

© 2022 by Gwendolyn Jin Yi Chee. All rights reserved.

FLUORIDE-SALT-COOLED HIGH-TEMPERATURE REACTOR GENERATIVE
DESIGN OPTIMIZATION WITH EVOLUTIONARY ALGORITHMS

BY

GWENDOLYN JIN YI CHEE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Nuclear, Plasma, and Radiological Engineering
with a concentration in Computational Science and Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Doctoral Committee:

Research Scientist Madicken Munk, Chair
Associate Professor Tomasz Kozlowski
Professor James F. Stubbins
Assistant Professor Huy Trong Tran

Abstract

Additive manufacturing of reactor core components removes the geometric constraints required by conventional manufacturing, such as slabs as fuel planks and cylinders as fuel rods, enabling further optimization and improvement of core geometries. Wide-spread adoption of additive manufacturing methods in the nuclear industry could drastically decrease reactor fabrication costs, reduce deployment timelines, and improve reactor safety. Due to the expansion of the potential design space facilitated through additive manufacturing, reactor designers need to find methods, such as generative design, to explore the design space efficiently. Generative design is an iterative design exploration process; designers define design goals and constraints in a generative design software and the software explores all the possible permutations of a solution, quickly generating design alternatives. Fully benefiting from the new ability to 3D print reactor components requires further research into generative reactor design optimization. Generative reactor design optimization for arbitrary geometries enabled by additive manufacturing is a new concept, and few research demonstrations have been done to explore the large new design space.

In this dissertation, I apply evolutionary algorithms to conduct generative Advanced High-Temperature Reactor (AHTR) design optimization. First, I participated in the Organisation for Economic Co-operation and Development (OECD) Nuclear Energy Agency (NEA) Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark to further our understanding of the AHTR design's complexities and to gain an intuition for the unique physics of the system. Next, I created the Reactor evOLutionary aLgorithm Optimizer (ROLLO) Python package tool that enables generative reactor design optimization with evolutionary algorithms for non-conventional reactor geometries and fuel distributions. I then applied ROLLO to conduct generative reactor design optimization for AHTR plank and one-third assembly models. ROLLO generated AHTR designs with varying fuel amounts, fuel distributions, and coolant channel shapes that optimize for three

key reactor performance metrics: minimize the total fuel amount (PF_{total}), minimize maximum temperature (T_{max}), and minimize power peaking in the fuel (PPF_{fuel}).

I reported the FHR benchmark Phase I-A and I-B results, demonstrating the AHTR's passive safety behavior with negative temperature coefficients. A comparison of k_{eff} results between the reference case and the AHTR configuration with high heavy metal loading demonstrated that increased fuel packing does not always correspond with increased k_{eff} due to self-shielding effects. In addition, temperature modeling of the AHTR full assembly showed temperature peaks in the fuel stripes near the spacers, highlighting the impact of spacer material and location.

Next, I reported and discussed the AHTR plank and one-third assembly optimization results. I characterized each objective's driving factors and relationship with each input parameter from the results. The final and largest optimization problem is the one-third assembly multi-objective optimization that optimized for all three objectives (PF_{total} , T_{max} , and PPF_{fuel}) while varying all the input parameters (PF_{total} , TRISO distribution, and coolant channel shape), also known as simulation a-3b. Simulation a-3b ran for ~ 1800 node-hours on the Theta supercomputer with 6 generations and 128 reactor models per generation. It demonstrated 12 one-third assembly reactor models on its Pareto front that met all three objectives. The reactor models on the Pareto Front have different PF_{total} , TRISO distributions, and coolant channel shapes, depending on the extent each objective is minimized due to the nature of multi-objective optimization that results in a tradeoff between objectives. The results demonstrated ROLLO's success in conducting multi-objective generative reactor design optimization, and underscored the challenges of multi-objective design optimization.

In conclusion, through participation in the FHR benchmark, I contribute to deepening our understanding of the promising AHTR technology. By designing the ROLLO tool and demonstrating ROLLO's success in optimization of the AHTR beyond classical input parameters, I contribute to optimization tool development for reactors of the future. As additive manufacturing technology advances and the Transformational Challenge Reactor (TCR) program demonstrates the first 3D printed operational reactor, more reactor designers will begin to explore the vast design space enabled by 3D printing. ROLLO can be utilized to optimize other reactor types for arbitrary geometries and parameters, enabling further optimization and improvement of reactor geometries.

Acknowledgments

For this dissertation and all I have learned from my five years in graduate school, I am deeply grateful to many amazing professors and colleagues at the University of Illinois at Urbana-Champaign and Argonne National Laboratory. I want to express my deepest gratitude to my advisors, Dr. Kathryn Huff, and Dr. Madicken Munk. I am grateful to Professor Huff for her unconditional support over the years and for giving me so much freedom in pursuing different research avenues as I found my footing in the early years of graduate school. I am especially grateful for her ability to inspire me to look at the bigger picture of my work and holding me to a high standard in all aspects. I am grateful to Dr. Munk for her guidance and encouragement in my final year of graduate school as I wrote this dissertation. Her enthusiasm for discussing results and her experience and insights challenged me to gain a deeper understanding and appreciation for reactor physics. I would like to thank the professors who served on my committee: Dr. Tomasz Kozlowski, Dr. James Stubbins, and Dr. Huy Trong Tran, for the knowledge and guidance they provided in assessing this work. I also wish to thank my Argonne internship advisors, Dr. Bo Feng, and Dr. Adam Nelson, for their technical guidance and invaluable career advice.

I am thankful for the friendship of my fellow ARFC groupmates who not only made great collaborators but also made grad school fun – Sun Myung Park, Greg Westphal, Jin Whan Bae, Anshuman Chaube, Dr. Andrei Rykhlevkii, Sam Dotson, Nathan Ryan, Lu Kissinger, Amanda Bachmann, Nataly Panczyk, Olek Yardas, Luke Seifert, and Zoe Richter. Special thanks to Nathan Ryan and Nataly Panczyk for their excellent proofreading and comments on this work. I am also thankful to the wonderful NPREG, WiN, and GSAC communities at UIUC, where I found great friendships with many students and rapport with caring faculty and staff.

Finally, I want to thank my family, Joy, Gerard, and Ben, for their unwavering support and encouragement from day one. I will always be grateful to my mum for instilling in me that I

can achieve and be anything I set my heart and mind to, and my dad for cultivating my sense of curiosity and wonder for life.

This work was carried out in the Advanced Reactors and Fuel Cycles Group (ARFC) at the Nuclear, Plasma, and Radiological Engineering (NPRE) Department at the University of Illinois at Urbana-Champaign (UIUC). This research was supported by the Nuclear Regulatory Commission Faculty Development Program (award NRC-HQ-84-14-G-0054 Program B) and the Department of Nuclear, Plasma, and Radiological Engineering. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993), the State of Illinois, and as of December 2019, the National Geospatial-Intelligence Agency. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. This research was also supported by the Argonne Leadership Computing Facility's Director's Discretionary Allocation Program.

Table of Contents

Chapter 1 Introduction	1
1.1 Objectives and Outline	4
Chapter 2 Literature Review	7
2.1 Fluoride-Salt-Cooled High-Temperature Reactor System	7
2.1.1 Advanced High-Temperature Reactor Design	9
2.1.2 Previous AHTR Modeling Efforts and Challenges	10
2.1.3 FHR Benchmark	12
2.1.4 Modeling Software	13
2.2 Additive Manufacturing	15
2.2.1 Benefits of 3D Printing Reactor Components	16
2.2.2 Efforts toward 3D Printing Reactor Components	17
2.3 Nuclear Reactor Design Optimization	19
2.3.1 Reactor Optimization for Classical Parameters	21
2.3.2 Reactor Optimization for Arbitrary Parameters	23
2.4 Evolutionary Algorithms	25
2.4.1 Genetic Algorithms	26
2.4.2 Genetic Algorithm Hyperparameter Tuning	29
2.5 Summary	31
Chapter 3 Fluoride-Salt-Cooled High-Temperature Reactor Benchmark	32
3.1 FHR Benchmark Advanced High-Temperature Reactor Design	33
3.2 FHR Benchmark Phase I Specifications	38
3.2.1 Phase I-A Specifications	41
3.2.2 Phase I-B Specifications	41
3.3 FHR Benchmark Phase I Results	42
3.3.1 Results: Phase I-A	42
3.3.2 Results: Phase I-B	46
3.4 FHR Benchmark Temperature Model	48
3.4.1 Group Constant Generation	48
3.4.2 Moltres Mesh Generation	55
3.4.3 Temperature Model Setup and Results	57
3.5 Summary	62

Chapter 4 ROLLO: Reactor evOLutionary aLgorithm Optimizer	63
4.1 Evolutionary Algorithm Driver	65
4.1.1 Distributed Evolutionary Algorithms in Python	66
4.1.2 ROLLO Evolutionary Algorithm Implementation	67
4.2 ROLLO Input File	68
4.2.1 Control Variables	68
4.2.2 Evaluators	68
4.2.3 Constraints	71
4.2.4 Algorithm	71
4.3 ROLLO Software Architecture	72
4.3.1 Installing and Running ROLLO	74
4.3.2 ROLLO Parallelization	75
4.3.3 ROLLO Results Analysis	76
4.4 ROLLO Verification	77
4.4.1 Ackley Function	77
4.4.2 Binh and Korn Function	77
4.4.3 Pu-239 Critical Bare Sphere	79
4.5 ROLLO Convergence Criteria	83
4.5.1 Convergence	83
4.6 Summary	85
Chapter 5 AHTR Modeling and Optimization Methodology	86
5.1 ROLLO Optimization Problem Definitions	86
5.2 AHTR Geometry for Optimization Problem	87
5.2.1 AHTR Plank Geometry	88
5.2.2 AHTR One-Third Assembly Geometry	89
5.3 AHTR Model Workflow	90
5.3.1 Input Parameter Modeling	91
5.3.2 AHTR OpenMC and Moltres Models	96
5.3.3 Output Parameter Calculation	105
5.4 AHTR Moltres Model Verification	106
5.4.1 AHTR Plank: Key Neutronics Parameters Verification	106
5.4.2 AHTR One-Third Assembly: Key Neutronics Parameters Verification	112
5.4.3 AHTR Plank and One-Third Assembly: Mesh Refinement Studies	117
5.4.4 AHTR Plank and One-Third Assembly: Group Constant Temperature Study	118
5.5 ROLLO Hyperparameter Tuning	120
5.5.1 ROLLO Single-Objective Optimization Hyperparameters	121
5.5.2 ROLLO Multi-Objective Optimization Hyperparameters	126
5.6 Summary	128
Chapter 6 AHTR Plank Optimization Results	129
6.1 ROLLO AHTR Plank Optimization Simulations Overview	130
6.2 AHTR Plank: Single-Objective Optimization Results	132
6.2.1 Objective: Minimize Total Packing Fraction (PF_{total})	132
6.2.2 Objective: Minimize Maximum Plank Temperature (T_{max})	134
6.2.3 Objective: Minimize Fuel-Normalized Power Peaking Factor (PPF_{fuel})	138
6.2.4 Single-Objective Optimization Discussion	143

6.2.5	Single-Objective Optimization Major Takeaways	151
6.3	AHTR Plank: Double-Objective Optimization Results	151
6.3.1	p-2a: Minimize PF_{total} and T_{max}	153
6.3.2	p-2b: Minimize PF_{total} and PPF_{fuel}	155
6.3.3	p-2c: Minimize T_{max} and PPF_{fuel}	158
6.3.4	Double-Objective Optimization Discussion	161
6.3.5	Double-Objective Optimization Major Takeaways	167
6.4	AHTR Plank: Triple-Objective Optimization Results	168
6.4.1	p-3a: Variation of PF_{total} and $\rho_{TRISO}(\vec{r})$	168
6.4.2	p-3b: Variation of PF_{total} , $\rho_{TRISO}(\vec{r})$, and Coolant Channel Shape	169
6.4.3	Triple-Objective Optimization Discussion	174
6.4.4	Triple-Objective Optimization Major Takeaways	177
6.5	AHTR Plank: Computational Cost Summary	178
6.6	Summary	179
Chapter 7 AHTR One-Third Assembly Optimization Results	181
7.1	ROLLO AHTR One-Third Assembly Optimization Simulations Overview	182
7.2	AHTR One-Third Assembly: Single-Objective Optimization Results	184
7.2.1	Objective: Minimize Total Packing Fraction (PF_{total})	184
7.2.2	Objective: Minimize Maximum Temperature (T_{max})	188
7.2.3	Objective: Minimize Fuel-Normalized Power Peaking Factor (PPF_{fuel})	192
7.2.4	Single-Objective Optimization Discussion	199
7.2.5	Single-Objective Optimization Major Takeaways	206
7.3	AHTR One-Third Assembly: Double-Objective Optimization Results	207
7.3.1	a-2a: Minimize PF_{total} and T_{max}	208
7.3.2	a-2b: Minimize PF_{total} and PPF_{fuel}	212
7.3.3	a-2c: Minimize T_{max} and PPF_{fuel}	216
7.3.4	Double-Objective Optimization Discussion	219
7.3.5	Double-Objective Optimization Major Takeaways	226
7.4	AHTR One-Third Assembly: Triple-Objective Optimization Results	226
7.4.1	a-3a: Variation of PF_{total} and $\rho_{TRISO}(\vec{r})$	226
7.4.2	a-3b: Variation of PF_{total} , $\rho_{TRISO}(\vec{r})$, and Coolant Channel Shape	232
7.4.3	Triple-Objective Optimization Discussion	238
7.4.4	Triple-Objective Optimization Major Takeaways	246
7.5	AHTR One-Third Assembly: Computational Cost Summary	246
7.6	Summary	247
Chapter 8 Conclusions and Future Work	249
8.1	Limitations and Future Work	252
References	254
Appendix A: Reproducibility	263
Appendix B: Simulation a-3b-256	265

Chapter 1

Introduction

Access to electricity plays a vital role in improving standards of living, education, and health worldwide [1]. As the human population increases and developing nations rapidly industrialize, global energy demand will continue to rise, with worldwide electricity use projected to grow 45% by 2040 [1, 2]. Expanding access to energy while at the same time drastically reducing greenhouse gas (GHG) emissions that are responsible for climate challenge is among the central challenges confronting humankind in the 21st century [1]. The latest scientific report by the Intergovernmental Panel on Climate Change (IPCC) finds changes in the Earth's climate in every region and across the whole climate system as well [3]. Increased global surface temperatures, sea levels, and severe weather events caused by elevated GHG concentrations show the negative impact of climate change on natural and human systems [4]. Concerted international efforts over the past 20 years have increased the amount of electricity generated by wind, solar and other renewable sources. However, they have failed to displace fossil fuels from the mix. In 2017, fossil fuels produced more electricity - in relative and absolute terms - than ever before [5]. Advancing and deploying carbon free technologies to solve climate change is imperative.

Nuclear power plants produce no greenhouse gas emissions during operation. Throughout its life cycle, nuclear energy produces about the same amount of carbon dioxide-equivalent emissions per unit of electricity as wind and one-third of the emissions per unit of electricity when compared with solar [5]. Nuclear power provides essential base load power that complements renewables' variability. Many projections of energy portfolios in carbon-limited scenarios include fission power as an important contributor to the rapid decarbonization of the electricity sector in tandem with energy diversification objectives [1, 5]. Large-scale emissions-free nuclear power deployment could significantly reduce GHG production but faces both high upfront cost and perceived adverse safety challenges [4, 1]. The nuclear power industry must overcome cost and safety challenges to ensure

continued global use and expansion of nuclear energy technology to provide low-carbon electricity worldwide.

Wide-spread adoption of additive manufacturing methods in the nuclear industry could drastically reduce reactor fabrication costs and deployment timelines and improve reactor safety [6]. These reductions are achieved by combining multiple systems and assembled components into single parts, tailoring local material properties, and enabling geometry redesign for increased safety and performance [6]. Additive manufacturing (i.e., 3D printing) has progressed rapidly in the last 30 years, from rapid design prototyping with polymers in the automotive industry to metal component-scale production. With further advancements in the material science research surrounding additive manufacturing technologies, a reactor core could be 3D printed within the next decade. Oak Ridge National Laboratory (ORNL) leads this initiative through the 2019 Transformational Challenge Reactor (TCR) Demonstration Program. The TCR program will 3D print a microreactor by leveraging recent scientific achievements in additive manufacturing, nuclear materials, machine learning, and computational modeling and simulation [7].

Additive manufacturing of reactor core components removes the geometric constraints of conventional fuel manufacturing. Reactor designers are no longer limited by conventional geometric shapes that are easy to manufacture with traditional processes, such as slabs as fuel planks, cylinders as fuel rods, spheres as fuel pebbles, and axis-aligned coolant channels [8]. Fabricating reactor core components using additive manufacturing can further optimize core geometries for reactor performance and safety at lower costs [9]. Due to the expansion of the potential design space facilitated through additive manufacturing, reactor designers need to find methods, such as generative design, to explore the design space efficiently. Generative design is an iterative design exploration process [10]; designers input design goals and constraints into a generative design software and the software explores all the possible permutations of a solution, quickly generating design alternatives. The software tests and learns from each iteration what designs works and what does not.

Generative reactor design optimization for arbitrary geometries enabled by additive manufacturing is a new concept, and few research demonstrations have explored the large new design space. Previous efforts include Sobes et al. [8] using genetic algorithms to find minimum volume geometric configurations for the TCR reactor. This study represented arbitrary core geometry variations us-

ing right cylinders. Also, See et al. [11] optimized the TCR’s outlet plenum design by varying inlet channel positioning and outer wall shape. The expanded design space associated with an arbitrary reactor geometry increases the time for reactor designers to find optimal geometries. Instead, we can leverage artificial intelligence (AI) optimization methods, such as evolutionary algorithms, to rapidly explore the large design space for global optimal designs. AI does not replace the human reactor designer but shifts the human designer’s focus away from conjecturing suitable geometries to defining the design criteria of optimal designs and evaluating their relative strengths [8]. Thus, when the human designer changes the reactor criteria, the AI model will quickly adapt and produce new global optimal designs to fit the new criteria.

Thoroughly exploring the design space enabled by additive manufacturing should allow the placement of fuel, moderation, and coolant material in any possible location within physical limits. In this dissertation, I explore the large design space while acknowledging that this work is only an intermediate step towards developing a truly arbitrary geometry expression. I apply evolutionary algorithms to conduct generative reactor design optimization. For this dissertation, I designed the Reactor evOLutionary aLgorithm Optimizer (ROLLO) tool [12] to drive the evolutionary algorithm optimization process and used OpenMC [13] and Moltres [14] software to model the reactor’s neutronics and temperature distribution. ROLLO generates reactor designs with varying fuel amounts, fuel distributions, and coolant channel shapes to minimize three key reactor performance metrics: total fuel amount, maximum temperature, and fuel-normalized power peaking factor.

