithms.

TABLE IX
Performance Metrics for SFR Optimization II.

Approach	HVI	DCI	PFS–RS	PFS–BU	PFS–PM
ABBS	0.2851 (0.0029)	0.484 (0.012)	645 (14)	45.8 (0.9)	111.4 (2.2)
NSGA-II	0.2954 (0.0055)	0.522 (0.027)	764 (31)	59.2 (1.8)	106.9 (5.5)

The CPU time is not examined as a performance metrics for this study, however it is important to mention. For SFR Optimization I and II, the ABBS is between 20–30 times slower than the NSGA-II algorithm. This result is expected, and there are two driving mechanisms: message passing between agents, and writing the optimization problem to an HDF5 file. Message passing is not required in traditional optimization algorithms and requires additional time to perform each action. In a parallel system, agents will be able to perform tasks simultaneously which will help alleviate some of the burden associated with message passing. Along with this, while agents are examining the PF, the blackboard will be able to update the HDF5 file, where agents will not be required to wait for this process. The requirements for a multi-agent environment require and initial upfront computational cost, however, implementing a full-parallel system is envisioned to help alleviate many of these costs.

Fig. 9 shows the PF for SFR Optimization II. The PF remains largely similar to the PF presented in Fig 6, but we find that solutions below a burnup of 50 GWD/MTU are no longer present. The addition of the plutonium mass objective does not significantly change the shape of the PF, as we found multiple designs in SFR Optimization I which prioritize a uranium-based fuel. The addition of constraints placed on the problem have a strong effect on the available designs for the problem. Fig. 10 shows the 2D projection of the PF along with the constraints present for each design. This allows the reader to visualize the relationships between the objectives and constraints for this problem. Of significant importance is the relationship between the max LHGR and burnup. The max LHGR plays an inhibiting role for solutions which attempt to have a large burnup, as this requires a reduced core height and fuel smear.

Fig. 11 shows the distribution of the design variables. The design variables are distributed in a similar fashion to those from Fig. 8, however, they are not as sharply peaked. Fig. 12 shows all three design variables simultaneously, which provides a context for the distribution seen in Fig. 11. SFR Optimization II has similar shape for the design space as SFR Optimization I, however, the fuel height has increased from 50 cm to 55 cm because smaller fuel heights violate the max LHGR constraint. We also find a tail for fuel heights near 70–80 cm, where there is a decrease in the fuel smear attributed to decreasing the plutonium mass.

To glean additional information from the optimization process, a correlation matrix is formed from the designs on the PF and is presented in Fig. 13. We confirm suspicions from the previous

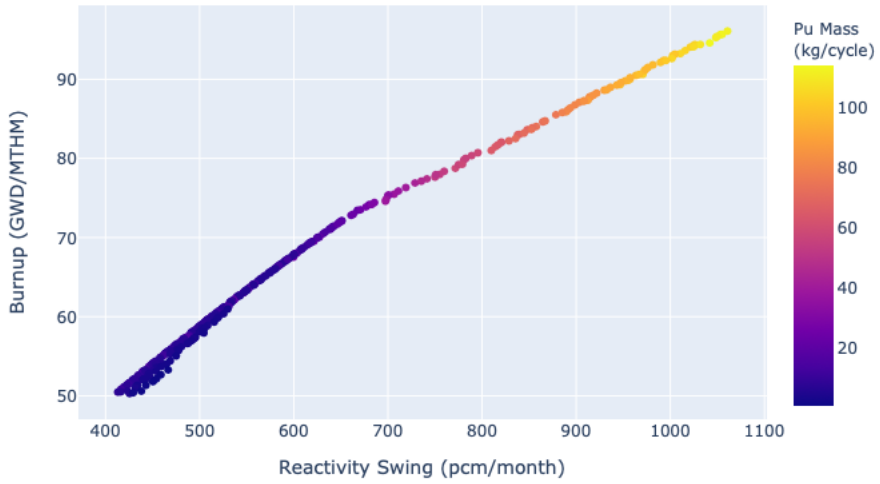


Fig. 9. Pareto Front for SFR Optimization II.

two SFR optimization problems that increasing the plutonium fraction causes an increase in both the reactivity swing and burnup. Similarly, there exists strong negative correlations with the fuel height and smear and the reactivity swing and burnup. These relationships are the driving factors behind the shape of the design space in Fig. 11. We also observe positive correlations between the max LHGR and the reactivity swing and burnup due to a flattening of the core height. Similarly, there is a strong negative correlation between the max LHGR and excess reactivity, as these cores tend to increase the core height. We also find strong negative correlations between the safety coefficients (void and Doppler) and the reactivity swing and burnup. The safety coefficients remain negative for all conditions; however, this relationship may be of consequence for future designers interested in larger cores.

The SFR Optimization I/II problems provide insight into the use of a non-plutonium based fuel for core design. This paper does not intend to favor a particular fuel source; however, through the use of an optimization algorithm, we find that enriched uranium provides a wide variety of optimal core designs. Uranium-driven designs provide some resemblance to known cores designs such as the ARC-100 [41]. In the ARC-100 design, a larger, taller core is required to provide a critical core due to the use of high assay low enriched uranium (HALEU) fuel (fissile wt% < 0.20).