In 2000, the Department of Energy (DOE) initiated the Generation IV International Forum, which aims to enhance the role of nuclear energy in our global energy ecosystem by leading and planning research and development to support new and innovative Generation IV nuclear energy systems [15]. Generation IV nuclear systems target goals in four areas: sustainability, economics, safety and reliability, and proliferation resistance and physical protection [15]. They identified six Generation IV systems that met these goals: Gas-Cooled Fast Reactors (GFRs), Lead-Cooled Fast Reactors (LFRs), Molten Salt Reactors (MSRs), Sodium-Cooled Fast Reactors (SFRs), Supercritical-Water-Cooled Reactors (SCWRs), and Very-High-Temperature Reactors (VHTRs) [15]. In this dissertation, I optimize the Fluoride-Salt-Cooled High-Temperature Reactor (FHR), which combines the best aspects of MSR and VHTR Generation IV systems. FHRs use high-temperature coated-

particle fuel (similar to the VHTRs) and a low-pressure liquid fluoride-salt coolant (similar to the MSRs) [16, 17]. This dissertation focuses on a prismatic FHR design with hexagonal fuel assemblies consisting of Tristructural Isotropic (TRISO) fuel particles embedded in planks, i.e., the Advanced High-Temperature Reactor (AHTR) design. The *triple heterogeneous* AHTR fuel, comprises of TRISO particles embedded in strategically arranged plates, presents complex reactor physics and significant modeling challenges [18]. To further understand and address the technical challenges associated with the AHTR design, I participate in the Organisation for Economic Co-operation and Development (OECD)-Nuclear Energy Agency (NEA)'s FHR benchmarking exercise [18].

1.1 Objectives and Outline

This dissertation's content is shaped by the following three objectives:

1. Furthering our understanding of the AHTR design's complexities through neutronics and temperature modeling
2. Creating an open-source tool that enables nuclear reactor design evolutionary algorithm optimization for non-conventional reactor geometries and fuel distributions
3. Applying the optimization tool to the AHTR design

Chapter 3 addresses objective 1, chapter 4 addresses objective 2, and chapters 5, 6, and 7 address objective 3.

Chapter 2 presents a literature review that organizes and reports on previous work pertinent to all three objectives. I provide an overview of the FHR concept, then detail one specific FHR design: the AHTR. I describe previous efforts and technical challenges of modeling the AHTR design and how these efforts led to the OECD NEA's FHR benchmark initiation. Next, I outline additive manufacturing's history and describe the current research on using additive manufacturing for nuclear reactor component fabrication. I review previous nuclear reactor design optimization efforts and describe how additive manufacturing of nuclear reactor components enables optimization for less constrained reactor geometries. I describe optimization methods that can be leveraged to find optimal reactor designs in the expanded design space. Finally, I provide a background of the

evolutionary algorithms and detail a specific evolutionary algorithm: the genetic algorithm and how it works to conduct global optimization robustly.

Chapter 3 describes the FHR benchmark specifications and the University of Illinois at Urbana-Champaign (UIUC) team’s results. The OECD-NEA and Georgia Institute of Technology (Georgia Tech) initiated the FHR benchmark for the AHTR design in 2019 [18] to address the AHTR modeling challenges such as multiple heterogeneity and material cross-section data. UIUC participates in the FHR benchmark with the OpenMC Monte Carlo code [19] and the ENDF/B-VII.1 material cross section library [20]. The UIUC team consists of myself and my advisors, Professor Kathryn Huff, and Dr. Madicken Munk. The 2D AHTR full assembly presented results are from the benchmark’s Phases I-A and I-B, which model steady-state and depletion, and the temperature model results. I will use this analysis to underscore the complex modeling and simulation needed to adequately model this design.

Chapter 4 describes the Reactor evOLutionary aLgorithm Optimizer (ROLLO) tool designed for this dissertation. ROLLO is a Python package that applies evolutionary algorithm techniques to optimize nuclear reactor design. The motivation for ROLLO is to enable reactor designers to utilize robust evolutionary algorithm optimization methods without going through the cumbersome process of setting up a genetic algorithm framework, selecting appropriate hyperparameters, and setting up its parallelization. ROLLO provides a general genetic algorithm framework, sets up parallelization for the user, and promotes usability with an input file that only exposes mandatory parameters. I designed ROLLO to be effective, flexible, accessible, parallel, reproducible, and usable. ROLLO is nuclear code-agnostic with no dependencies on nuclear software and is hosted on Github [12].

Chapter 5 describes the modeling and optimization methodology of the AHTR plank and one-third assembly optimization for non-conventional geometries and parameters conducted using the ROLLO software. Because this is a proof of concept, I limit my approach to several key reactor parameters. I define and describe the AHTR input parameters varied: TRISO packing distribution, total fuel packing fraction, and coolant channel shape. I define and motivate the key optimization objectives: minimizing fuel amount, maximizing heat transfer, and minimizing power peaking. I describe the AHTR geometries, outline the AHTR modeling workflow, verify the AHTR models,

and describe the process of tuning ROLLO’s hyperparameters for the optimization problems.

Chapters 6 and 7 report the AHTR plank and one-third assembly ROLLO optimization results, respectively. I vary the following AHTR plank and one-third assembly input parameters: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), total fuel packing fraction (PF_{total}), and coolant channel shape; to minimize the following objectives: total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}). For each AHTR geometry, I conduct six single objective, single input parameter ROLLO optimizations to understand the individual impacts of each objective on each input parameter and inform the multi-objective optimization simulation setup. I conduct three two-objective ROLLO optimizations and two three-objective ROLLO optimizations for different numbers of input parameters based on the results from the single objective studies. I then investigate the driving factors for each objective and how their combined effects result in the optimal reactor models found by the multi-objective optimization simulations.

Finally, Chapter 8 provides general conclusions and discussion about future work.

Chapter 2

Literature Review

This chapter provides a literature review of relevant past research efforts giving context to this dissertation. I begin with an overview of the FHR concept, then detail one specific FHR design: the AHTR. I describe previous efforts and technical challenges of modeling the AHTR design, and how these efforts led to the OECD NEA's FHR benchmark initiation. Next, I outline additive manufacturing's history and describe the current research on using additive manufacturing for nuclear reactor component fabrication. I review previous nuclear reactor design optimization efforts and describe how additive manufacturing of nuclear reactor components enables optimization for less constrained reactor geometries. I describe optimization methods that can be leveraged to find optimal reactor designs in the expanded design space. Finally, I give a background of the evolutionary algorithms and detail a specific evolutionary algorithm: the genetic algorithm and how it works to conduct global optimization robustly.

2.1 Fluoride-Salt-Cooled High-Temperature Reactor System

To ensure continued global use and expansion of nuclear energy technology, in 2001, the OECD and nuclear energy experts initiated the Generation IV International Forum (GIF) [15]. The GIF aims to enhance the role of nuclear energy in our global energy ecosystem by coordinating global research and development to test the feasibility and performance of Generation IV nuclear reactor systems to enable their industrial deployment by 2030 [15]. The GIF selected six Generation IV systems for further research and development based on target goals in four areas: sustainability, economics, safety and reliability, and proliferation resistance and physical protection [15]. Table 2.1 summarizes the goals in each area. The systems selected are GFRs, LFRs, MSRs, SFRs, SCWRs, and VHTRs [15]. The Fluoride-Salt-Cooled High-Temperature Reactor (FHR) concept introduced

Table 2.1: Goals of Generation IV Nuclear Systems [15, 15]

Area	Goals
Sustainability	<ul style="list-style-type: none"> - Have a positive impact on the environment through the displacement of polluting energy and transportation sources by nuclear electricity generation and nuclear-produced hydrogen - Promote long-term availability of nuclear fuel - Minimize volume, lifetime, and toxicity of nuclear waste
Economics	<ul style="list-style-type: none"> - Have a life cycle and energy production cost advantage over other energy sources - Reduce economic risk to nuclear projects by developing plants using innovative fabrication and construction techniques
Safety and Reliability	<ul style="list-style-type: none"> - Increase the use of robust designs and inherent and transparent safety features that non-experts can understand - Enhance public confidence in the safety of nuclear energy
Proliferation Resistance and Physical Protection	<ul style="list-style-type: none"> - Provide continued effective proliferation resistance of nuclear energy systems through improved design features and other measures - Increase the robustness of new facilities

in 2003 uses a low-pressure liquid fluoride-salt coolant and high-temperature coated-particle TRISO fuel, combining the best aspects of the MSR and VHTR systems respectively [21, 17].

MSR systems produce fission power in a circulating molten salt fuel mixture. Molten salt reactor coolants introduce inherent safety due to the low system vapor pressure and the salts' high boiling temperature and volumetric heat capacity [22]. Fluoride salt used in FHRs is Li_2BeF_4 , also known as Fluoride-Lithium-Beryllium (FLiBe), which remains liquid without pressurization up to 1400 °C and has a greater heat capacity than water [22, 16]. Researchers recommend molten fluoride salts because they have high uranium solubility, chemical stability, low vapor pressure at high temperatures, good heat transfer properties, radiation damage resistance, and are inert to common structural materials [23]. VHTR systems use a once-through uranium cycle and leverage high outlet temperatures for high-temperature heat applications, such as hydrogen production. Graphite-moderated and helium-cooled VHTRs use TRISO fuel which withstands high burnup and temperature, enabling higher operating temperatures [15]. Higher operating temperatures advantages include increased power conversion efficiency, reduced waste heat generation, and co-generation and process heat capabilities [24]. However, the VHTRs system's helium coolant is at 100 atm requiring a thick concrete vessel.

By combining FLiBe coolant from MSR technology and TRISO fuel from VHTR technology,

the FHR benefits from low operating pressure and large thermal margin enabled by the molten fluoride salt coolant and the thermal resilience of TRISO particle fuel. The low vapor pressure in molten fluoride salts ensures reduced stress on the system and increased safety compared to the VHTR's high operating pressure. TRISO solid fuel cladding in the FHR system adds an extra barrier to fission product release compared to MSRs with liquid fuel [22].

Several types of FHR conceptual designs exist worldwide: the Pebble-Bed Fluoride-Salt-Cooled High-Temperature Reactor (PB-FHR) developed at University of California Berkeley (UCB) with circulating pebble-fuel [25, 26], the Solid Fuel Thorium Molten Salt Reactor (SF-TMSR) developed at the Shanghai Institute of Applied Physics (SINAP) with static pebble-fuel [27], the large central-station AHTR at ORNL [28, 29], and the Small Modular AHTR (SmAHTR) at ORNL [30] with static, plate fuel.

2.1.1 Advanced High-Temperature Reactor Design

This dissertation focuses on the prismatic FHR design with hexagonal fuel assemblies consisting of TRISO fuel particles embedded in planks, i.e., the AHTR design developed by ORNL. The AHTR has 3400 MWt thermal power and 1400 MW electric power with inlet/outlet temperatures of 650/700°C [29]. Figure 2.1 shows the prismatic AHTR's fuel assembly and core configuration. Each hexagonal fuel assembly features plate-type fuel consisting of eighteen planks arranged in

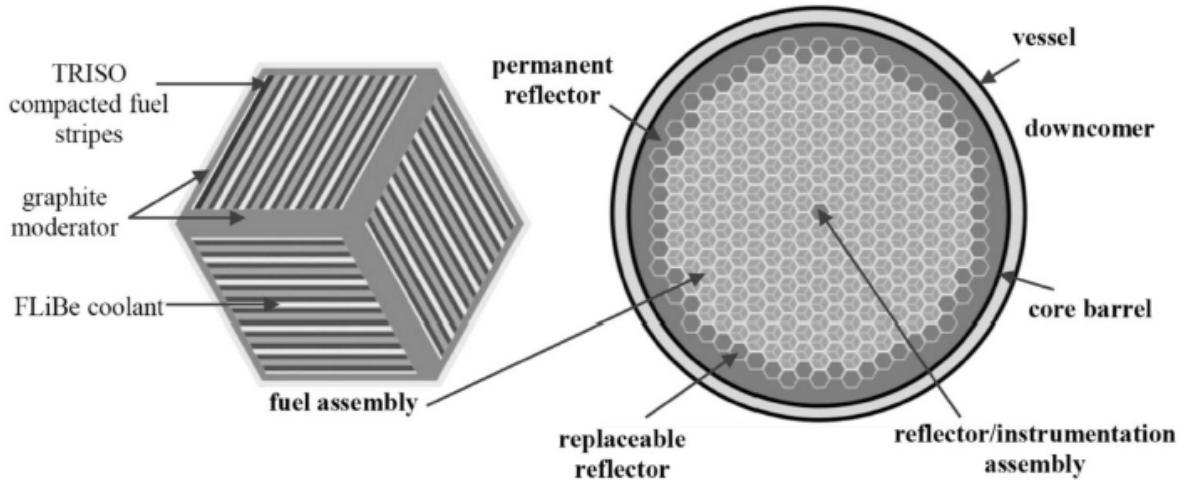


Figure 2.1: Advanced High-Temperature Reactor fuel assembly (left) and core configuration (right) reproduced from [31].

three diamond-shaped sectors, with a central Y-shaped structure and external channel (wrapper). The fuel planks contain an isostatically pressed carbon with fuel stripes on each plank's outer side. Within each fuel stripe is a graphite matrix filled with TRISO particles. The core consists of 252 assemblies radially surrounded by reflectors [31]. Chapter 3 details the specifications of the AHTR geometry modeled in this dissertation.

2.1.2 Previous AHTR Modeling Efforts and Challenges

Most state-of-the-art reactor simulation tools today have been designed for Light Water Reactor (LWR) analysis. The AHTR core design differs significantly from the currently operating LWR cores. These differences lead to modeling challenges with the current tools, highlighting the need to verify and validate current simulation tools for AHTR physics [31]. Verification and validation of AHTR neutronics and thermal-hydraulics simulation capabilities support the AHTR design's licensure and the eventual goal of AHTR deployment [32, 33]. This section outlines the previous efforts to model and validate the AHTR's neutronics and thermal-hydraulics.

AHTR Neutronics Modeling

Several neutronics studies conducted to support the current iteration of the AHTR design have illuminated the design's technical challenges [31, 34, 30]. Georgia Tech led an Integrated Research Project to understand AHTR material and modeling challenges [35]. During the research project, a panel of subject matter experts generated a Phenomena Identification and Ranking Table (PIRT). The PIRT identifies AHTR areas that need additional research to better understand important phenomena for adequate future modeling [32]. Table 2.2 lists the phenomena identified as requiring further research.

The *triple heterogeneous* AHTR fuel, comprised of TRISO particles embedded in strategically arranged plates, presents simulation challenges. Researchers must obtain detailed reference power distributions with individual TRISO particle fidelity to understand nuances in the physics, such as self-shielding. Deterministic codes that use multigroup cross sections and traditional homogenization methods [31], insufficiently capture the AHTRs physics due to these multiple heterogeneities [31]. In the AHTR, single and multiple slab homogenization decreased total neutron transport

Table 2.2: Phenomena Identification and Ranking Table identified Advanced High-Temperature Reactor physical phenomena requiring further research [32].

Category	Phenomena
Fundamental cross section data	<ul style="list-style-type: none"> - Moderation in FLiBe - Thermalization in FLiBe - Absorption in FLiBe - Thermalization in carbon - Absorption in carbon
Material Composition	<ul style="list-style-type: none"> - Fuel particle distribution
Computational Methodology	<ul style="list-style-type: none"> - Solution Convergence - Granularity of depletion regions - Multiple heterogeneity treatment for generating multigroup cross sections - Selection of multigroup structure - Boundary conditions for multigroup cross section generation
General Depletion	<ul style="list-style-type: none"> - Spectral history

simulation time by an order of 10; however, the homogenization introduced a nontrivial k_{eff} error of $\sim 3\%$ [31, 36]. To determine the feasibility and safety of the AHTR design, researchers must calculate core physics parameters to an acceptable uncertainty. With Monte Carlo neutron transport, increasing neutron histories reduces statistical uncertainty but increases computational cost typically, requiring supercomputers to run the simulations.

This AHTR presents another technical challenge: the uncertainty of graphite moderator material properties: densities, temperatures, and thermal scattering data. Also, the thermal scattering data ($S(\alpha, \beta)$ matrices) for the bound nuclei in FLiBe salts are lacking [31]. Mei et al. [37] and Zhu et al. [38] examined the thermal scattering behavior of solid and liquid FLiBe. They concluded that the bound and free atom cross section of FLiBe are identical above 0.1eV and diverge below 0.01eV, which means that the use or absence of thermal scattering data will impact the accuracy of the results [31].

AHTR Multiphysics Modeling

In a past effort toward AHTR multiphysics modeling, Gentry et al. [39] developed an adapted lattice physics-to-core simulator two-step procedure with Serpent [40] and Nodal Eigenvalue, Steady-state, Transient, Le core Evaluator (NESTLE) [41] for the AHTR design. The adapted lattice physics-to-core simulator two-step procedure proved to be successful for LWRs. A 2-D transport

lattice calculation generates the LWR’s group assembly homogenized group constants, and then core analysis is performed by 3-D nodal simulation [42, 39]. NESTLE’s thermal-hydraulics utilizes a Homogenous Equilibrium Mixture (HEM) model for two-phase flow, and it solves the few-group neutron diffusion equation utilizing the Nodal Expansion Method (NEM) for cartesian and hexagonal reactor geometries. Gentry et al. concluded that the method required accuracy improvements by improving the reflector model and further optimizing the coarse energy group structure. Lin [43] used RELAP5, a system-level code, to perform AHTR thermal hydraulics transient simulations to investigate the capability of the passive heat removal system. In the RELAP5 model, Lin separated the AHTR’s 252 assemblies into four concentric rings with uniform power distribution per ring. Lin also gives higher fidelity in the primary and Direct Reactor Auxiliary Cooling System (DRACS) system loops. Lin utilized RELAP5 in transient scenarios to determine the temperature at various locations in the system loop by assuming a power value for fuel assemblies in each ring [43]. However, this method is not ideal for transient scenarios with tightly coupled neutronics and thermal-hydraulics.

2.1.3 FHR Benchmark

The previous section highlights the singular efforts to model different aspects of the AHTR’s neutronics and thermal hydraulics, with each author describing their modeling difficulties. However, there lacked a robust and methodical method for evaluating the simulation software and comparing the AHTR modeling results generated by individual researchers.

To gain a comprehensive view of the AHTR’s modeling challenges and cross-verify available AHTR modeling tools, in 2019, the OECD-NEA initiated the FHR benchmarking exercise of the AHTR design [18]. Several organizations participate in the benchmark with various Monte Carlo and Deterministic neutronics software, such as Serpent [40], OpenMC [19], and WIMS [44].

The benchmark has three phases: a single fuel assembly simulation without burnup (Phase I), full core depletion (Phase II), and multiphysics feedback (Phase III). The benchmark aims to identify the latest codes’ applicability, accuracy, and practicality to assess the current state-of-the-art FHR simulation and modeling [45]. The benchmark also enables the cross-verification of software and methods for the challenging AHTR geometry, which is especially useful since

applicable reactor physics experiments for code validation are scarce [46, 45]. Chapter 3 will provide a detailed description of the benchmark’s phases and results.

2.1.4 Modeling Software

In this dissertation, I use OpenMC [13] to model AHTR’s neutronics and Moltres [14] to model the AHTR’s temperature. In the following subsections, I will describe the capabilities of each software and motivate their use for my dissertation.

OpenMC

OpenMC [13] is a continuous-energy Monte Carlo neutron transport open-source code hosted on Github. It was created in 2011 at Massachusetts Institute of Technology (MIT) as part of an effort to develop scalable parallel Monte Carlo neutron transport algorithms for future exascale supercomputers [13], and is currently supported by resources from Argonne National Laboratory (ANL). OpenMC runs in parallel using Message Passing Interface (MPI) and OpenMP. Many components of OpenMC have been validated and verified with experimental data and other well-established neutron transport codes through benchmarks [13].

OpenMC is capable of performing fixed source and k-eigenvalue calculations. This dissertation uses OpenMC’s k-eigenvalue calculations for AHTR neutronics modeling. OpenMC represents neutron interaction with various nuclei in the ACE format [47]. The NJOY nuclear data processing system [48] converts the raw ENDF/B data into linearly-interpolable ACE-format data, enabling the nuclear data libraries to provide continuous energy cross section data at discrete temperatures. OpenMC models use constructive solid geometry representation; they provide standard geometric surfaces (e.g. planes, cylinders) for defining reactor geometries. This dissertation uses v0.12.2 of OpenMC with the ENDF VII.1 nuclear data library [20].

Moltres

Moltres is an open-source tool designed to simulate MSRs using deterministic neutronics and thermal-hydraulics implemented as an application atop the Multiphysics Object-Oriented Simulation Environment (MOOSE) finite-element framework. Moltres solves arbitrary group neutron

diffusion, temperature, and precursor governing equations on a single mesh and can be deployed on an arbitrary number of processing units [14, 49]. To perform neutronics calculations, Moltres requires group constant data from high-fidelity neutronics software such as OpenMC [13] or Serpent 2 [40]. Users run a Python script in Moltres’ Github repository, which automatically reads user-provided OpenMC/Serpent 2 output files and creates Moltres-compatible JSON containing all required group constant data.

Moltres solves for the neutron fluxes governed by the multigroup neutron diffusion equations:

$$\frac{1}{v_g} \frac{\partial \phi_g}{\partial t} = \nabla \cdot D_g \nabla \phi_g - \Sigma_g^r \phi_g + \sum_{g' \neq g}^G \Sigma_{g' \rightarrow g}^s \phi_{g'} + \chi_g^p \sum_{g'=1}^G (1 - \beta) \nu \Sigma_{g'}^f \phi_{g'} + \chi_g^d \sum_i^I \lambda_i C_i \quad (2.1)$$

where

v_g = average speed of neutrons in group g [$cm \cdot s^{-1}$],

ϕ_g = neutron flux in group g [$cm^{-2} \cdot s^{-1}$],

t = time [s],

D_g = diffusion coefficient of neutrons in group g [$cm^{-2} \cdot s^{-1}$],

Σ_g^r = macroscopic cross section for removal of neutrons from group g [cm^{-1}],

$\Sigma_{g' \rightarrow g}^s$ = macroscopic cross section of scattering from groups g' to g [cm^{-1}],

χ_g^p = prompt fission spectrum for neutrons in group g [-],

G = total number of discrete neutron groups [-],

ν = average number of neutrons produced per fission [-],

Σ_g^f = macroscopic fission cross section for neutron in group g [cm^{-1}],

χ_g^d = delayed fission spectrum for neutrons in group g [-],

I = total number of delayed neutron precursor groups [-],

β = total delayed neutron fraction [-].

Moltres’ governing temperature equation is an advection-diffusion equation with a fission heat

source term Q_f :

$$\rho c_p \frac{\partial T}{\partial t} = -\rho c_p \vec{u} \cdot \nabla T + \nabla \cdot (k \nabla T) + Q_f \quad (2.2)$$

and

$$Q_f = \sum_{g=1}^G \epsilon_g \Sigma_g^f \phi_g \quad (2.3)$$

where

ρ = density of the molten salt [$g \cdot cm^{-3}$],

c_p = specific heat capacity of molten salt [$J \cdot g^{-1} \cdot K^{-1}$],

T = temperature of molten salt [K],

k = thermal conductivity of molten salt [$W \cdot cm^{-1} \cdot K^{-1}$],

Q_f = fission heat source [$W \cdot cm^{-3}$],

ϵ_g = average fission energy released by neutrons in group g [J].

Σ_g^f = macroscopic fission cross section for neutron in group g [cm^{-1}],

ϕ_g = neutron flux in group g [$cm^{-2} \cdot s^{-1}$],

2.2 Additive Manufacturing

Additive manufacturing is the formal term for what is popularly known as ‘3D printing’ [50]. The basic principle of additive manufacturing is that a model is initially generated using a three-dimensional Computer-Aided Design (3D CAD) system and then fabricated directly without process planning. As the name implies, additive manufacturing adds material in layers. Each layer is a thin cross section of a 3D CAD-designed part, as opposed to traditional machining, which subtracts material instead [51]. All commercialized additive manufacturing machines to date use a layer-based approach. The methods differ in the following ways: materials, layer creation method, and how the layers are bonded [50]. These major differences will determine the: accuracy of

the final part, material and mechanical properties, time required to manufacture the part, need for post-processing, size of additive manufacturing machine, and overall cost of the machine and the process [50]. Initially, industries only utilized additive manufacturing for manufacturing prototypes. However, with improvements in material properties, accuracy, and overall quality of additive manufacturing output, the applications for additive manufacturing have advanced. Industries have begun 3D printing parts for direct assembly purposes, such as air-cooling ducts for aircraft, hearing aids, and prosthesis [52].

Additive manufacturing has progressed rapidly in the last 30 years, from rapid design prototyping with polymers in the automotive industry to scale production of metal components. Examples include Boeing using additive manufacturing to reduce the 979 Dreamliner's weight [53] and General Electric using additive manufacturing to produce fuel injection nozzles [54]. The most common metal additive manufacturing technologies, selective laser melting (SLM), electron beam melting (EBM), laser directed energy deposition (L-DED), and binder jetting, are not currently used to manufacture nuclear power plant parts.

The U.S. DOE, National Laboratories, and Electric Power Research Institute (EPRI) support research and development efforts toward deploying, testing, and qualifying additive manufacturing methods for nuclear components. However, the nuclear industry's efforts to incorporate additive manufacturing into the supply chain lag behind the auto and aerospace industries due to the lack of clarity on regulatory pathways. The aerospace and automotive industries benefit from long-standing and resourced regulatory and standards development activities [55]. Thus, in 2019 the Nuclear Regulatory Commission (NRC) addressed these regulatory challenges by issuing a draft action plan to prepare the agency to review applications for additive manufacturing of nuclear components and clarify the industry's expectations of their use [55].

2.2.1 Benefits of 3D Printing Reactor Components

Wide-spread adoption of additive manufacturing methods in the nuclear industry could drastically reduce fabrication costs and timelines. These reductions are achieved by combining multiple systems and assembled components into single parts, tailoring local material properties, and enabling geometry redesign for increased safety and performance [6]. Many Generation IV advanced reactor

concepts have complex geometries, such as hexagonal-ducts for sodium-cooled fast reactors, that are costly and difficult to fabricate using standard processing techniques [56]. These complex designs can leverage 3D printed parts to enormous benefit. Additive manufacturing advancements for reactor core components remove conventional fuel manufacturing geometric constraints. Reactor designers can now approach the nuclear design problems with truly arbitrary geometries, no longer limited by conventional geometric shapes that are easy to manufacture with traditional processes: slabs as fuel planks, cylinders as fuel rods, spheres as fuel pebbles, axis-aligned coolant channels, etc [8]. In summary, reactor core component fabrication with additive manufacturing enables further fuel geometry optimization and improvement to enhance reactor performance at lower costs [9].

2.2.2 Efforts toward 3D Printing Reactor Components

In 2019, ORNL initiated the TCR Demonstration Program. The TCR program leverages recent scientific achievements in additive manufacturing, nuclear materials, machine learning, and computational modeling to reduce deployment costs and timelines for advanced nuclear energy systems. The TCR program aims to utilize additive manufacturing technology to establish advanced nuclear energy system designs unconstrained by conventional manufacturing and additively manufacture a microreactor [7].

The TCR program followed a downselection process based on the program's design constraints to select the reactor's design, materials, and components [57]. The downselected TCR design is a TRISO-fueled and yttrium hydride moderated gas-cooled reactor [57]. At ORNL, Trammel et al. [58] demonstrated the fabrication of a SiC fuel element with embedded TRISO fuel using additive manufacturing techniques: binder jet printing and chemical vapor infiltration (CVI). They followed the following fabrication steps (depicted in Figure 2.2):

1. Binder jet technology prints a SiC fuel element with coolant channel structures.
2. The designated fueled region of the element is loaded with surrogate TRISO particles and additional SiC powder to fill interstitial spaces between particles.
3. The loaded fuel element is densified in a CVI process to achieve microencapsulation of TRISO particles in a SiC matrix.



Figure 2.2: Stages of additive manufacturing fabrication conducted at Oak Ridge National Laboratory to produce a fuel demonstration element with TRISO particles embedded in a SiC matrix [58]. Figure reproduced from [58].

Figure 2.3 shows the fuel element manufactured at ORNL for the TCR program [57]. The current design uses axially uniform fuel shapes and coolant channels. The TCR program plans to optimize the coolant channels to vary in axial and radial directions within the fuel element [57, 8].

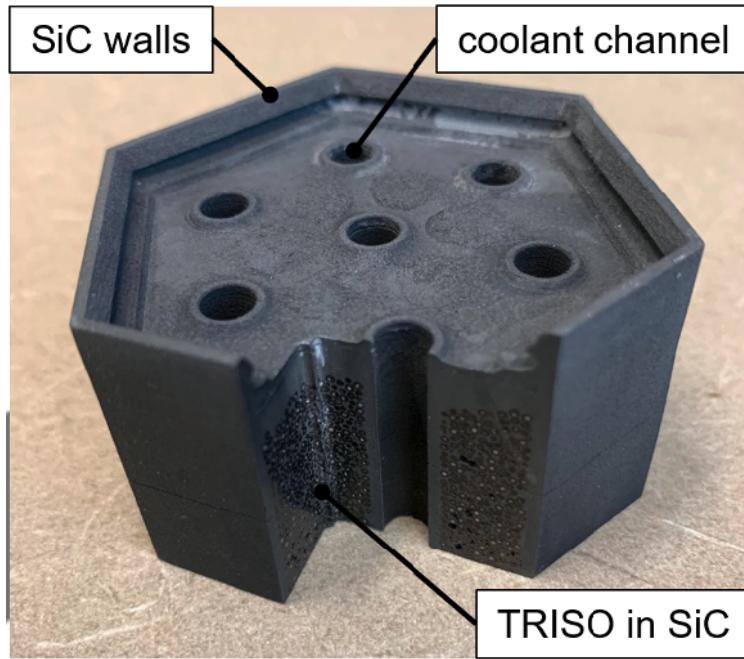


Figure 2.3: The top portion of the advanced manufactured fuel element for the Transformational Challenge Reactor (TCR) program at Oak Ridge National Laboratory (ORNL). A cutaway of the element shows surrogate TRISO particles packed in the SiC matrix. Figure reproduced from [57].

Besides the TCR program, the nuclear materials research community has made significant progress in demonstrating the application of additive manufacturing to nuclear fuel and structural core material fabrication. Rosales et al. [59] conducted a feasibility study of direct routes to

fabricate dense uranium silicide (U_3Si_2) fuel pellets using the Idaho National Laboratory (INL) approach known as Additive Manufacturing as an Alternative Fabrication Technique (AMAFT). U_3Si_2 demonstrates desirable accident-tolerant nuclear fuel properties such as high uranium density and improved thermal properties; however, it has an expensive and long metallurgical fabrication process. Thus, using AMAFT to fabricate U_3Si_2 will lower costs and ensure a timely and commercially-reliable fabrication process [59]. Sridharan et al. [56] demonstrated applying the laser-blown-powder additive manufacturing process to fabricate ferritic/martensitic (FM) steel, commonly used for cladding and structural components in nuclear reactors. Koyanagi et al. [60] presented the latest additive manufacturing technology for manufacturing nuclear-grade silicon carbide (SiC) materials. They demonstrated that combining additive manufacturing techniques and traditional SiC densification methods enabled new designs of SiC components with complex shapes. SiC demonstrates excellent strength at elevated temperatures, chemical inertness, relatively low neutron absorption, and stability under neutron irradiation up to high doses [61, 62, 60]. These qualities make SiC suitable for many applications in nuclear systems, such as fuel cladding, constituents of fuel particles [62] and pellets [63], and core structural components in fission reactors [61].

2.3 Nuclear Reactor Design Optimization

A nuclear reactor's complexity results in reactor design optimization being a multi-objective design problem requiring a tradeoff between desirable attributes [64, 65]. When multiple conflicting objectives compete, no single optimum solution simultaneously optimizes all objectives. Instead, multi-objective optimization returns multiple optimal solutions that meet each objective to varying degrees; this set of solutions is the Pareto front [66]. Figure 2.4 illustrates a two-dimensional problem's objective space and Pareto Front. For each solution in the Pareto front, none of the objective functions can be improved without degrading another objective. An ideal optimization method for a multi-objective problem like reactor design should find widely spread solutions in the obtained Pareto front [66].

Traditional manufacturing constraints result in most of the past nuclear reactor optimization work focusing on optimizing classical reactor parameters such as radius of fuel pellet and clad,

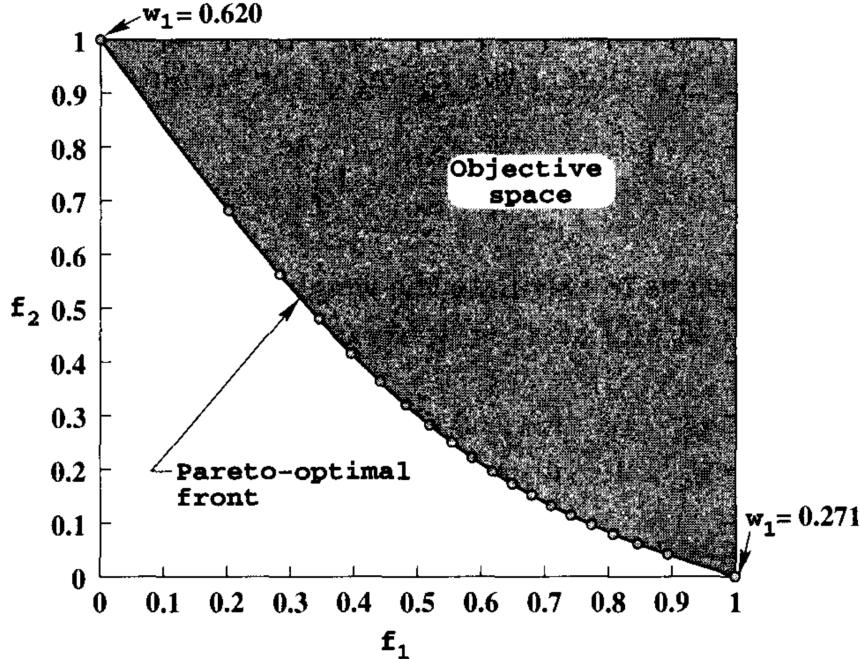


Figure 2.4: Illustration of an objective space and Pareto Front. Figure reproduced from [66]. The f_1 and f_2 axes correspond to the optimization objectives.

enrichment of fuel, pin pitch, etc. The optimization methods used for reactor design optimization are either deterministic or stochastic. Deterministic optimization methods usually start from a guess solution. Then, the algorithm suggests a search direction by applying local information to a pre-specified transition rule. Any better solution becomes the new solution, and the above procedure continues several times [66]. Drawbacks of deterministic methods include: algorithms tend to get stuck at suboptimal solutions, and an algorithm efficient in solving one type of problem may not solve a different problem efficiently [66]. Stochastic optimization methods such as evolutionary algorithms and simulated annealing minimize or maximize an objective function with randomness present. Stochasticity enables them to find globally optimal solutions more reliably than deterministic methods. Due to stochastic methods' many advantages, most efforts toward nuclear reactor optimization use these methods.

Additive manufacturing advancements for reactor core components have removed conventional reactor manufacturing geometric constraints such as slabs as fuel planks, cylinders as fuel rods, spheres as fuel pebbles, axis-aligned coolant channels, etc [8]. Reactor design objectives remain consistent with past objectives, such as minimizing fuel amount and minimizing the maximum fuel

temperature for a given power level. However, reactor designers can now approach nuclear design problems with truly arbitrary geometries and optimize beyond classical parameters to further enhance fuel performance and safety. This has opened the door for a complete re-examination of reactor core optimization, determining the optimal arbitrary geometry and fuel distribution for a given objective function with a much smaller set of constraints [8].

I discuss the previous nuclear reactor optimization efforts for classical and arbitrary parameters in the subsequent sections.

2.3.1 Reactor Optimization for Classical Parameters

This section describes previous efforts towards reactor optimization for classical geometry parameters, such as radius of fuel pellet and clad, enrichment of fuel, pin pitch, etc. The most commonly used stochastic optimization methods for reactor design optimization are simulated annealing and evolutionary algorithms.

Reactor Optimization with Simulated Annealing Method

Simulated annealing iteratively updates one candidate solution until it reaches the termination criteria. At each iteration, the simulated annealing algorithm selects a random move. If the selected move improves the solution, it is always accepted; however, if it does not improve the solution, the algorithm updates the solution with some probability of less than 1.

Sacco et al. [67, 68] used stochastic simulated annealing and deterministic-stochastic hybrid optimization techniques to optimize reactor dimensions, enrichment, materials, etc., to minimize the average peaking factor in a three-enrichment-zone reactor. Odeh et al. [69] used the simulated annealing stochastic algorithm coupled with neutronics and thermal-hydraulics simulation tools, Purdue Advanced Reactor Core Simulator (PARCS) and RELAP5 [70], to develop an optimal Purdue Novel Modular Reactor (NMR-50) core design with a 10-year cycle length and minimal fissile loading. Kropaczek et al. [71] demonstrated the constraint annealing method: a highly scalable method based on the parallel simulated annealing method with mixing of states [72] to solve large-scale, multiconstrained LWR fuel cycle optimization problems. These papers demonstrate the simulated annealing optimization method's success in reactor design optimization problems.