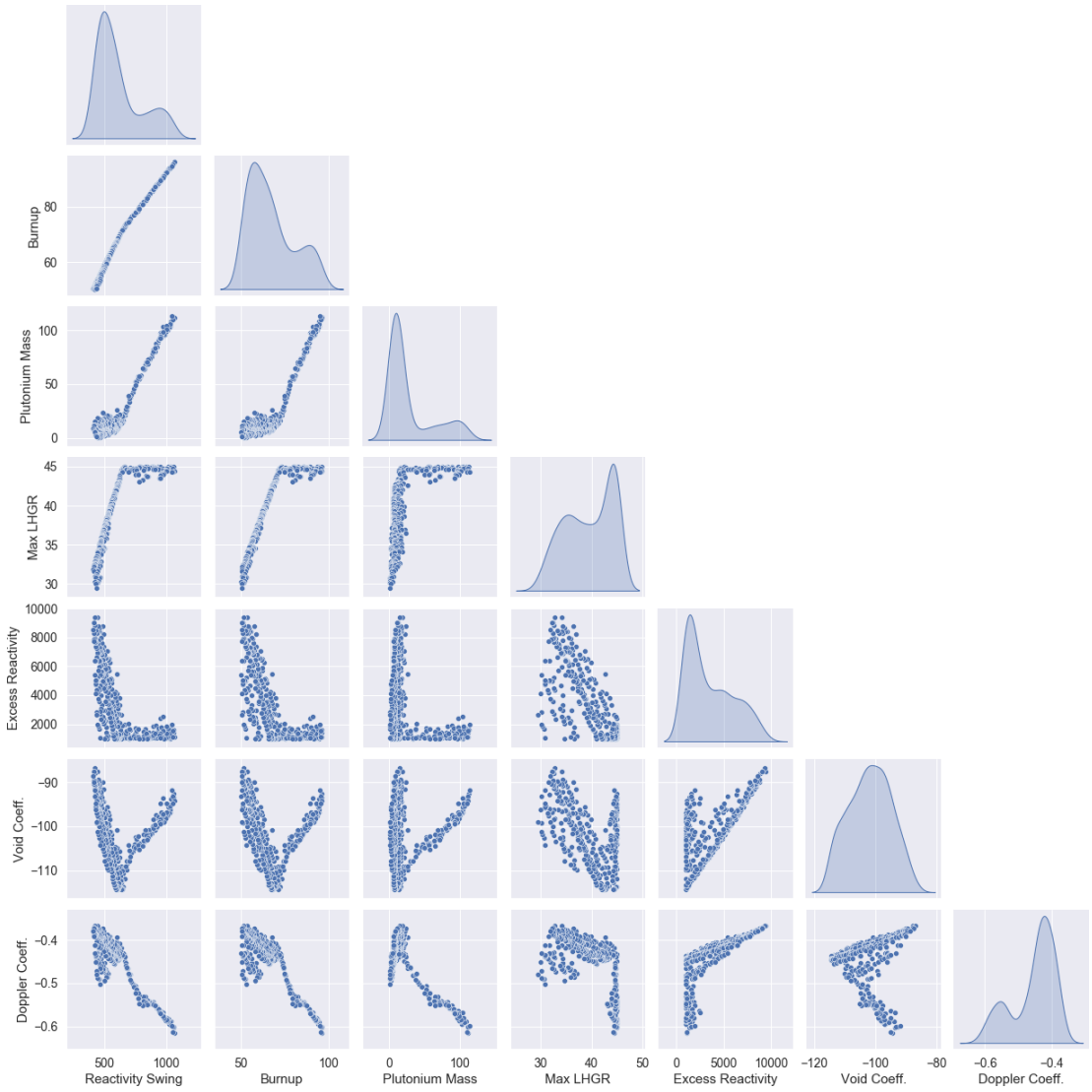
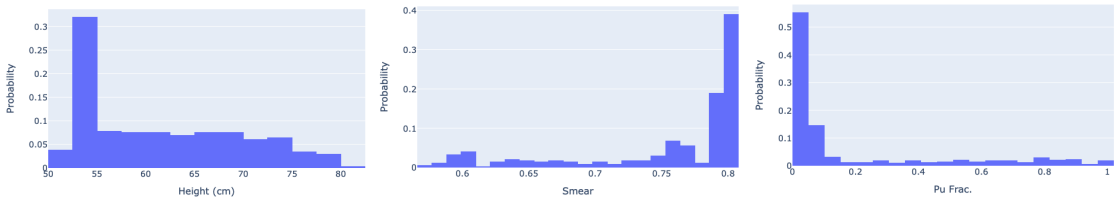


Fig. 10. 2D Projection of the PF and Constraints for SFR Optimization II.



(a) Distribution for Height (b) Distribution for Smear (c) Distribution for Pu Frac

Fig. 11. Probability Distribution of Design Variables for SFR Optimization II.