Nuclear reactor optimization problems require computationally expensive neutronics and thermal-hydraulics software to compute the objective function and constraints. Multiple papers utilized stochastic optimization methods with surrogate models to reduce computational cost. The surrogate models reduce the computational cost by replacing high-fidelity neutronics or thermal hydraulics simulations. Betzler et al. [73] developed a systematic approach to build a surrogate model to serve in place of high-fidelity computational analyses. They leveraged the surrogate model with a simulated annealing optimization algorithm to generate optimized designs at a lower computational cost and understand the impact of design decisions on desired metrics for High Flux Isotope Reactor (HFIR) low-enriched uranium (LEU) core designs.

The simulated annealing method uses a point-by-point approach: one solution gets updated to a new solution in one iteration, which does not exploit parallel systems' advantages. Finding an optimal solution with simulated annealing methods takes very long if high-fidelity computationally expensive codes compute the objective function and constraints. Using a simulated annealing method is only practical with surrogate evaluation models, as described in Betzler et al. [73].

Reactor Optimization with Evolutionary Algorithm Method

Contrary to a single solution per iteration in deterministic and stochastic simulated annealing methods, evolutionary algorithms use a population of solutions in each iteration [66]. Evolutionary algorithm methods mimic nature's evolutionary principles by driving the search toward an optimal solution through population elimination and combinations.

Peireira et al. [74, 75] used a coarse-grained parallel genetic algorithm and a niching genetic algorithm to minimize the average peaking factor in a three-enrichment-zone reactor. Kamalpour et al. [76] utilized the imperialist competitive algorithm, an evolutionary algorithm, to optimize an fully ceramic microencapsulated (FCM) fuelled Pressurized Water Reactor (PWR) to extend the reactor core cycle length. Kumar et al. [77] combined genetic algorithm optimization with a surrogate model to optimize for high breeding of ^{233}U and ^{239}Pu in desired power peaking limits and k_{eff} by varying: fuel pin radius, fissile material isotopic enrichment, coolant mass flow rate, and core inlet coolant temperature.

With the affordability and availability of parallel computing systems, the evolutionary algo-

rithm optimization method stands out as a method that easily and conveniently exploits parallel systems. Further, evolutionary algorithms have proved amenable to high-performance computing (HPC) solutions and scalable to tens of thousands of processors [72]. Thus, for optimization problems that require high-fidelity evaluation software, the evolutionary algorithm method can leverage parallel computing to find a solution faster than the simulated annealing method. Therefore, I will utilize the evolutionary algorithm optimization method in this dissertation. Section 2.4 provides an overview of the literature on evolutionary algorithms.

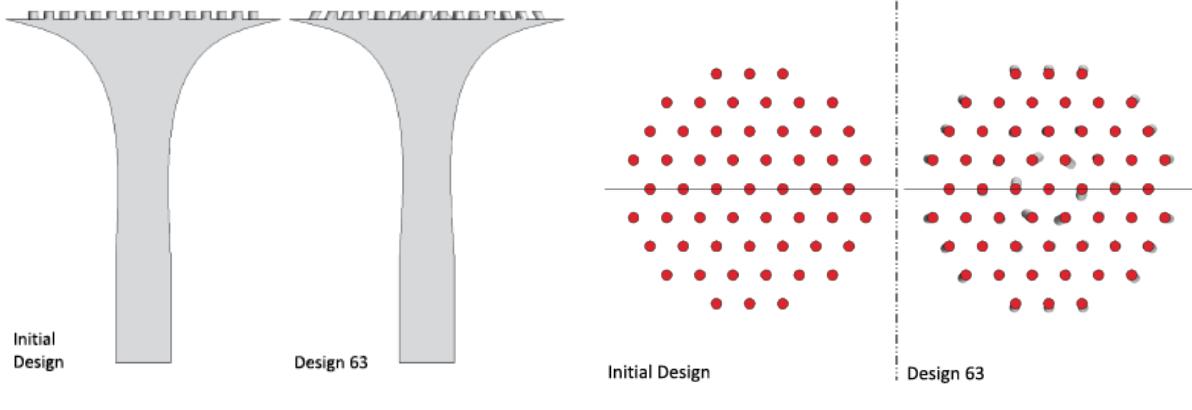
2.3.2 Reactor Optimization for Arbitrary Parameters

Additive manufacturing advancements for reactor core components have removed conventional reactor manufacturing geometric constraints such as slabs as fuel planks, cylinders as fuel rods, spheres as fuel pebbles, axis-aligned coolant channels, etc [8]. Reactor designers can now approach nuclear design problems with truly arbitrary geometries and optimize beyond classical parameters to further enhance fuel performance and safety. This section describes previous efforts towards reactor optimization for arbitrary geometry parameters.

Sobes et al. [8] used a genetic algorithm to find minimum volume geometric configurations for a TCR-like reactor with multiphysics constraints of 1500 pcm excess reactivity and maximum fuel temperature of 618°C under forced-flow cooling conditions. This study represented arbitrary geometry variations using right cylinders. The authors acknowledge that this work is only a first step towards truly arbitrary geometry expression. They found that the optimal cone-like core configuration has a truncated annular cone geometry, with the inlet surface being larger than the outlet [8].

See et al. [11] conducted design optimization of the TCR's outlet plenum design with a Siemens design optimization tool, HEEDS, and the Computational Fluid Dynamics (CFD) code, STAR-CCM+. They had to design an instrumentation plane to monitor core coolant flow average temperature within $\pm 5^{\circ}\text{C}$ while maintaining a tight design constraint of 0.5-psi pressure drop across the outlet plenum. The optimal design maintained a pressure drop to 0.49 psi with an instrumentation plane temperature standard deviation of 1.03°C . Figure 2.5 shows the initial and optimal outlet plenum designs. Figure 2.6 demonstrates how the optimal design impacted the stream tubes

and enabled a smaller temperature standard deviation in the instrumentation plane.



(a) Geometric representation of outer wall spline. (b) Geometric representation of inlet channels.

Figure 2.5: Initial design versus the optimal design of outlet plenum from the study by See et al. The optimal design meets the following design objectives: 0.5-psi pressure drop across the outlet plenum, monitor core coolant flow average temperature within $\pm 5^{\circ}\text{C}$. Figures reproduced from [57].

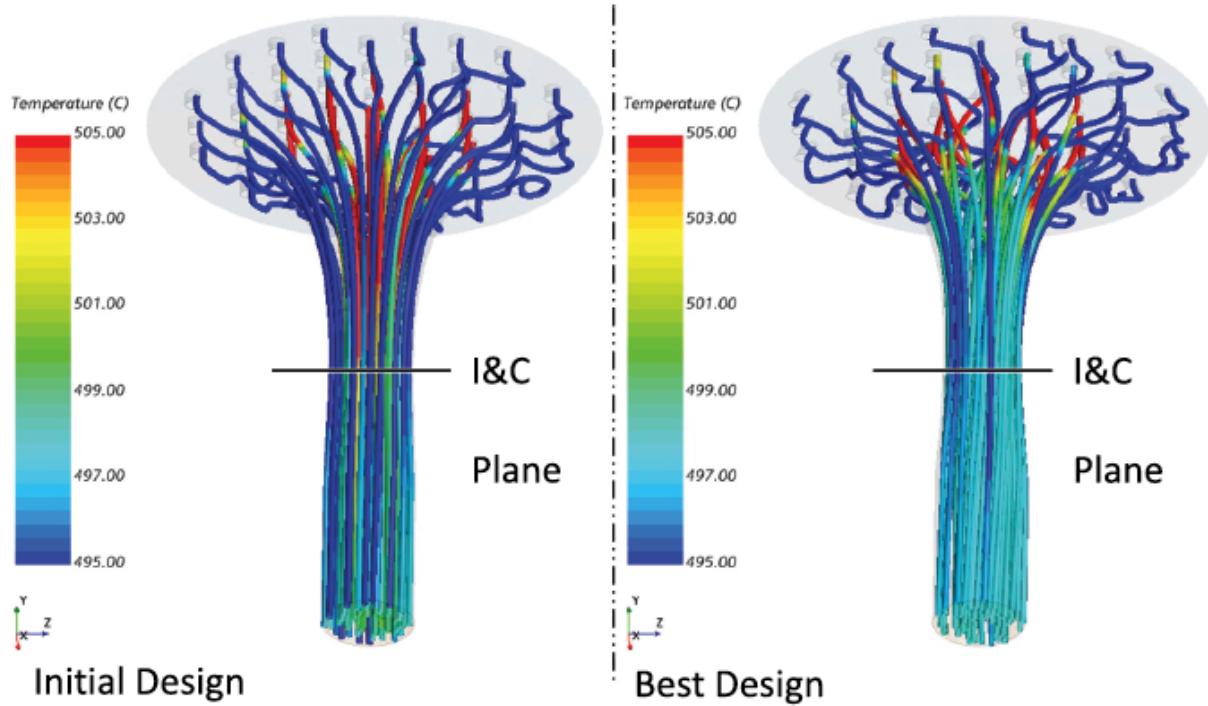


Figure 2.6: Initial design versus optimal design of stream tubes in the outlet plenum from the study by See et al. The optimal design meets the following design objectives: 0.5-psi pressure drop across the outlet plenum, monitor core coolant flow average temperature within $\pm 5^{\circ}\text{C}$. Figures reproduced from [11].

Reactor optimization for arbitrary parameters is a new concept within the reactor optimization field, and only a few research demonstrations have begun exploring the large new design space. Other fields that require 3D design optimization, such as architecture and mechanical design, are further along than the nuclear reactor optimization industry and have coined this type of optimization as generative design [78]. As additive manufacturing technology advances and the TCR program demonstrates the first 3D printed operational reactor, more reactor designers will begin to explore the huge design space enabled by 3D printing. Thoroughly exploring the design space using additive manufacturing should allow the placement of fuel, moderation, and coolant material in any possible location, within physical limits. In this dissertation, I will begin to explore the large design space while also acknowledging (like the authors in [8]) that this work is only an intermediate step towards developing a truly arbitrary geometry expression.

2.4 Evolutionary Algorithms

The expanded design space associated with an arbitrary reactor geometry increases the time required for reactor designers to thoroughly explore and find optimal geometries. Instead, reactor designers can leverage AI optimization methods, such as evolutionary algorithms, to rapidly explore the large design space to find global optimal designs. AI does not replace the human reactor designer, but shifts the human designer's focus away from conjecturing suitable geometries to defining design criteria to find optimal designs [8]. Thus, when the human designer changes the reactor criteria, the AI model will quickly adapt and produce new global optimal designs to fit the new criteria. This optimization method is also known as generative design. In this dissertation, I utilize the evolutionary algorithm optimization method for generative reactor design as evolutionary algorithms have proven to find globally optimal solutions robustly and can take advantage of parallel systems.

Evolutionary algorithms create a population of individual solutions inspired by biological evolution and induce goals by using a ‘fitness function’ to mutate and preferentially replicate high-scoring individuals to reach an optimal solution. Evolutionary algorithms often perform well at approximating solutions to many problem types because they do not make assumptions about the underlying fitness landscape. Genetic algorithms are the most popular evolutionary algorithms for

solving multi-objective problems [64, 79]. In this dissertation, I use the terms evolutionary and genetic algorithms interchangeably to refer to genetic algorithms.

2.4.1 Genetic Algorithms

Genetic algorithms imitate natural genetics and selection to evolve solutions by maintaining a population of solutions, allowing fitter solutions to reproduce, and letting less fit solutions die off, resulting in final solutions that are better than the previous generations [80]. I will refer to a solution as an individual within the population. Genetic algorithms efficiently exploit historical information to speculate new search points, improving each subsequent population's performance [81]. They are theoretically and empirically proven to provide robust search in complex spaces and are computationally simple yet powerful in their search for improvement [81]. Genetic algorithms trounce deterministic and stochastic simulated annealing optimization methods because they:

1. search from a population of points
2. use objective function information, not derivatives or other auxiliary knowledge of the problem
3. use probabilistic transition rules, not deterministic rules

Figure 2.7 depicts the iterative process of using a genetic algorithm to solve a problem. The genetic algorithm generates new populations iteratively until it meets the termination criteria.

Genetic algorithms use mechanisms inspired by biological evolution, such as selection, crossover, and mutation. The three operators are simple and straightforward. The crossover operator recombines good individuals to form a better individual. The mutation operator alters individuals to create better individuals [66]. The selection operator selects good individuals. The selection operator does not create new individuals in the population and only makes more copies of good individuals at the expense of not-so-good individuals. The crossover and mutation operators perform the creation of new solutions. Next, we provide more descriptions and common methods for each operator.

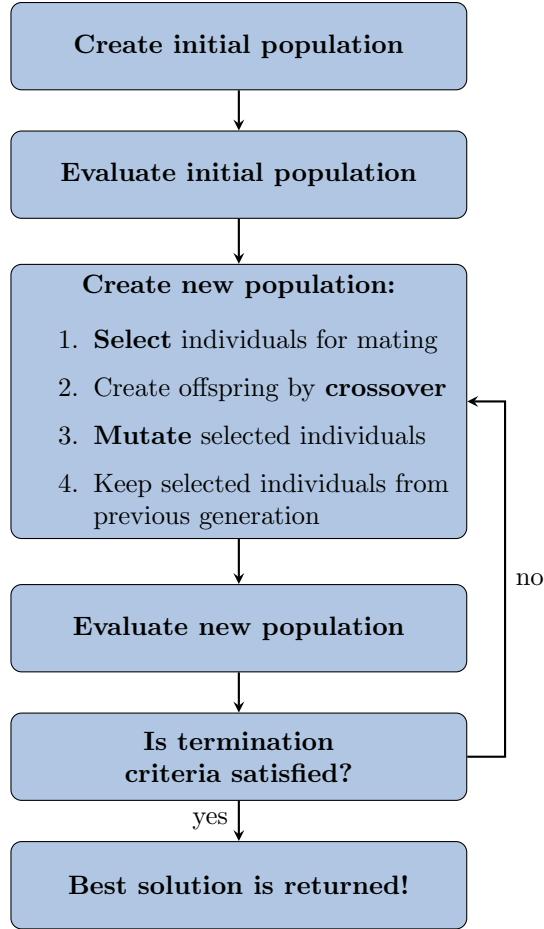


Figure 2.7: Process of finding optimal solutions for a problem with a evolutionary algorithm [80].

Crossover/Mating Operator

In most crossover operators, the operator randomly picks two individuals from the population. The operator exchanges some portion of each individuals' attributes with one another to create two new individuals [66]. Crossover operator methods utilized in this dissertation include *single-point crossover*, *uniform crossover*, and *blend crossover*. In the *single-point crossover*, the operator randomly selects two individuals from the population and a site along the individual's definition. For example, if the individual is a list, the operator randomly chooses an element in the list as the cross-site. Then, the attributes on the cross site's right side are exchanged between the two individuals, creating two new offspring individuals. In a *uniform crossover*, the user defines an independent exchange probability for each individual's attribute. In *blend crossover*, the operator creates two offspring (O) individuals based on a linear combination of two-parent (P) individuals

using the following equations:

$$O_1 = P_1 - \alpha(P_1 - P_2) \quad (2.4)$$

$$O_2 = P_2 + \alpha(P_1 - P_2) \quad (2.5)$$

where

α = Extent of the interval in which the new values can be drawn

for each attribute on both side of the parents' attributes (user-defined)

The user defines a crossover probability (p_c) to preserve some good individuals selected during the selection operator stage. Therefore, the crossover operator only operates on $100p_c\%$ of the population; the rest proceed to the new population [66]. The crossover operator covers the search aspect of the genetic algorithms, whereas the mutation operator keeps diversity in the population [66].

Mutation Operator

The mutation operator alters one or more attributes of an individual within a population. Mutation operator methods utilized in this dissertation include *polynomial bounded mutation*, in which each attribute in each individual is mutated based on a polynomial distribution. The user also defines each attribute's upper and lower bounds and the crowding degree of the mutation, η (a big η will produce a mutant resembling its parent, while a small η will produce the opposite). Mutation occurs in the genetic algorithm based on a user-defined mutation probability (p_m). A low p_m prevents a primitive random search.

Selection Operator

The selection operator duplicates good individuals and eliminates bad individuals while keeping the population constant [66]. It achieves this by identifying above-average individuals, eliminating bad individuals from the population, and replacing them with copies of good individuals. Selection

operator methods utilized in this dissertation include *tournament selection*, *best selection*, and *Non-dominated Sorting Genetic Algorithm II (NSGA-II) selection*. In *tournament selection*, a user-defined number of individuals play in a tournament, and the best individual proceeds to the next population. The tournament repeats until all the population’s spots are filled. In *best selection*, the operator selects a user-defined number of best individuals, and copies are made to keep the population size constant. In *NSGA-II selection*, the elitist operator selects the best individuals from the combination of parent and offspring populations [82]. *NSGA-II selection* works well for multi-objective optimization.

2.4.2 Genetic Algorithm Hyperparameter Tuning

Hyperparameters refer to parameters whose value controls the genetic algorithm’s process, such as the population size. A well-performing genetic algorithm needs to balance the extent of exploration and exploitation; by finding a balance between the conservation of valuable individuals obtained until the current generation while exploring new individuals. With overexploitation of previously obtained individuals, the population loses diversity, resulting in premature convergence to a sub-optimal solution. Alternatively, with over-exploration, the algorithm does not appropriately utilize the information obtained thus far, and the genetic algorithm’s search procedure behaves like a random search process [66].

A quantitative balance between these two issues, exploitation and exploration, is challenging to achieve. Deb et al. [66] and Goldberg et al. [83] quantified the relationship between exploitation and exploration. They found that for the one-max test problem, in which the objective seeks to maximize the number of 1s in a string, a genetic algorithm with any arbitrary hyperparameter setting does not work well even on a simple problem. Only genetic algorithms with a selection pressure (s) and crossover probability (p_c) falling inside the control map (Figure 2.8) will find the desired optimum. Another consideration is the population size. A function with considerable variability in objective function values demands a large population size to find a global optimum [66]. Therefore, finding an optimized solution with genetic algorithms requires the user to conduct a hyperparameter search.

Ng et al. [84] suggest that a coarse-to-fine sampling scheme is the best way to perform a

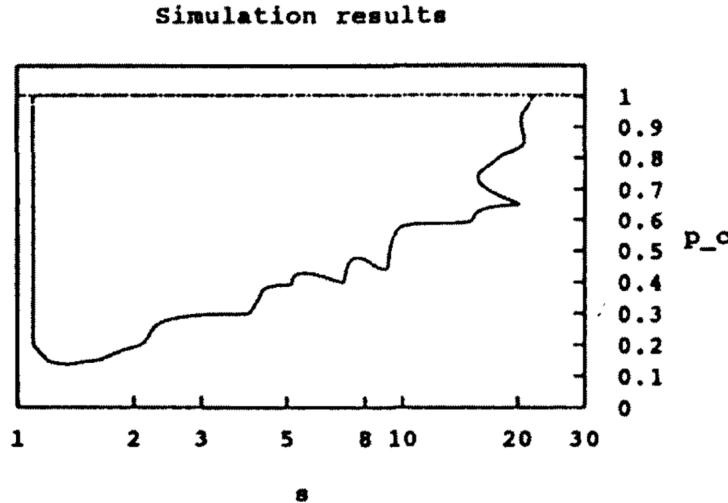


Figure 2.8: Control map region of selection pressure (s) and crossover probability (p_c) values in which the genetic algorithm will find the desired optimum for the one-max problem. Figure reproduced from [83, 66].

systematic hyperparameter search. For a two-dimensional example of a coarse-to-fine sampling scheme, the user first does a coarse sample of the entire square, then a fine search on the coarse search's best-performing region. Ng et al. also suggest using random sampling over grid sampling because of the former's efficiency in high-dimensional spaces. Figure 2.9 illustrates how grid sampling gives even coverage in the original 2-d space but provides inefficient coverage in projections onto either the x_1 or x_2 subspace. In contrast, random sampling produces a less even distribution in the original space but a far more even distribution in the subspaces.

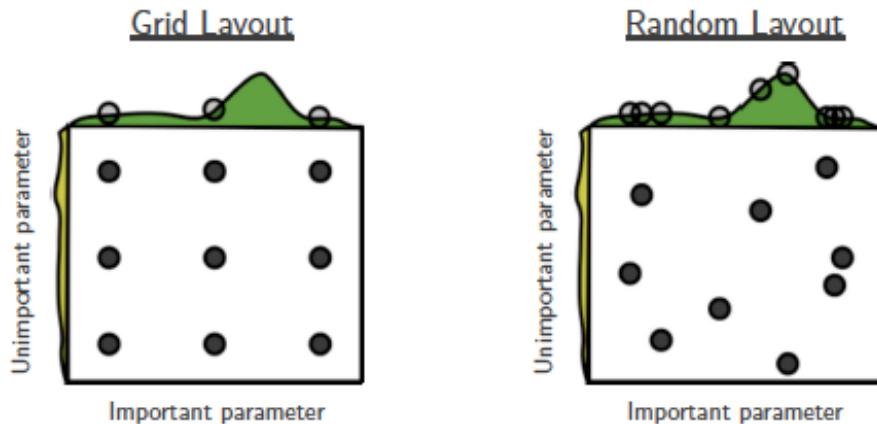


Figure 2.9: The impact of grid sampling vs random sampling on coverage of projections into subspaces (reproduced from [85]). Random sampling has better coverage in the subspaces.

2.5 Summary

This chapter provided a literature review of relevant past research efforts that give context to and support the work in this dissertation. In summary, participation in the Organisation for Economic Co-operation and Development (OECD)-Nuclear Energy Agency (NEA) Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmarking exercise contributes to assessing the current neutron transport and thermal-hydraulics modeling and simulation capabilities for the Advanced High-Temperature Reactor (AHTR) design. Furthermore, additive manufacturing of nuclear reactor components is a quickly developing field due to advancements driven by the aerospace and automotive industries, which led to breakthroughs in metal component additive manufacturing fabrication. The promise of cheaper and faster manufacturing of reactor components with additive manufacturing frees complex reactor geometries from previous manufacturing constraints and allows reactor designers to reexamine reactor design optimization. Stochastic optimization methods such as evolutionary algorithms have proven to work well for finding global optima in multi-objective design problems such as nuclear reactor optimization and can be leveraged to explore the vast exploration design space enabled by additive manufacturing.

Chapter 3

Fluoride-Salt-Cooled High-Temperature Reactor Benchmark

Fluoride-Salt-Cooled High-Temperature Reactor (FHR) systems use Tristructural Isotropic (TRISO) fuel and a low-pressure liquid fluoride-salt coolant. FHR technology combines Fluoride-Lithium-Beryllium (FLiBe) coolant from Molten Salt Reactors (MSRs) and TRISO particles from Very-High-Temperature Reactors (VHTRs) to enable a reactor with low operating pressure, a large thermal margin, and accident-tolerant qualities. Within the FHR reactor class, Advanced High-Temperature Reactors (AHTRs) have plate-based fuel in a hexagonal fuel assembly. In section 2.1, I gave an FHR concept overview, an AHTR design description, a review of previous efforts towards modeling these designs, and how these efforts led to the FHR benchmark initiation.

To address the AHTR modeling challenges described in Chapter 2, such as multiple heterogeneity and material cross-section data, the Organisation for Economic Co-operation and Development (OECD)-Nuclear Energy Agency (NEA) and Georgia Institute of Technology (Georgia Tech) initiated the FHR benchmark for the AHTR design in 2019 [18]. University of Illinois at Urbana-Champaign (UIUC) participates in the FHR benchmark with the OpenMC Monte Carlo code [19] using the ENDF/B-VII.1 material cross section library [20]. The UIUC team consists of myself and my advisors, Professor Kathryn Huff and Dr. Madicken Munk.

The three-phase FHR benchmark begins with a single fuel assembly simulation without burnup and gradually extends to full core depletion. Table 3.1 outlines the complete and incomplete benchmark phases as of August 2022.

In the subsequent sections, I will describe the benchmark’s AHTR design details (Section 3.1 and Phase I specifications (Section 3.2. Then, I will share our Phase I-A and I-B results generated with the OpenMC neutronics code [19] (Section 3.3) and the AHTR temperature model’s results (Section 3.4). Appendix A lists all the data and analysis related to this chapter to enable the reproduction of all the simulations.

Table 3.1: Organisation for Economic Co-operation and Development (OECD) Nuclear Energy Agency (NEA) Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark Phases [18].

Phases	Sub-phases	Description	Completed?
Phase I: fuel assembly	I-A	2D model, steady-state	✓
	I-B	2D model depletion	✓
	I-C	3D model depletion	
Phase II: 3D full core	II-A	Steady-state	
	II-B	Depletion	
Phase III: 3D full core with feedback & multicycle analysis	III-A	Full core depletion with feedback	
	III-B	Multicycle analysis	

3.1 FHR Benchmark Advanced High-Temperature Reactor Design

Figure 3.1 shows the Advanced High-Temperature Reactor (AHTR) schematic and a vertical cut of the reactor vessel. The AHTR operates at 3400 MWt thermal power and 1400 MWe electric power [29]. The 10m-diameter exterior reactor vessel contains an 8m-diameter reactor core that contains 252 hexagonal fuel assemblies.

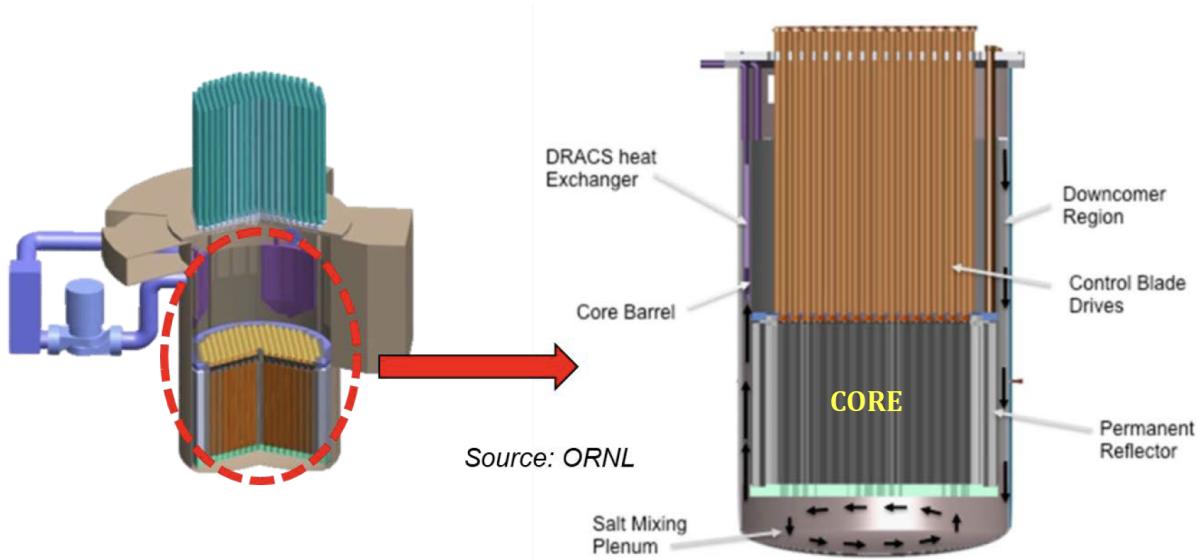


Figure 3.1: Advanced High-Temperature Reactor (AHTR) schematic (left) and vessel (right) reproduced from [18].

Each 6m high fuel assembly comprises a 5.5m active core region containing TRISO particles and 0.25m top and bottom non-fuelled reflector regions. Figure 2.1, from Chapter 2, shows a single

hexagonal fuel assembly geometry and the arrangement of all assemblies in the core. All dimensions specified are at room temperature. The benchmark's phases I and II use room temperature dimensions, while Phase III will address dimensional changes brought about by temperature expansion. Figure 3.2 shows a x-y plane view of the AHTR's hexagonal fuel assembly. The hexagonal fuel assembly consists of eighteen fuel-containing graphite planks arranged in three diamond-shaped sectors, with an external channel wrapper and structural Y-shape, made of C-C composite with extra notches to hold the fuel planks in place. The diamond-shaped sections have 120° rotational symmetry with each other [29, 31, 18]. Semi-cylindrical graphite spacers attach to the fuel planks with a radius equalling to coolant channel thickness. FLiBe coolant fills the gaps between the fuel planks and assemblies (note: FLiBe layer around the single assembly). The Y-shaped control rod slot at the center of the Y-shape structure contains FLiBe coolant when the control blade is not in the slot (as seen in Figure 3.2) [29, 31, 18]. For a single fuel assembly, the internal 120° rotational symmetry is represented by periodic boundary conditions, as seen in Figure 3.3.

Figure 3.4 magnifies a single fuel plank. Fuel stripes line the upper and lower sides of each graphite fuel plank. Each fuel stripe contains a graphite matrix filled with a cubic lattice of TRISO particles with 40% packing fraction. The lattice is 210 TRISO particles wide in the x-direction, four particles deep in the y-direction, and 5936 particles tall in the z-direction. Figure 3.5 shows the TRISO particle's cross section, consisting of five layers: oxycarbide fuel kernel, porous carbon buffer, inner pyrolytic carbon, silicon carbide layer, and the outer pyrolytic carbon.

The FHR benchmark includes AHTR configurations with burnable poisons and control rods to control reactivity. The burnable poisons consist of europium oxide (Eu_2O_3) and have a discrete or integral (dispersed) option. Figure 3.6 shows the discrete option with z-direction axially stacked small spherical Eu_2O_3 particles at five XY locations in each fuel plank. The integral options consist of Eu_2O_3 homogenously mixed with the graphite fuel plank (including the graphite in fuel stripes matrix and plank ends indented to structural sides, but excluding the graphite in spacers and graphite in TRISO particles). The molybdenum–hafnium carbide alloy (MHC) control rod inserts into the Y-shaped control rod slot, displacing the FLiBe that occupies the slot (shown in Figure 3.2). Detailed AHTR geometry and material information can be found in the FHR benchmark's specifications [18].

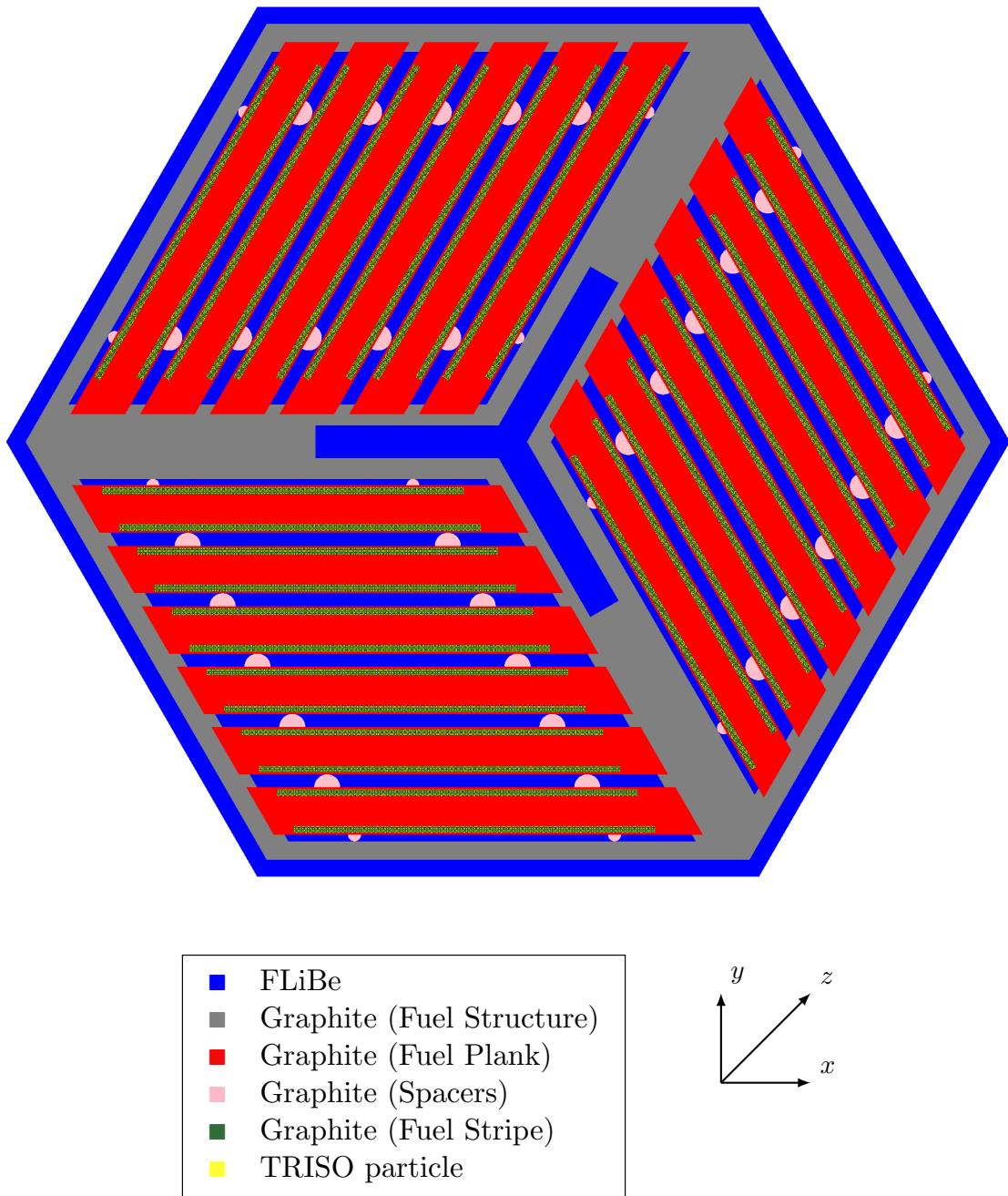


Figure 3.2: The Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark's Advanced High-Temperature Reactor (AHTR) fuel assembly with 18 fuel planks (red) arranged in three diamond-shaped sectors, with a central Y-shaped and external channel graphite structure (grey). TRISO fuel (yellow) is arranged in a lattice structure within fuel stripes (green). Spacers (pink) hold the graphite planks apart. Flibe coolant (blue) flows through the assembly.

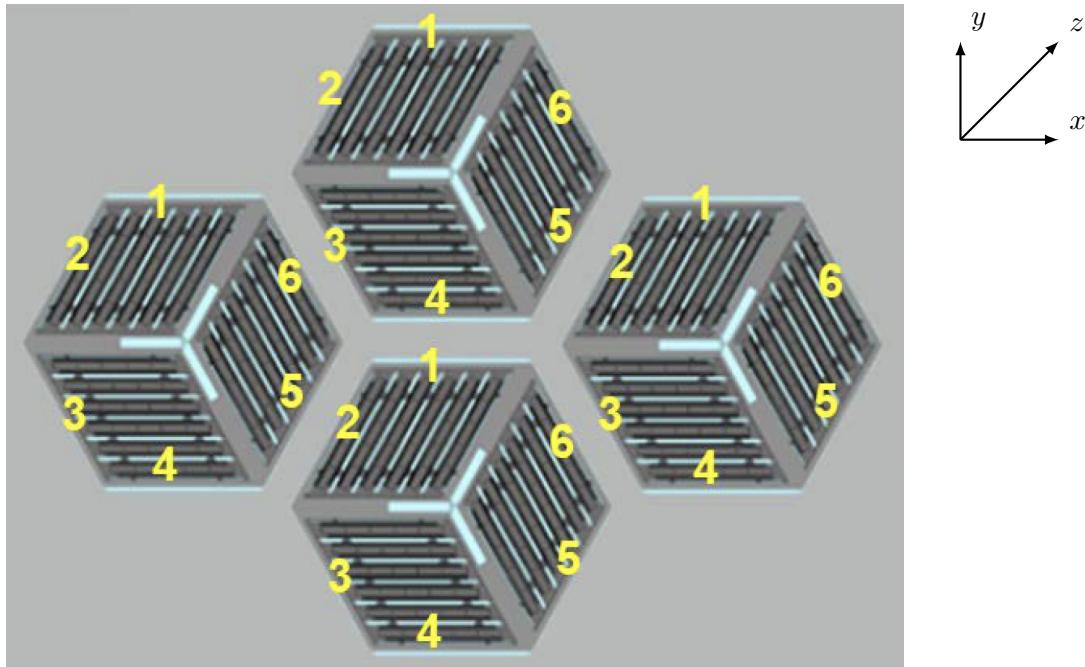


Figure 3.3: Visualization of periodic boundary conditions for a single fuel assembly in the Advanced High-Temperature Reactor (AHTR), reproduced from [18].