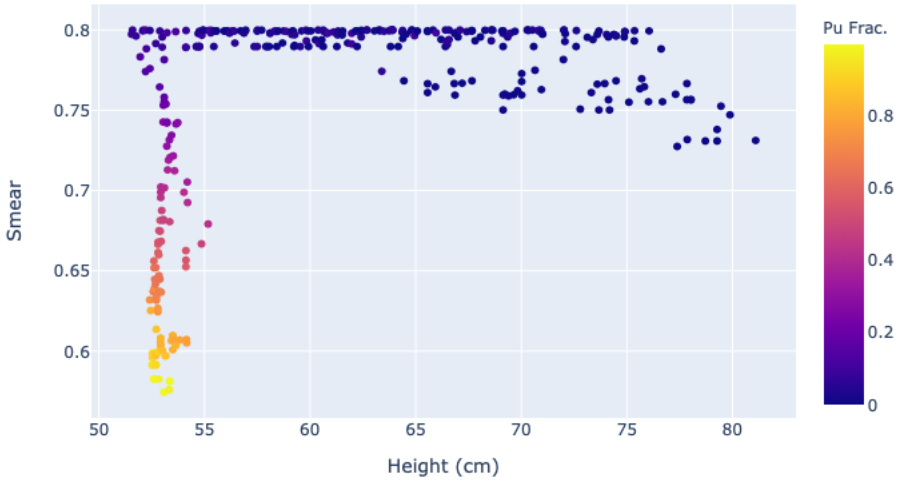


Fig. 12. Design Space for SFR Optimization II.

III.D. SFR Optimization III

The final optimization problem examines the ability of a small SFR to dilute and burn weapons-grade plutonium fuel. The objective functions are to maximize the amount of plutonium burned per cycle and minimize the ^{239}Pu/Pu ratio. The burnup objective function is updated to obtain a target burnup of 100 GWd/MT, rather than maximizing the burnup. `pymoo` does not currently have the ability to optimize to a target value, and as such, no comparison is made between the ABBS and NSGA-II. The objectives and constraints can be seen in Table X, along with their respective ranges. A new constraint, the ^{240}Pu/^{239}Pu ratio, is introduced to ensure a plutonium vector that is allowable under previous nonproliferation treaties [25]. The cycle length is increased to 240 days. The increase in cycle length is to promote plutonium dilution and disposition.

The PF for the SFR Optimization III problem can be seen in Fig. 14. We find that the maximum plutonium disposition rate corresponds well with a minimized ^{239}Pu/Pu ratio, and a

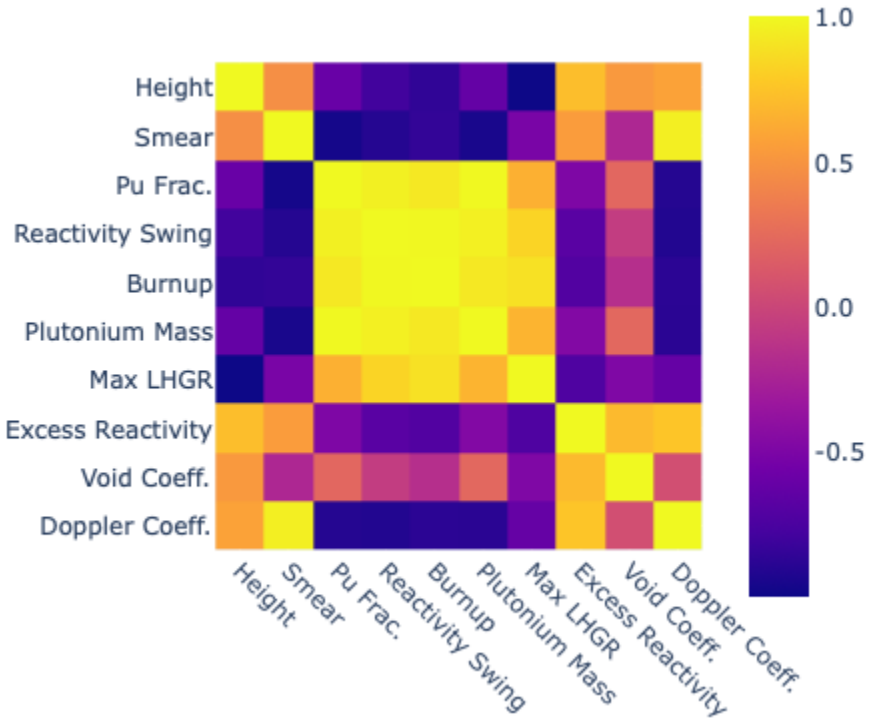


Fig. 13. Correlation Matrix for Objectives and Constraints for SFR Optimization II.

TABLE X
Constraints and Objectives for SFR Optimization Problem III.

Constraint	Range	Objective	Range	Goal
Reactivity Swing (pcm/month)	<1500	^{239}Pu/Pu Ratio	0.80–0.90	Min.
Doppler Coeff. (pcm/K)	<0.0	Pu Disposed (kg/cycle)	0–1000	Max.
Void Coeff. (pcm/%void)	<0.0	Burnup (GWd/MT)	50–150	Target : 100
Excess Reactivity (pcm)	1,000–10,000			
^{240}Pu/^{239}Pu	>0.1			
Max LHGR (kW/m)	<45.0			

maximum disposition rate of approximately 11.5 kg/cycle (17.5 kg/yr) is achieved. Table XI provides a comparison between the design selected by the ABBS and previous research on plutonium burner reactors. The Advanced Burner Test Reactor (ABTR) and Advanced Burner Reactor (ABR) are metallic-fueled SFRs designed specifically for burning transuranic waste (TRU) [40, 39]. The conversion ratio (CR) is used to distinguish the variants for the the ABTR and ABR core designs. Two variations of the ABTR (CR=0.7 and CR=0.25) and three variations of the ABR (CR=0.70, CR=0.5, and CR=0.25) were examined. For both the ABTR and ABR designs, the amount of disposed material is TRU rather than Pu. The BN-800 reactor is an oxide-fueled SFR

and was proposed to be used to reduce weapons-grade plutonium in Russia [25]. ABTR, ABR and the BN-800 designs all have variable fuel zoning, leading to a range of enrichment values.