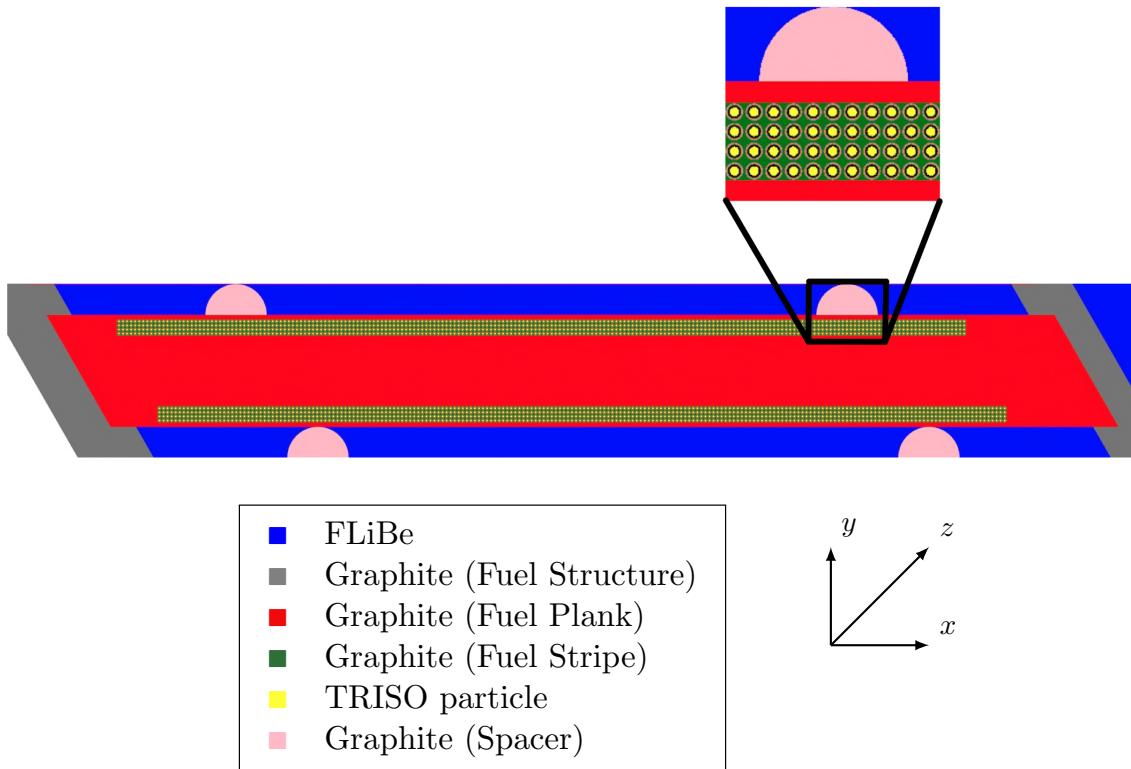


Figure 3.4: Advanced High-Temperature Reactor (AHTR)'s fuel plank, with the magnification of a spacer and segment of the fuel stripe with embedded TRISO particles.

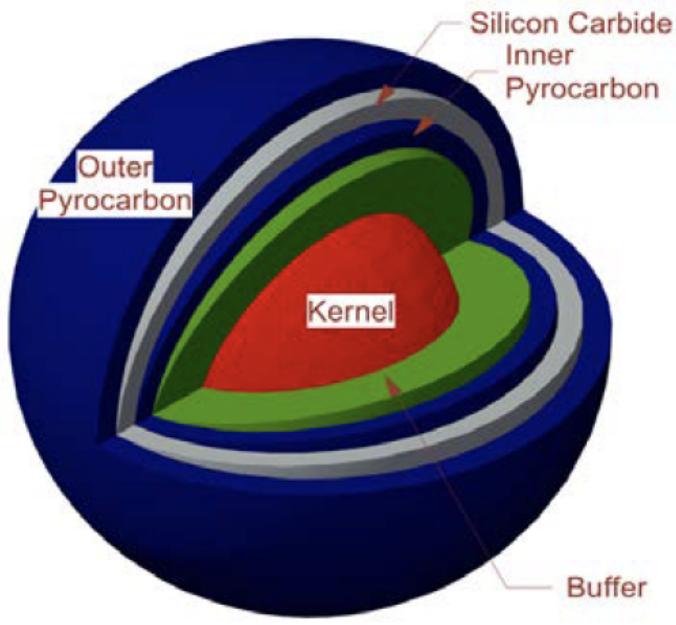


Figure 3.5: Advanced High-Temperature Reactor's TRISO particle schematic reproduced from [18].

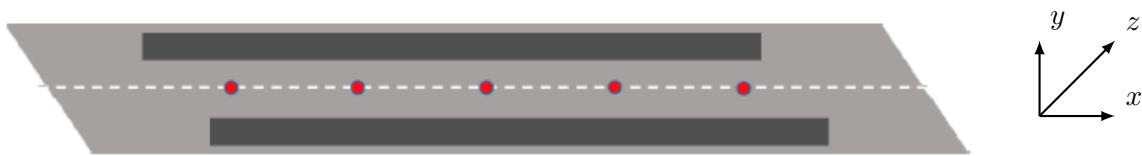


Figure 3.6: XY plane's placement of axial burnable poisons in the Advanced High-Temperature Reactor plank [18]. The burnable poisons are stacked in the z-direction.

3.2 FHR Benchmark Phase I Specifications

The FHR benchmark Phase I consists of three subphases. First, the steady-state 2D model (Phase I-A), next, the depletion of one 2D FHR fuel assembly (Phase I-B), and finally depletion of one 3D FHR fuel assembly (Phase I-C). Benchmark organizers have only released Phase I-A and I-B's specifications. Thus Phase I-C's specifications will be omitted in this section.

The benchmark requires the following results for Phases I-A and I-B:

- (a) k_{eff} (effective multiplication factor)
- (b) reactivity coefficients: β_{eff} (effective delayed neutron fraction), α_D (doppler coefficient), $\alpha_{T,FliBe}$ (FLiBe temperature coefficient), α_M (moderator temperature coefficient)
- (c) tabulated fission source distribution by one-fifth fuel stripe
- (d) $\bar{\phi}_1, \bar{\phi}_2, \bar{\phi}_3$ (neutron flux averaged over the whole model tabulated in three coarse energy groups)
- (e) $\phi_1(\vec{x}, \vec{y}), \phi_2(\vec{x}, \vec{y}), \phi_3(\vec{x}, \vec{y})$ (neutron flux distribution in three coarse energy groups)
- (f) fuel assembly averaged neutron spectrum

Next, I report the equations used to calculate these required results.

Reactivity Coefficients (b)

I assumed one energy group and six delayed groups for β_{eff} . Reactivity coefficient, α , is the change in reactivity (ρ) of the material per degree change in the material's temperature (T). I calculated each reactivity coefficient and its corresponding uncertainty with these equations:

$$\frac{\Delta\rho}{\Delta T} = \frac{\rho_{T_{high}} - \rho_{T_{low}}}{T_{high} - T_{low}} \left[\frac{pcm}{K} \right] \quad (3.1)$$

$$\delta \frac{\Delta\rho}{\Delta T} = \frac{\sqrt{\delta(\rho_{T_{high}})^2 + (\delta\rho_{T_{low}})^2}}{T_{high} - T_{low}} \left[\frac{pcm}{K} \right] \quad (3.2)$$

Fission Source Distribution / Fission Density (c)

I calculated fission density (FD) with OpenMC's `fission` tally score (f) for each region divided by the average `fission` tally score of all the regions:

$$FD_i = \frac{f_i}{f_{ave}} [-] \quad (3.3)$$

where

f_i = fission reaction rate in a single region i [$\text{reactions} \cdot \text{src}^{-1}$]

f_{ave} = average of all f_i [$\text{reactions} \cdot \text{src}^{-1}$]

The uncertainty calculations for FD_i and f_{ave} :

$$\delta FD_i = |FD_i| \sqrt{\left(\frac{\delta f_i}{f_i}\right)^2 + \left(\frac{\delta f_{ave}}{f_{ave}}\right)^2} \quad (3.4)$$

$$\delta f_{ave} = \frac{1}{N} \sqrt{\sum_i^N f_i^2} \quad (3.5)$$

where

N = No. of fission score values [-]

Neutron Flux (d, e, f)

OpenMC's `flux` score is normalized per source particle simulated, resulting in [$\frac{\text{neutrons cm}}{\text{src}}$] units. This is an unnatural unit for system analysis, and thus to better compare OpenMC results with other software results in the benchmark, I converted flux to [$\frac{\text{neutrons}}{\text{cm}^2 \text{s}}$] units using the following equations:

$$\Phi_c = \frac{N \cdot \Phi_o}{V} \quad (3.6)$$

$$N = \frac{P \cdot \nu}{Q \cdot k} \quad (3.7)$$

where

$$\Phi_c = \text{converted flux} [\text{neutrons} \cdot \text{cm}^{-2} \cdot \text{s}^{-1}]$$

$$\Phi_o = \text{original flux} [\text{neutrons} \cdot \text{cm} \cdot \text{src}^{-1}]$$

$$N = \text{normalization factor} [\text{src} \cdot \text{s}^{-1}]$$

$$V = \text{volume of fuel assembly} [\text{cm}^3]$$

$$P = \text{power} [\text{J} \cdot \text{s}^{-1}]$$

$$\nu = \frac{\nu_f}{f} [\text{neutrons} \cdot \text{fission}^{-1}]$$

$$Q = \text{Energy produced per fission} [\text{J} \cdot \text{fission}^{-1}]$$

$$= 3.2044 \times 10^{-11} \text{ J per } U_{235} \text{ fission}$$

$$k = k_{eff} [\text{neutrons} \cdot \text{src}^{-1}]$$

The flux standard deviation is:

$$\delta\Phi_c = \Phi_c \times \sqrt{\left(\frac{\delta\Phi_o}{\Phi_o}\right)^2 + \left(\frac{\delta\nu_f}{\nu_f}\right)^2 + \left(\frac{\delta k}{k}\right)^2 + \left(\frac{\delta f}{f}\right)^2} \quad (3.8)$$

I calculated reactor power based on the given reference specific power (P_{sp}) of 200 $\frac{W}{gU}$:

$$P = P_{sp} \times V_F \times \rho_F \times \frac{wt\%_U}{100} \quad (3.9)$$

where

$$V_F = \text{volume of fuel} [\text{cm}^3]$$

$$= \frac{4}{3}\pi r_f^3 \times N_{total}$$

$$r_f = \text{radius of fuel kernel within TRISO particle} [\text{cm}]$$

$$N_{total} = \text{total no. of TRISO particles in fuel assembly} [\#]$$

$$= 101 \times 210 \times 4 \times 2 \times 6 \times 3$$

$$\rho_F = \text{density of fuel} [g \cdot \text{cm}^{-3}]$$

$$wt\%_U = \frac{at\%_{U235} \times m_{U235} + at\%_{U238} \times m_{U238}}{\sum(at\%_{fuel} \times m_{fuel})} \times 100$$

m = atomic mass [amu]

3.2.1 Phase I-A Specifications

For Phase I-A, the benchmark specifies that each participant must produce a steady-state 2D model of one fresh fuel assembly for nine cases and report the required results listed in Section 3.2. Table 3.2 describes each case.

Table 3.2: Description of the Fluoride-Salt-Cooled High-Temperature Reactor benchmark Phase I-A cases [18].

Case	Description
1A	Reference case. Hot full power (HFP), with temperatures of 1110K for fuel kernel and 948K for coolant and all other materials (including TRISO particle layers other than fuel kernel). Nominal (cold) dimensions, 9 wt% enrichment, no burnable poison, control rod out.
2AH	Hot zero power (HZP) with uniform temperature of 948 K, otherwise same as Case 1A. Comparison with Case 1A provides HZP-to-HFP power defect.
2AC	Cold zero power (CZP). Same as Case 2AH, but with uniform temperature of 773 K. Comparison with Case 2AH provides isothermal temperature coefficient.
3A	Control rod inserted, otherwise same as Case 1A.
4A	Discrete europia burnable poison, otherwise same as Case 1A.
4AR	Discrete europia burnable poison and control rod inserted, otherwise same as Case 1A.
5A	Integral (dispersed) europia burnable poison, otherwise same as Case 1A.
6A	Increased heavy metal (HM) loading (4 to 8 layers of TRISO) decreased C/HM ratio (from about 400 to about 200) and decreased specific power to 100 W/gU, otherwise same as Case 1A.
7A	Fuel enrichment 19.75 wt%, otherwise same as Case 1A.

3.2.2 Phase I-B Specifications

For Phase I-B, the benchmark specifies that each participant must produce depletion results for three cases: 1B, 4B, and 7B. These are the same as cases 1A, 4A, and 7A (described in Table 3.2), but with depletion steps added. The depletion steps are 0, 0.1, 0.5, 1, 2, 4, 6, 8, 10, 14, 18, 22, 26, 30, 40, 50, 60, and 70 GWd/tU. Case 7B adds seven more depletion steps: 80, 90, 100, 120, 140, and 160 GWd/tU. The benchmark assumes that depletion occurs only in the fuel and burnable poisons and that the depletion performs under the critical spectrum assumption.

3.3 FHR Benchmark Phase I Results

The `arfcc/fhr-benchmark` Github repository [86] contains all the results submitted by UIUC for the FHR benchmark. The benchmark used a phased blind approach – participants were asked to submit Phase I-A and I-B results without knowledge of other submissions. Petrovic et al. [45] describes the preliminary results of the benchmark across several institutions and concludes that the overall observed agreement is satisfactory. In the subsequent sections, I will share the results obtained by UIUC.

3.3.1 Results: Phase I-A

In an American Nuclear Society (ANS) Mathematics & Computation (M&C) 2021 conference, Petrovic et al. [45] compared FHR benchmark participants' Phase I-A k_{eff} results. We reported that the standard deviation between participants for each case was in the 231 to 514 pcm range, acceptable and notably close given a blind benchmark, assuring that UIUC's Phase I-A results are acceptable and in agreement with other benchmark participants.

Results: Effective Multiplication Factor (a)

Table 3.3 reports Phase I-A k_{eff} results. I ran the simulations on UIUC's BlueWaters supercomputer with 64 XE nodes, which each have 32 cores [87]. To reduce the statistical uncertainty of k_{eff} to ~ 10 pcm, I ran each simulation with 500 active cycles, 100 inactive cycles, and 200000 neutrons. Each simulation took wall-clock-time (WCT) ranging from 2 to 5 hours.

Cases 2AH and 2AC are at zero power, meaning that the fuel assembly is exactly critical but not producing any energy. For both cases, k_{eff} is higher than the reference Case 1A, which I attribute to lower fuel temperatures. At lower fuel temperatures, less doppler broadening occurs, resulting in less neutron capture, thus, increasing k_{eff} . As expected, k_{eff} is lower for Cases 3A, 4AR, and 5A than reference case 1A since these cases introduce burnable poisons and control rods to the fuel assembly. Also, as expected, k_{eff} is higher for Case 7A than reference Case 1A since it has a higher enrichment. However, Case 6A deviated from expectations with a lower k_{eff} despite an increase in heavy metal loading. This behavior is due to reduced moderation and worsened fuel utilization brought about by self-shielding, demonstrating that increased fuel packing fraction does

Table 3.3: UIUC's Fluoride-Salt-Cooled High-Temperature Reactor Benchmark Phase I-A results [86].

Case	Summary	WCT [hr]	k_{eff}^*	β_{eff}^{**}	Fuel $\frac{\Delta\rho}{\Delta T}$	FliBe $\frac{\Delta\rho}{\Delta T}$	Graphite $\frac{\Delta\rho}{\Delta T}$
1A	Reference	2.82	1.39389	0.006534	-2.24±0.15	-0.15±0.15	-0.68±0.15
2AH	HZP	2.82	1.40395	0.006534	-3.14±0.15	-0.20±0.14	-0.85±0.14
2AC	CZP	2.75	1.41891	0.006534	-3.36±0.14	-0.11±0.14	0.07±0.14
3A	CR	2.49	1.03147	0.006534	-4.03±0.28	-0.83±0.27	-3.18±0.29
4A	Discrete BP	5.08	1.09766	0.006542	-4.06±0.24	-1.55±0.23	-6.51±0.24
4AR	Discrete BP + CR	4.59	0.84158	0.006553	-5.60±0.49	-1.78±0.46	-10.44±0.47
5A	Dispersed BP	2.33	0.79837	0.006556	-5.09±0.40	-4.87±0.40	-22.99±0.38
6A	Increased HM	3.52	1.26294	0.006556	-4.46±0.19	0.16±0.20	-0.39±0.20
7A	19.75% Enriched	2.21	1.50526	0.006530	-2.49±0.13	-0.12±0.12	-0.62±0.12

BP: burnable poison, CR: control rod

* All k_{eff} values have an uncertainty of 0.00010.

** All β_{eff} values have an uncertainty of 0.000001.

not always correspond with an increased k_{eff} .

Results: Reactivity Coefficients (b)

Table 3.3 reports Phase I-A reactivity coefficients results. β_{eff} increased by 10-20 [pcm] for Cases 4A, 4AR, 5A, and 6A compared to reference Case 1A due to the introduction of control rods and poisons that shift the average neutron velocity to higher values, resulting in decreased thermal fission and increased fast fission [88]. In Table 3.3, most temperature coefficients are negative, exemplifying the AHTR's passive safety behavior. Negative reactivity feedback results in a self-regulating reactor; if the reactor power rises, resulting in a temperature increase, the negative reactivity reduces power.

Results: Fission Source Distribution (c)

Figure 3.7 shows Cases 1A and 3A's fission source distribution discretized by one-fifth fuel stripe. Case 4AR has a similar fission source distribution to Case 3A since both cases have control rod insertion. Case 7A has a similar fission source distribution to case 6A since both have higher heavy metal loading. All other cases have similar fission source distributions to Case 1A.