The ABTR and ABR design with a CR near 0.7 are the two designs which are most similar to the ABBS design in regards to fuel enrichment, and disposition per MW-year. We find that the disposition per MW-year of the ABBS is within 15-30% of the ABTR and ABR designs with a CR near 0.7. This is expected given that the number of design variables was restricted and the core was not specifically designed as a burner core. The ABBS is able to generate core designs which dispose of plutonium at a rate comparable with previous burner technologies, despite not being specifically designed as a burner reactor.

Along with the plutonium that has been destroyed, an additional approximately 125 kg of weapons-grade plutonium has been degraded per cycle (190 kg/year). While other reactors such as the BN-800 are able to degrade significantly more plutonium (up to 1,790 kg/year), the designs found by the ABBS could provide a steady avenue for disposing of weapons-grade plutonium [25].

TABLE XI
Comparison of Disposition Rates Between Various Reactor Designs.

Reactor Design	Power (MWth)	Average BU (GWd/MTU)	Enrichment (%)	TRU/Pu Disposed (kg/year)	TRU/PU Disposed per MW (kg/MW-year)
ABBS Design	300	150	27	17.5	0.058
ABTR (CR=0.70)	250	78	18–23	20.4	0.080
ABTR (CR=0.25)	250	94	52–66	60.0	0.191
ABR (CR=0.75)	1000	100	16–24	70.6	0.071
ABR (CR=0.50)	1000	131	27–41	147.7	0.148
ABR (CR=0.25)	1000	171	46–69	231.9	0.232
BN-800	800	69	19-25	153.0	0.191

Fig. 15, shows the 2D projection of the PF along with the constraints present for each design. For brevity, the safety coefficients (Doppler and void) are not included, as they are not limiting factors in the core design. There appears to be a limit for plutonium disposition at 11.5 kg/cycle; however, this is a consequence of the upper limit we have placed on the burnup. The burnup limit also limits the ^{239}Pu/Pu ratio. The excess reactivity constraint nullifies many designs that have smaller plutonium disposition rates. Low plutonium disposition cores tend to have a much higher excess reactivity, as they favor fuel heights of approximately 100 cm with increasing fuel smear corresponding with a decreasing plutonium fraction. Since the fissile enrichment is nearly constant, increasing the fuel height and smear results in additional fuel being added to the core. Unlike the SFR Optimization II problem, the maximum LHGR does not play a significant role in

reducing the design space, as fuel heights below 55 cm are non-existent.

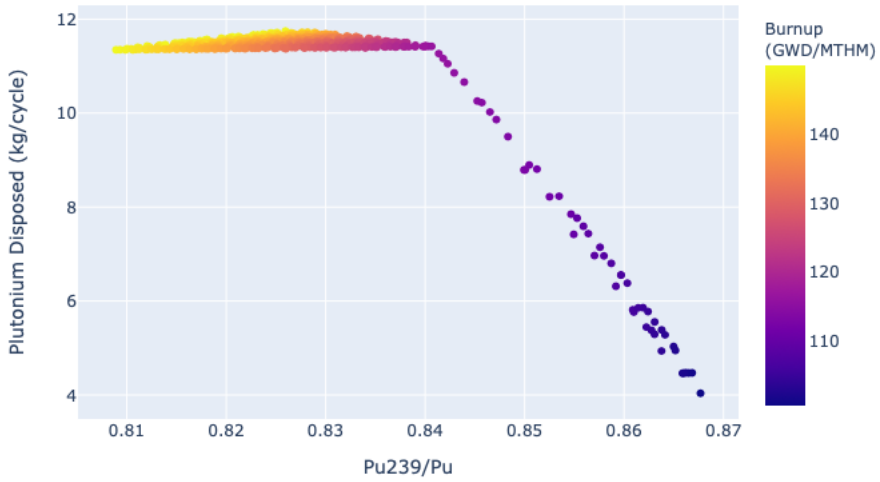


Fig. 14. Pareto Front for SFR Optimization III.

Fig. 16 and 17 show the distribution of the designs for the SFR Optimization III problem. We find that a majority of the solutions tend towards a plutonium fraction near 1.0 to maximize the plutonium present in the core. Corresponding with this are distributions of fuel height and smear values that maximize the plutonium in the core while maintaining the excess reactivity constraint. The vertical line of designs in Fig. 17 with a fuel height near 100 cm represents the trade off between fuel smear and plutonium fraction with a fixed fuel height. Designs with a fuel height of nearly 100 cm have limited solutions due to the constraint on the maximum amount of excess reactivity. Fig. 15 highlights this with the plateaus at 10,000 pcm for each plot examining the excess reactivity.