For Case 1A, intuitively, one might assume that the highest fission source would occur in the center of the diamond fuel segment; however, the opposite is true. Power peaking occurs on exterior stripes and is minimum on the interior stripes. Gentry et al. [39] reported similar power peaking

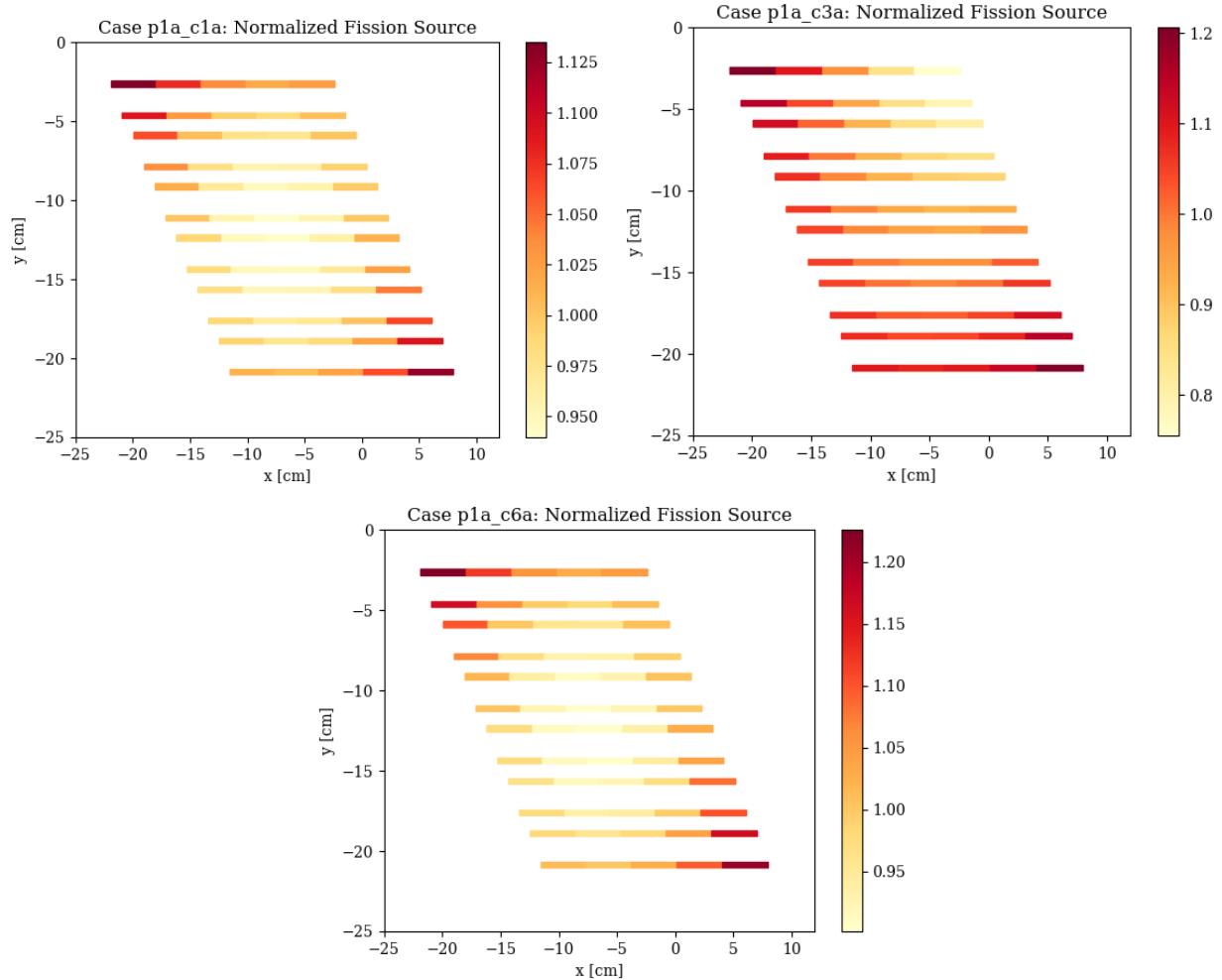


Figure 3.7: UIUC FHR benchmark results: normalized fission source distribution [-] per one-fifth fuel stripe for Fluoride-Salt-Cooled High-Temperature Reactor Benchmark’s Phase I-A Case 1A (top left), Case 3A (top right), and Case 6A (bottom).

phenomena towards the lattice cell’s exterior closest to the Y-shaped carbon support structure. This fission source distribution is caused by diminished resonance escape probability in the interior due to the higher relative fuel-to-carbon volume ratio.

For Case 3A with an inserted control rod, the fission source is lower in the one-fifth stripes closer to the control rod. Case 6A demonstrates a further diminished fission source in the interior stripes due to the higher fuel-to-carbon ratio. This is seen in Figure 3.7, in which case 1A and 6A have similar fission distribution trends, but case 6A has a bigger fission source value range.

Results: Average Neutron Flux (d)

Figure 3.8 shows the average neutron flux in the fuel assembly in three coarse energy groups. Most cases have the most flux in the intermediate group, followed by the thermal group, and the least flux in the fast group.

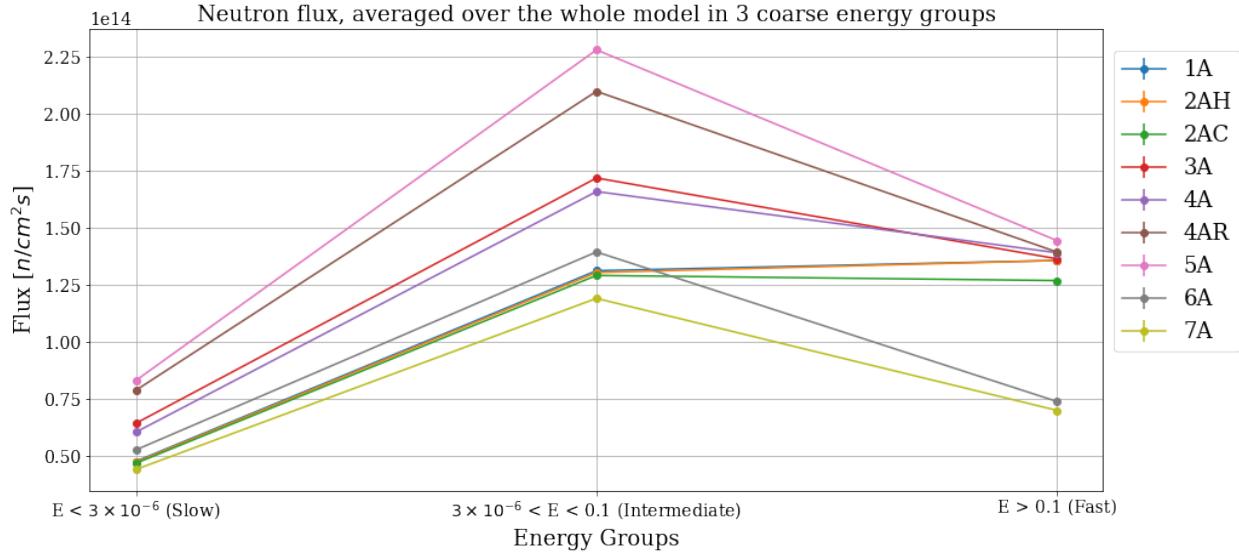


Figure 3.8: UIUC FHR benchmark results: neutron flux, averaged over the whole model, tabulated in three coarse energy groups for each Phase I-A case. Neutron flux uncertainty is on the order of 1e10.

Results: Neutron Flux Distribution (e)

Figure 3.9 shows the neutron flux distribution in a 100×100 mesh for Cases 1A, 3A, and 6A for three coarse energy groups. For all three cases, fast-flux (energy group 1) peaks in the diamond-shaped sectors containing the fuel stripes, whereas thermal flux (energy group 3) peaks outside the diamond-shaped sectors. This can be attributed to fission occurring at thermal energies in the fuel stripe area. For Case 3A, the thermal and intermediate neutron flux is depressed in the fuel assembly's control rod region. An increased heavy metal loading in Case 6A results in a more pronounced fast-flux peaking and thermal flux dip in the fuel stripe area.

Results: Neutron Spectrum (f)

Figure 3.10 shows the neutron spectrum for Cases 1A and 6A. Case 7A has a similar neutron

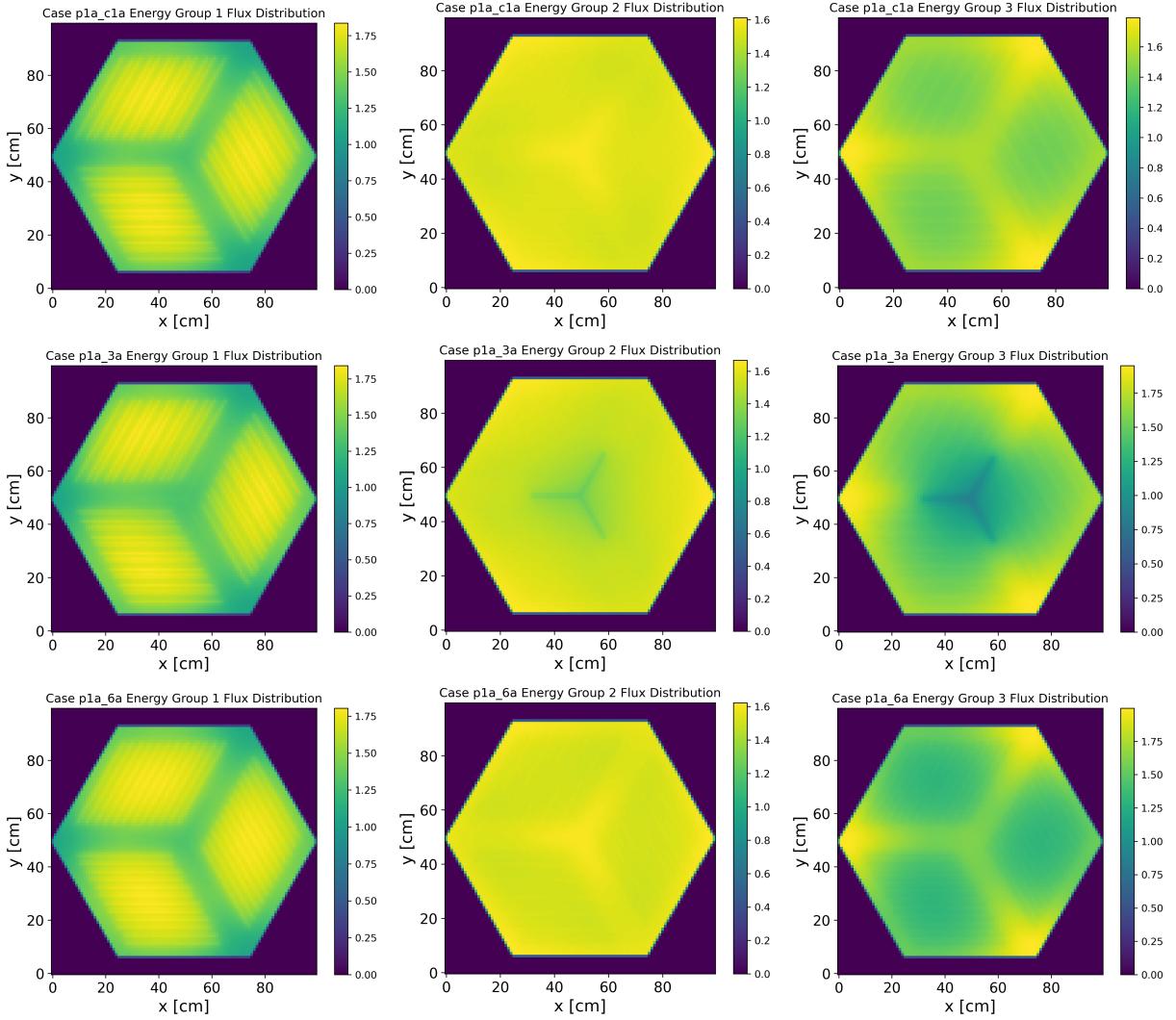


Figure 3.9: UIUC FHR benchmark flux results: neutron flux distribution in 100×100 mesh for three coarse energy groups normalized by each group's average flux. Three cases are shown: Case 1A (above), Case 3A (middle), Case 6A (below). Energy group 1: $E > 0.1$ MeV, Energy group 2: $3 \times 10^{-6} < E < 0.1$ MeV, Energy group 3: $E < 3 \times 10^{-6}$ MeV.

spectrum to Case 6A since both cases have higher fuel content. All other cases have a similar neutron spectrum to Case 1A. The neutron spectra in Cases 6A and 7A are faster due to more heavy metal loading and higher enrichment, respectively.

3.3.2 Results: Phase I-B

Figure 3.11 shows the k_{eff} evolution during depletion for Cases 1B, 4B, and 7B. The k_{eff} at zero burnup corresponds to each case's Phase I-A k_{eff} value reported in Table 3.3. Case 1B is the

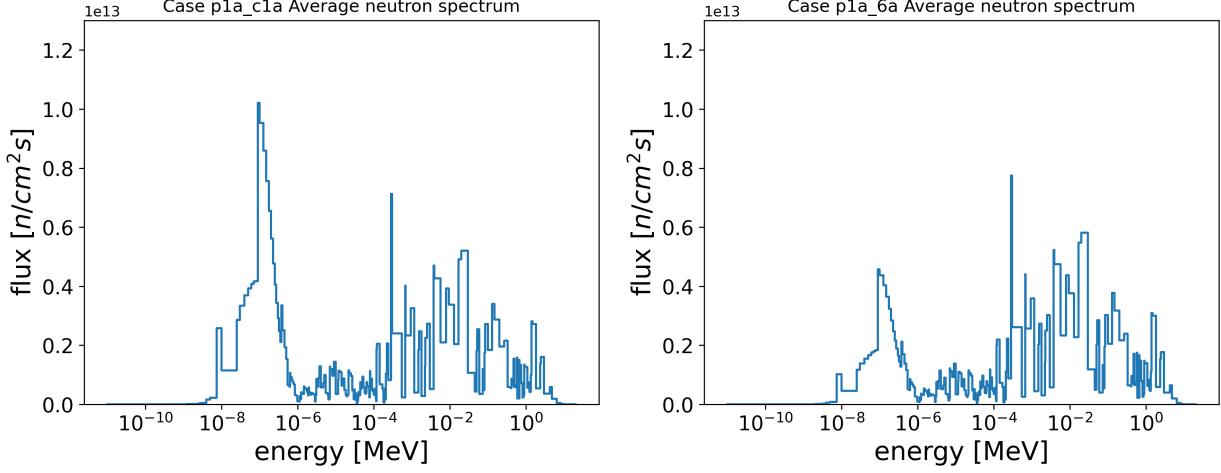


Figure 3.10: UIUC FHR benchmark results: neutron spectrum for Phase I-A Case 1A (left) and Case 6A (right).

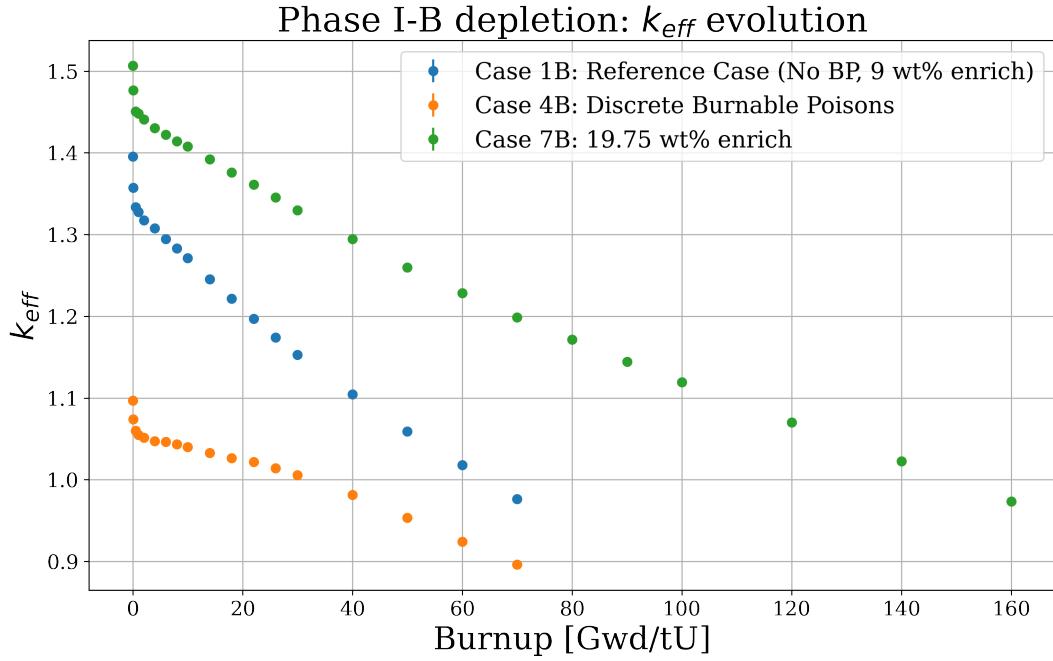


Figure 3.11: UIUC FHR benchmark results: Phase I-B depletion k_{eff} evolution for Cases 1B, 4B, and 7B. Case 1B is the reference case, Case 4B is the discrete burnable poison case, and Case 7B is the 19.75% enrichment case. Error bars are included but are barely visible due to the low ~ 60 pcm uncertainty.

reference case with 9% fuel enrichment and no burnable poisons (BPs). Case 1B's k_{eff} steadily decreases until it reaches 0.967845 at the final 70 GWd/tU burnup. Case 4B includes burnable

poisons resulting in a lower initial k_{eff} . Case 4B's k_{eff} decreases at a slower rate in the beginning due to the presence of burnable poisons, which decreases the flux in the core. At approximately 20 GWd/tU, k_{eff} begins decreasing faster, presumably due to burn-up of the poison material. Case 7B has a 19.75% fuel enrichment, resulting in a higher initial k_{eff} . With a higher enrichment, the assembly stays critical till 140 Gwd/tU.

3.4 FHR Benchmark Temperature Model

I used Moltres [89] to conduct AHTR full assembly temperature simulations. AHTR Moltres simulations will capture thermal feedback effects absent from the purely neutronics OpenMC simulations conducted in the FHR benchmark's Phase I-A. To run Moltres simulations, the user provides group constant data from a neutron transport solver, such as OpenMC, for the Moltres multigroup neutron diffusion calculations and a mesh file representing the reactor geometry. In the following subsections, I will describe the steps conducted to produce the AHTR temperature model with Moltres:

- OpenMC neutronics model produces group constants data for the Moltres model
- Moltres model mesh generation
- Run Moltres model to calculate temperature distribution in the system (Moltres model accepts group constants data and mesh)

3.4.1 Group Constant Generation

Unlike the OpenMC neutronics model from the previous sections, Moltres does not explicitly model each TRISO particle. Because I am using Moltres for temperature distributions and not neutron transport, a TRISO-level fidelity mesh file is impractical and results in extremely long Moltres simulation runtimes. Instead, Moltres relies on the OpenMC model to generate group constants data for the Moltres simulations. Previously, Moltres could only generate group constant data from Serpent [40] or SCALE [90] output files. I implemented functionality in Moltres for group constant data generation with OpenMC [89].

To enable successful Moltres AHTR temperature model simulations, I established suitable spatial and energy homogenization for group constant generation. These homogenizations preserve accuracy while maintaining an acceptable runtime. I used eight precursor groups and a four-group energy structure derived by Gentry et al. [39] for AHTR geometries. Table 3.4 defines the 4-group energy boundaries.

Table 3.4: 4-group energy structures for Advanced High-Temperature Reactor (AHTR) geometry derived by Gentry et al. [39].

Group Boundaries [MeV]		
Group #	Upper Bound	Lower Bound
1	2.0000×10^1	9.1188×10^{-3}
2	9.1188×10^{-3}	2.9023×10^{-5}
3	2.9023×10^{-5}	1.8554×10^{-6}
4	1.8554×10^{-6}	1.0000×10^{-12}

For spatial homogenization of the AHTR fuel assembly, I used OpenMC’s *cell* domain type to compute multigroup cross sections for different *cells*. I discretized the fuel assembly into 61 *cells*: inter-assembly FLiBe, Y-shaped graphite structure, control rod slot FLiBe, graphite spacers, each diamond shape section’s inter-plank FLiBe (3), each graphite plank (18), and each fuel stripe (36). I used reflective boundary conditions. Figure 3.12 illustrates the AHTR assembly’s spatial homogenization used for group constant generation.