Fig. 18 shows the correlation matrix for the SFR Optimization III problem. From the correlation matrix, there is a strong correlation between the fuel height, plutonium fraction, and the objectives for SFR Optimization III. The plutonium fraction has the greatest effect on objectives related to the plutonium. Following this, the fuel height has a large effect on the objectives than the fuel smear. We find that the ^{239}Pu/Pu ratio and plutonium disposition rate both have strong correlations with other objectives and constraints, such as the burnup and reactivity swing. These relationships, along with data shown in Fig. 15, imply that, to build a core that can dispose of upwards of 10 kg of plutonium per cycle, a burnup level of approximately 120–130 must be achieved. As with SFR Optimization I/II, increasing the burnup can significantly increase the

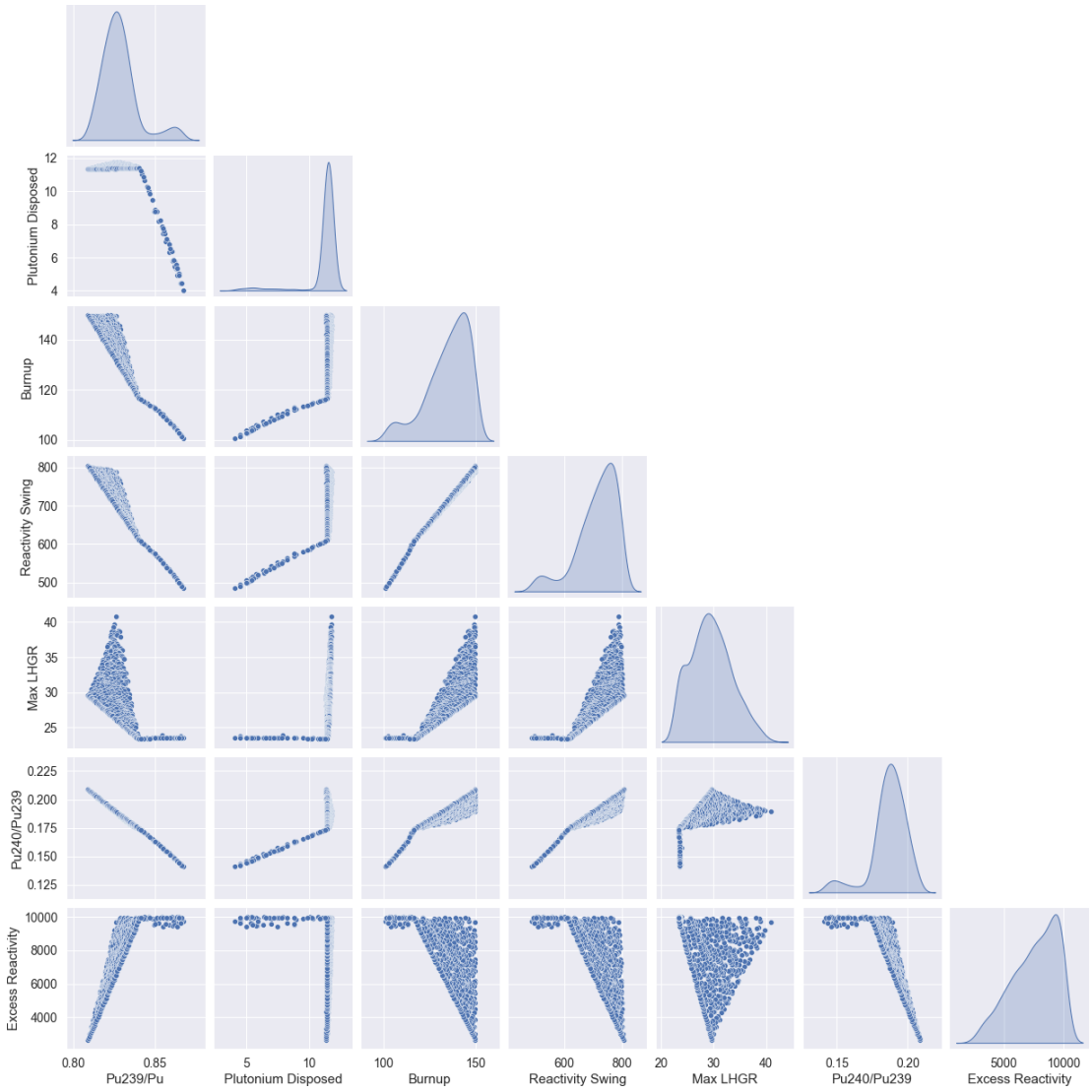
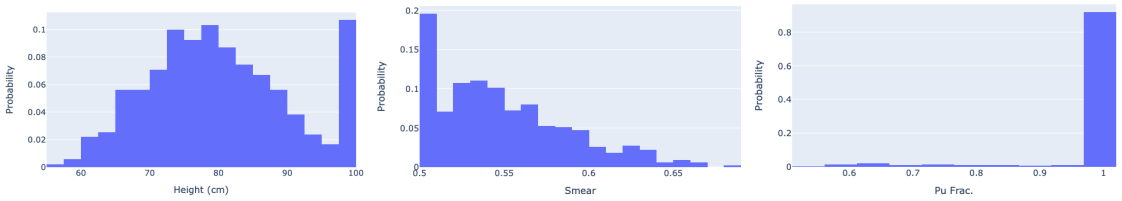


Fig. 15. 2D Projection of the PF and Constraints for SFR Optimization III.



(a) Distribution for Height (b) Distribution for Smear (c) Distribution for Pu Frac

Fig. 16. Probability Distribution of Design Variables for SFR Optimization III.

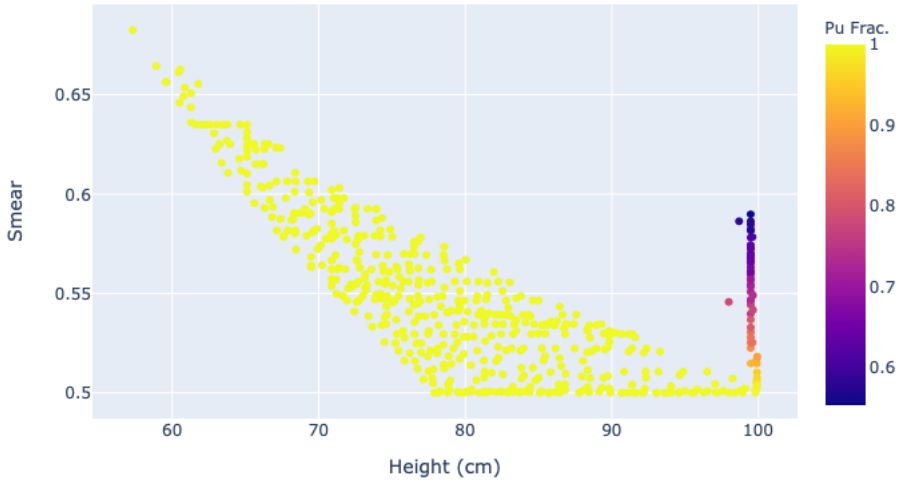


Fig. 17. Design Space for SFR Optimization III.

reactivity swing. Decreasing the burnup, however, causes an increase in the excess reactivity, such that core designs with a burnup closer to 100 GWd/MT have excess reactivity near 10,000 pcm.

We observe that the ABBS is able to find a PF for the design of a small SFR core, which could be utilized as a plutonium burner. The design of the plutonium burner core differs significantly from the previous SFR optimization problems, involving higher plutonium fractions and an larger fuel height and smear. A balance must still be struck to develop a core that is able to efficiently burn plutonium while adhering to limits for burnup, reactivity swing, and excess reactivity.

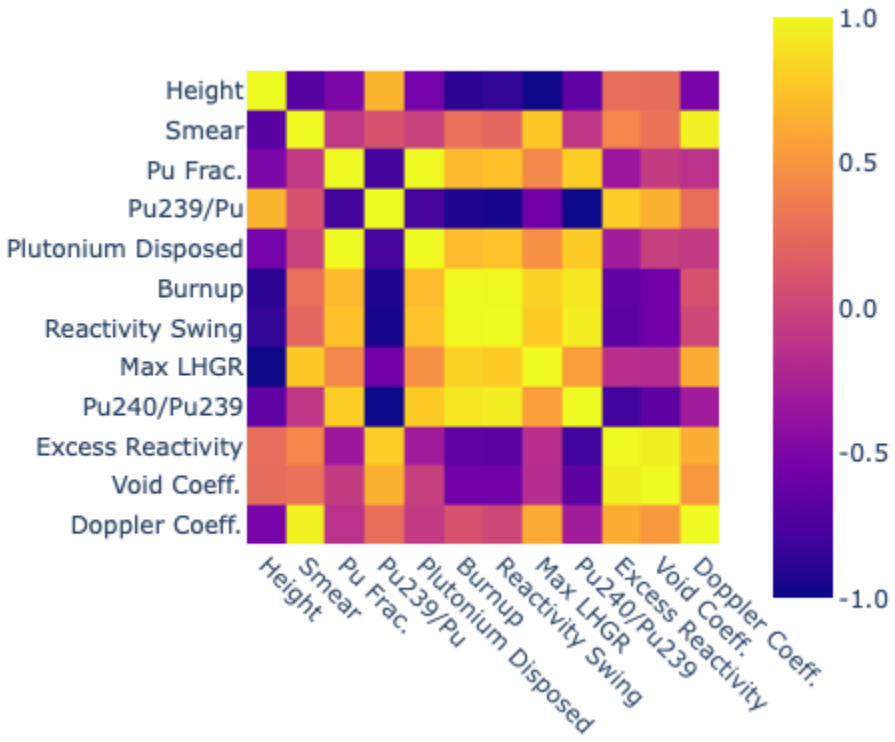


Fig. 18. Correlation Matrix for SFR Optimization III.

IV. CONCLUSIONS

In this paper we present a multi-objective optimization algorithm based on an ABBS. The ABBS was applied to three benchmark problems, and the implementation of the algorithm is demonstrated to effectively converge to a known PF. We then apply the ABBS to examine three reactor design optimization problems. The first two optimization cases examine a core design that is consistent with a small SFR, where we try to maximize the burnup of the fuel, minimize the reactivity swing, and reduce the amount of plutonium required to operate the core. The optimization process is compared against the NSGA-II algorithm, where we find that the ABBS converges to the same PF. When comparing the hyper-volume and diversity comparison indicator, the ABBS and NSGA-II are statistically similar. After the optimization process we are able to utilize the database created to examine the various correlations among the design variables and the objectives and constraints.

The last optimization problem examined the ability for a small SFR to burn weapons-grade plutonium. Through this process, we found that, for a 240 day cycle with four batches, we are able to dispose a maximum of nearly 12 kg of weapons-grade plutonium, and degrade another 125 kg of plutonium. The remaining degraded plutonium then has an plutonium vector of ^{239}Pu/Pu < 0.90, indicating that the plutonium is no longer of weapons-grade material.

The ABBS is the first step in creating a full MABS, where agents can act independently and learn from solutions presented by other agents in real time. The overhead associated with message passing and the continual archiving of the blackboard to an HDF5 file requires an order of magnitude more time than the NSGA-II algorithm when solving the same problem. Future work will explore the parallelizability of the MABS on high-performance computing environments to help reduce the time requirements for solving a multi-objective optimization problem. This could allow the MABS to utilize high-fidelity simulations, rather a less computationally expensive surrogate model. Using a high performance computing environment would allow for agents to submit high-fidelity simulations to explore the design space, while other agents examine areas of interest on the PF or move designs around on the blackboard. The ability of a MABS to utilize high-fidelity simulations and work in parallel would significantly reduce the effects of message passing and blackboard archiving seen in the first two optimization studies.

REFERENCES

[1] K. Inoue, "Fast-Reactor-Core Design Optimization by Linear Programming," *Nuclear Science and Engineering*, **39**, *3*, 394 (1970); 10.13182/NSE70-A20000., URL https://doi.org/10.13182/NSE70-A20000.

[2] G. T. Parks, "An Intelligent Stochastic Optimization Routine for Nuclear Fuel Cycle Design," *Nuclear Technology*, **89**, *2*, 233 (1990); 10.13182/NT90-A34350., URL https://doi.org/10.13182/NT90-A34350.

[3] A. A. de Moura Meneses, M. D. Machado, and R. Schirru, "Particle Swarm Optimization applied to the nuclear reload problem of a Pressurized Water Reactor," *Progress in Nuclear Energy*, **51**, *2*, 319 (2009); https://doi.org/10.1016/j.pnucene.2008.07.002., URL https://www.sciencedirect.com/science/article/pii/S0149197008001078.

[4] O. Fabbris, S. Dardour, P. Blaise, J.-H. Ferrasse, and M. Saez, "Surrogates Based Multi-Criteria Predesign Methodology of Sodium-Cooled Fast Reactor Cores – Application to CFV-Like Cores," *Nuclear Engineering and Design*, **305**, 314 (2016); https://doi.org/10.1016/j.nucengdes.2016.05.021.

[5] K. D. Huff, M. J. Gidden, R. W. Carlsen, R. R. Flanagan, M. B. McGarry, A. C. Opotowsky, E. A. Schneider, A. M. Scopatz, and P. P. Wilson, "Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework," *Advances in Engineering Software*, **94**, 46 (2016); https://doi.org/10.1016/j.advengsoft.2016.01.014., URL http://www.sciencedirect.com/science/article/pii/S0965997816300229.

[6] D. Hulse, B. Gigous, K. Tumer, C. Hoyle, and I. Y. Tumer, "Towards a Distributed Multiagent Learning-Based Design Optimization Method," vol. 2A: 43rd Design Automation Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2017); 10.1115/DETC2017-68042., URL https://doi.org/10.1115/DETC2017-68042, v02AT03A008.

[7] N. F. Soria Zurita, M. K. Colby, I. Y. Tumer, C. Hoyle, and K. Tumer, "Design of Complex Engineered Systems Using Multi-Agent Coordination," *Journal of Computing*

and Information Science in Engineering, **18**, *1* (2017); 10.1115/1.4038158., URL https://doi.org/10.1115/1.4038158, 011003.

[8] D. Hulse, K. Tumer, C. Hoyle, and I. Tumer, "Modeling multidisciplinary design with multiagent learning," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **33**, *1*, 85–99 (2019); 10.1017/S0890060418000161.

[9] D. Corkill, "Collaborating Software: Blackboard and Multi-Agent Systems & the Future," *Proceedings of the International Lisp Conference*, New York, New York (2003).

[10] S. Ljasenko, N. Lohse, L. Justham, I. Pereira, and M. Jackson, "A Self-Organisation Model for Mobile Robots in Large Structure Assembly Using Multi-Agent Systems," *Service Orientation in Holonic and Multi-Agent Manufacturing*, 83–91, Springer International Publishing (2017).

[11] D. Vallejo, J.J.Castro-Schez, C. Glez-Morcillo, and J. Albusac, "Multi-Agent Architecture for Information Retrieval and Intelligent Monitoring by UAVs in Known Environments Affected by Catastrophes," *Engineering Applications of Artificial Intelligence*, **87** (2020); https://doi.org/10.1016/j.engappai.2019.103243.

[12] A. Lilia Laureano-Cruces et al, "Multi-Agent System for Real Time Planning Using Collaborative Agents," *International Journal of Intelligence Science*, **4**, 91 (2014); http://dx.doi.org/10.4236/ijis.2014.44011.

[13] R. Stewart, "Multi-Agent Blackboard System," https://github.com/ryanstwrt/multi_agent_blackboard_system (2021).

[14] J. Blank and K. Deb, "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, **8**, 89497 (2020).

[15] J. Arora, *Introduction to Optimum Design*, Elsevier (2013).

[16] D. Whitley, "A Genetic Algorithm Tutorial," *Statistics and Computing*, **4**, 65 (1994).

[17] D. D. Corkill, K. Q. Gallagher, and P. M. Johnson, "Achieving Flexibility, Efficiency, and Generality in Blackboard Architectures," *Readings in Distributed Artificial Intelligence*, 541–546, Morgan Kaufmann; https://doi.org/10.1016/B978-0-934613-63-7.50053-X.

[18] I. Craig, *Blackboard Systems*, Alex Publishing Corporation (1995).

[19] G. J. Youinou, S. E. Bays, and G. Palmiotti, "Scoping Analysis of Sodium Cooled Fast Spectrum Test Reactor Cores," INL/EXT-20-57186-Rev000, Idaho National Laboratory (2020); 10.2172/1598332.

[20] C. J. Werner, J. S. Bull, C. J. Solomon, F. B. Brown, G. W. McKinney, M. E. Rising, D. A. Dixon, R. L. Martz, H. G. Hughes, L. J. Cox, A. J. Zukaitis, J. C. Armstrong, R. A. Forster, and L. Casswell, "MCNP6.2 Release Notes," LA-UR-18-20808, Los Alamos National Laboratory (2018).

[21] R. Stewart, "FRIDGe: Fast Reactor Input Deck Generator," *Journal of Open Source Software*, **4**, *40* (2019); https://doi.org/10.21105/joss.01486.

[22] R. Stewart, "Sodium Fast Reactor Database (v0.2)," (2019); 10.5281/zenodo.3464101., URL https://doi.org/10.5281/zenodo.3464101.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, **12**, 2825 (2011).

[24] D. Porter and H. Tsai, "Full-length U–xPu–10Zr (x=0, 8, 19wt.%) fast reactor fuel test in FFTF," *Journal of Nuclear Materials*, **427**, *1*, 46 (2012); https://doi.org/10.1016/j.jnucmat.2012.03.047.

[25] M. E. Moritz Kütt, Friederike Friess, "Plutonium Disposition in the BN-800 Fast Reactor: An Assessment of Plutonium Isotopics and Breeding," *Science & Global Security*, **22**, *3*, 188 (2014).

[26] "Unofficial Composite Text Of The United States-Russian Federation 2000 Plutonium Management And Disposition Agreement, As Amended," State Department, Washington, DC, USA (2000)URL https://2009-2017.state.gov/documents/organization/213493.pdf.

[27] K. Daugirdas and J. D. Mortenson, "Russia Suspends Bilateral Agreement with United States Disposal of Weapons-Grade Plutonium," *The American Journal of International Law*, **111**, *1*, 181 (2017)URL https://www.jstor.org/stable/26568831.

[28] "osBrain," https://osbrain.readthedocs.io/en/stable/ (2019).

[29] The HDF Group, "Hierarchical data format version 5," (2000-2020)URL http://www.hdfgroup.org/HDF5.

[30] M. Li, S. Yang, and X. Liu, "Diversity Comparison of Pareto Front Approximations in Many-Objective Optimization," *IEEE Transactions on Cybernetics*, **44**, *12*, 2568 (2014).

[31] M. D. McKay, R. J. Beckman, and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, **21**, *2*, 239 (1979).

[32] "pyDOE: Design of Experiments for Python," (2013)URL https://pythonhosted.org/pyDOE/index.html.

[33] M. A. Lones, "Metaheuristics in Nature-Inspired Algorithms," *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO Comp '14, 1419–1422, Association for Computing Machinery, New York, NY, USA (2014); 10.1145/2598394.2609841.

[34] L. Bradstreet, "The hypervolume indicator for multi-objective optimisation: calculation and use," PhD Thesis, the University of Western Australia (2011).

[35] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," PhD Thesis, Swiss Federal Institute of Technology Zurich (1999).

[36] D. V. Veldhuizen and G. Lamont, "Multiobjective Evolutionary Algorithm Research: A History and Analysis," TR-98-03, Air Force Institute of Technology (1998).

[37] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 242 (2002).

[38] K. Zeng et al, "Development of Multi-Objective Core optimization Framework and Application to Sodium-Cooled Fast Test Reactors," *Progress in Nuclear Energy*, **120**, 103184 (2020); https://doi.org/10.1016/j.pnucene.2019.103184.

[39] E. Hoffman, W. Yang, and R. Hill, "Preliminary Core Design Studies for the Advanced Burner Reactor over a Wide Range of Conversion Ratios," IANL-AFCI-177, Argonne National Laboratory (2006).

[40] Y. Chang, P. Finck, and C. Grandy, "Advanced Burner Test Reactor Preconceptual Design Report," IANL-AFCI-173, Argonne National Laboratory (2006).

[41] D. C. Wade and L. Walters, "ARC-100: A Sustainable, Modular Nuclear Plant for Emerging Markets," *Proceedings of ICAPP '10*, – (2010).