With the above-described energy discretization and spatial homogenization, I generated group constants with Moltres using four OpenMC simulations at 948K, 1024K, 1100K, and 1200K. The OpenMC simulations are run with 80 active cycles, 20 inactive cycles, and 8000 particles resulting in an uncertainty of $\sim 150\text{pcm}$. In the next subsection, I compare the key neutronics parameters for the continuous OpenMC and multigroup Moltres simulations for the AHTR assembly at 948K to ensure that the spatial homogenization and energy discretization preserve accuracy.

Key Neutronics Parameters Verification

In this section, I verify that the spatial homogenization and energy discretization I chose are acceptable by verifying key neutronics parameters. I compare the key neutronics parameters between two simulations:

1. OpenMC simulation with continuous energy and TRISO-level spatial fidelity

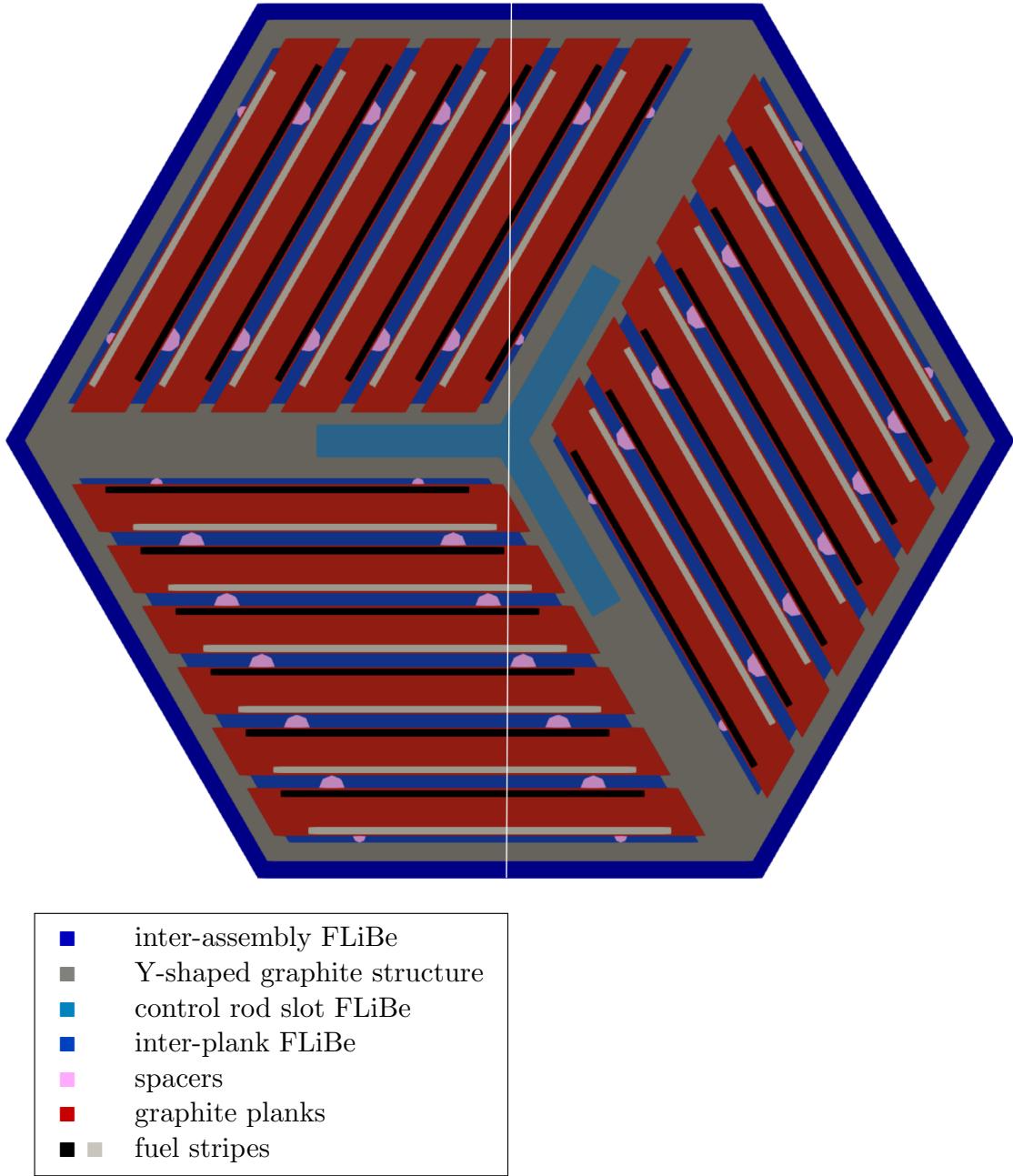


Figure 3.12: Advanced High-Temperature Reactor (AHTR) assembly spatially discretized into 61 *cells* for OpenMC multigroup calculation to produce group constants data for the Moltres model. 61 *cells*: inter-assembly FLiBe, Y-shaped graphite structure, control rod slot FLiBe, graphite spacers, each diamond shape section's inter-plank FLiBe (3), each graphite plank (18), and each fuel stripe (36). The white line corresponds to the centerline where flux distribution is measured.

2. Moltres simulation with 4-group energy and spatial homogenization

All materials' in the OpenMC simulation are 948K. The OpenMC simulation with TRISO-level fidelity generates the group constants for the energy and spatially homogenized Moltres simulation. I compare the following key neutronics parameters: effective multiplication factor, reactivity coefficients, flux distribution, and neutron energy spectrum. The comparisons are to verify that the Moltres model is replicating the OpenMC model's neutronics correctly. And of these, comparisons of the reactivity coefficients and flux distributions are key to ensuring that Moltres accurately calculates the AHTR's temperature distribution. The reactivity coefficients capture temperature reactivity feedback on the flux when the temperature varies with space in the Moltres model. The heat produced per fission, $\epsilon_{f,g}$, and macroscopic cross section for fission, $\Sigma_{f,g}$, terms in the Moltres source term (Equation 5.5) are provided to Moltres through the group constants generated by the transport software, OpenMC. Thus, differences in the source term between OpenMC and Moltres are dependent on the flux.

Effective Multiplication Factor Comparing k_{eff} and reactivity produced by Moltres and OpenMC verify that the Moltres model replicates the OpenMC model's neutronics correctly. Table 3.5 compares the effective multiplication factor (k_{eff}) and reactivity (ρ) for OpenMC simulation with continuous energy and TRISO-level spatial fidelity, OpenMC simulation with 4-group energy and spatial homogenization, and Moltres simulation with 4-group energy and spatial homogenization. I include results from the homogenized OpenMC simulation to distinguish between differences caused by spatial homogenization and energy discretization or differing OpenMC and Moltres solve methods.

Table 3.5: Advanced High-Temperature Reactor (AHTR) assembly's k_{eff} and reactivity values from the OpenMC simulation with continuous energy and TRISO-level spatial fidelity, OpenMC simulation with 4-group energy and spatial homogenization, and Moltres simulation with 4-group energy and spatial homogenization. All simulations are at 948K. The normalized difference is the pcm difference normalized by OpenMC non-homogenized model's k_{eff} .

Software	Homogenized?	k_{eff}	Diff [pcm]	Reactivity [pcm]	Reactivity Diff [pcm]
OpenMC	No	1.39850 ± 0.00126	-	28495 ± 64	-
OpenMC	Yes	1.398373 ± 0.00115	-13	28488 ± 58	-6
Moltres	Yes	1.40273	+423	28710	+216

The 13pcm k_{eff} and 6pcm reactivity difference, that can be observed in Table 5.6, between continuous and homogenized OpenMC simulations are within uncertainty. This shows that the selected spatial homogenizations and energy discretizations are acceptable. However, the Moltres simulation shows a 423pcm difference in k_{eff} and 216pcm difference in reactivity. The summary at the end of this section explains the differences in these values.

Reactivity Coefficients A comparison of reactivity coefficients produced by Moltres and OpenMC verifies that the Moltres model is replicating the OpenMC model's neutronics correctly and are also important to ensure that Moltres accurately calculates the temperature distribution. Moltres' delayed neutron fraction, β_{eff} , is calculated by taking the normalized difference between k_{eff} values with and without delayed neutron precursors (DNPs). Table 3.6 shows that the β_{eff} values from OpenMC and Moltres show excellent agreement with a discrepancy of 0.84pcm . I calculated the temperature reactivity coefficients using Equation 3.1. Temperature reactivity feedback arises mainly from Doppler broadening of resonance absorption peaks and thermal expansion. Table 3.6 also shows that the total temperature coefficients from OpenMC and Moltres have good agreement with a discrepancy of $0.43 \text{ pcm} \cdot K^{-1}$.

Table 3.6: Advanced High-Temperature Reactor (AHTR) assembly's β_{eff} values from OpenMC and Moltres simulations at 948K and total reactivity coefficient values calculated from OpenMC and Moltres at 948K and 1100K . The OpenMC simulation has continuous energy and TRISO-level spatial fidelity, and the Moltres simulation has 4-group energy and spatial homogenization.

Software	Homogenized?	β_{eff} [pcm]	Diff [pcm]	Total	$\frac{\Delta\rho}{\Delta T}$ [pcm $\cdot K^{-1}$]	Diff [pcm $\cdot K^{-1}$]
OpenMC	No	653.40	-	-3.63	-	-
Moltres	Yes	652.57	-0.84	-4.06	-	-0.43

Flux Distribution A comparison of flux distributions produced by Moltres and OpenMC verifies that the Moltres model is replicating the OpenMC model's neutronics correctly and are also important to ensure that Moltres accurately calculates the temperature distribution. Figure 3.13 shows the 4-group flux distributions for OpenMC and Moltres models' on the AHTR assembly's centerline, along the y-axis at the x-axis' midpoint (white line on Figure 3.12). Table 3.7 reports the 2-norm percentage difference (Equation 3.10) and maximum percentage difference between

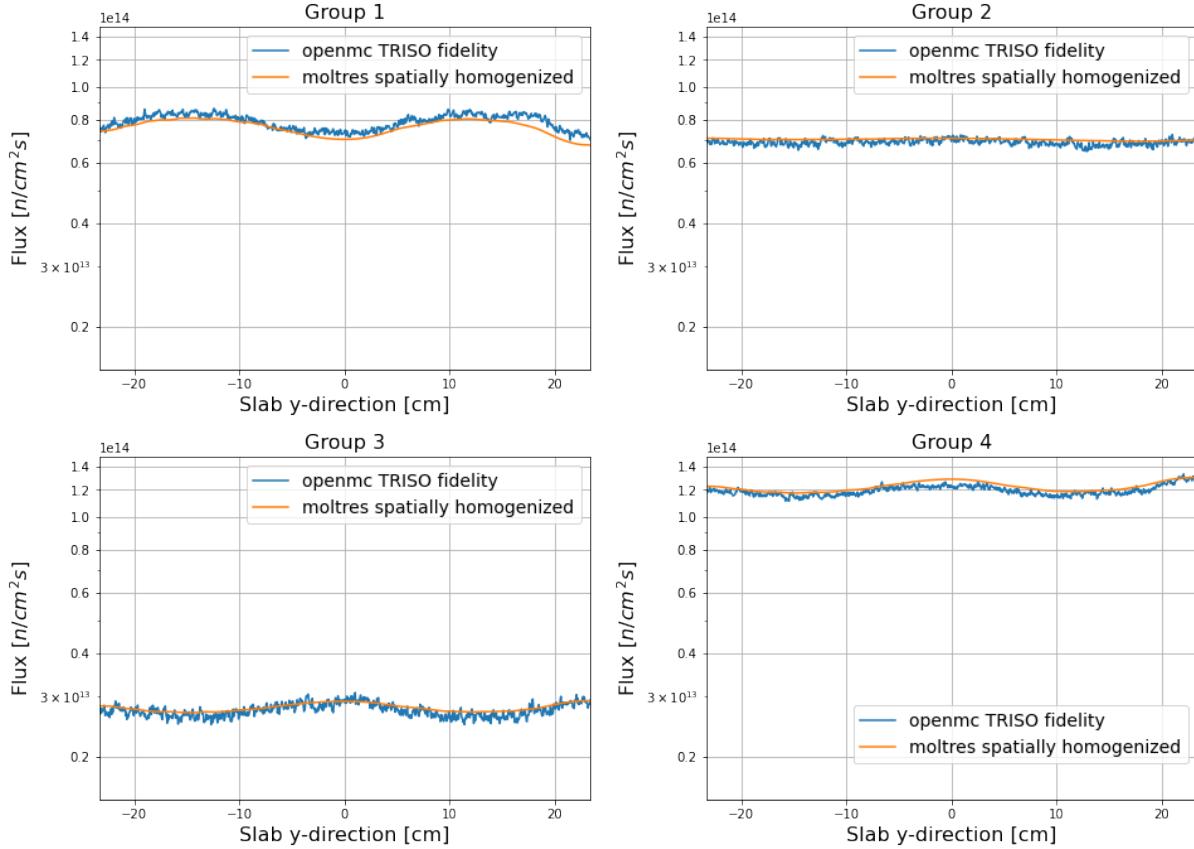


Figure 3.13: Advanced High-Temperature Reactor (AHTR) assembly’s centerline neutron flux distribution in 4 groups at 948K (discretized into 1000 points). Centerline is the white line in Figure 3.12. Comparison is between OpenMC simulation with continuous energy and TRISO-level spatial fidelity and Moltres simulation with 4-group energy and spatial homogenization. Energy Group 1: $E > 9.1188 \times 10^{-3}$ MeV, Energy Group 2: $2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$ MeV, Energy Group 3: $1.8556 \times 10^{-5} < E < 2.9023 \times 10^{-5}$ MeV, Energy Group 4: $1.0 \times 10^{-12} < E < 1.8554 \times 10^{-6}$ MeV.

centerline flux values from OpenMC and Moltres models.

$$\|\Delta\phi\|_N = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\frac{\phi_{moltres,i} - \phi_{openmc,i}}{\phi_{openmc,i}} \times 100 \right)^2} \quad (3.10)$$

where

$$\|\Delta\phi\|_N = \text{normalized 2-norm flux percentage difference between Moltres and OpenMC [\%]}$$

N = total number of discretized points

$$\phi_{moltres} = \text{Moltres model's centerline flux } \left[\frac{n}{cm^2 s} \right]$$

$$\phi_{openmc} = \text{OpenMC model's centerline flux } [\frac{n}{cm^2 s}]$$

Table 3.7: Advanced High-Temperature Reactor (AHTR) assembly’s centerline normalized 2-norm of flux percentage difference and maximum flux percentage difference. Centerline is the white line in Figure 3.12. The difference values are calculated from comparison between the OpenMC simulation with continuous energy and TRISO-level spatial fidelity and Moltres simulation with 4-group energy and spatial homogenization.

Energy Group	2-norm difference [%]	Max difference [%]
1	0.13	-10.57
2	0.08	+7.58
3	0.10	+8.96
4	0.09	+6.97

Figure 3.13 shows that the OpenMC model has good flux agreement for all groups. Table 5.8 shows that the 2-norm percentage differences between OpenMC and Moltres models’ flux values are slight, demonstrating a good overall agreement for each group’s flux. The summary at the end of this section explains the differences in the flux’s maximum percentage differences between the OpenMC and Moltres models.

Neutron Energy Spectrum Figure 3.14 shows the neutron spectrum of the OpenMC simulation for both 252 and 4 groups and the 4-group Moltres simulation. There is good agreement between OpenMC and Moltres models 4-group spectrums.

Key Neutronics Parameters Verification Summary

The verification study found a 216 pcm reactivity difference between OpenMC and Moltres models and good agreement in their reactivity coefficients and 4-group neutron energy spectrum. The verification study also found good agreement in overall flux distributions (based on the 2-norm difference); however, there were larger flux differences at specific points. The reactivity and flux differences are due to Moltres utilizing the neutron diffusion method instead of neutron transport methods, resulting in flux not being reproduced well in small regions which are less than a few mean free paths in length (i.e., the FLiBe coolant channel has 0.7cm which is smaller than the diffusion coefficient). The differences in reactivity and flux at specific points might result in slightly

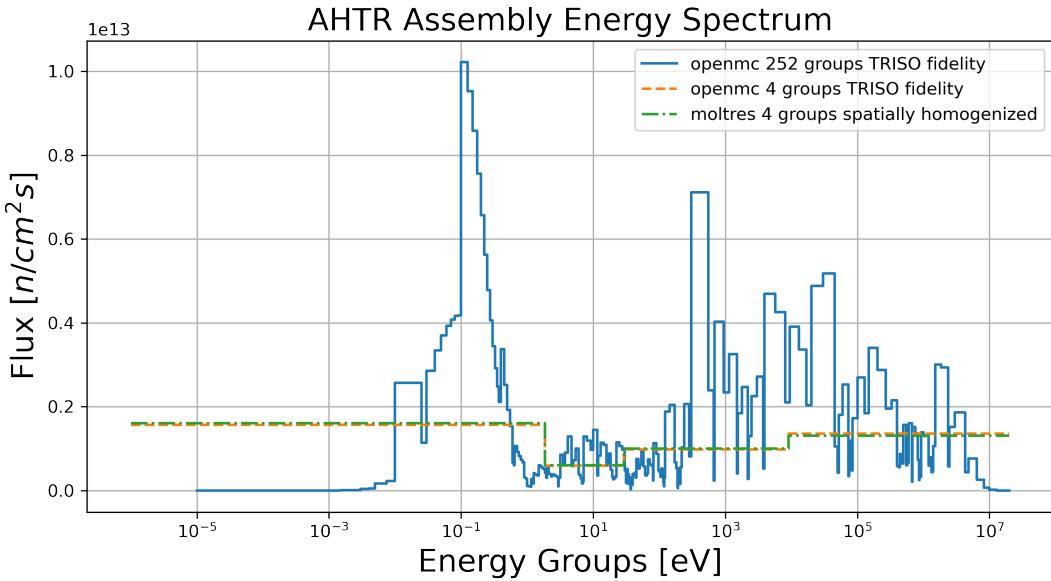


Figure 3.14: Advanced High-Temperature Reactor (AHTR) full assembly’s neutron spectrum. Spectrums include 252 and 4 group spectrums from OpenMC simulation with continuous energy and TRISO-level spatial fidelity and 4 group spectrum from Moltres simulation with 4-group energy and spatial homogenization.

inaccurate temperatures at certain points. However, since the reactivity coefficients and overall flux distribution are in agreement, OpenMC’s group constants are sufficient to use to calculate and gain an overall perspective of the temperature distribution in the AHTR full assembly model. Future work includes using higher fidelity software to model AHTR full assembly’s temperature distribution.

3.4.2 Moltres Mesh Generation

To run Moltres simulations, the user must provide a mesh file representing the reactor geometry. I created an AHTR assembly geometry file (`.geo`) based on the spatial homogenization described in Section 3.4.1. The AHTR mesh is then generated from the geometry file using Gmsh [91]. I used Gmsh’s *refine by splitting* functionality to refine the mesh. Figure 3.15 shows the AHTR assembly’s Gmsh rendered geometry file.

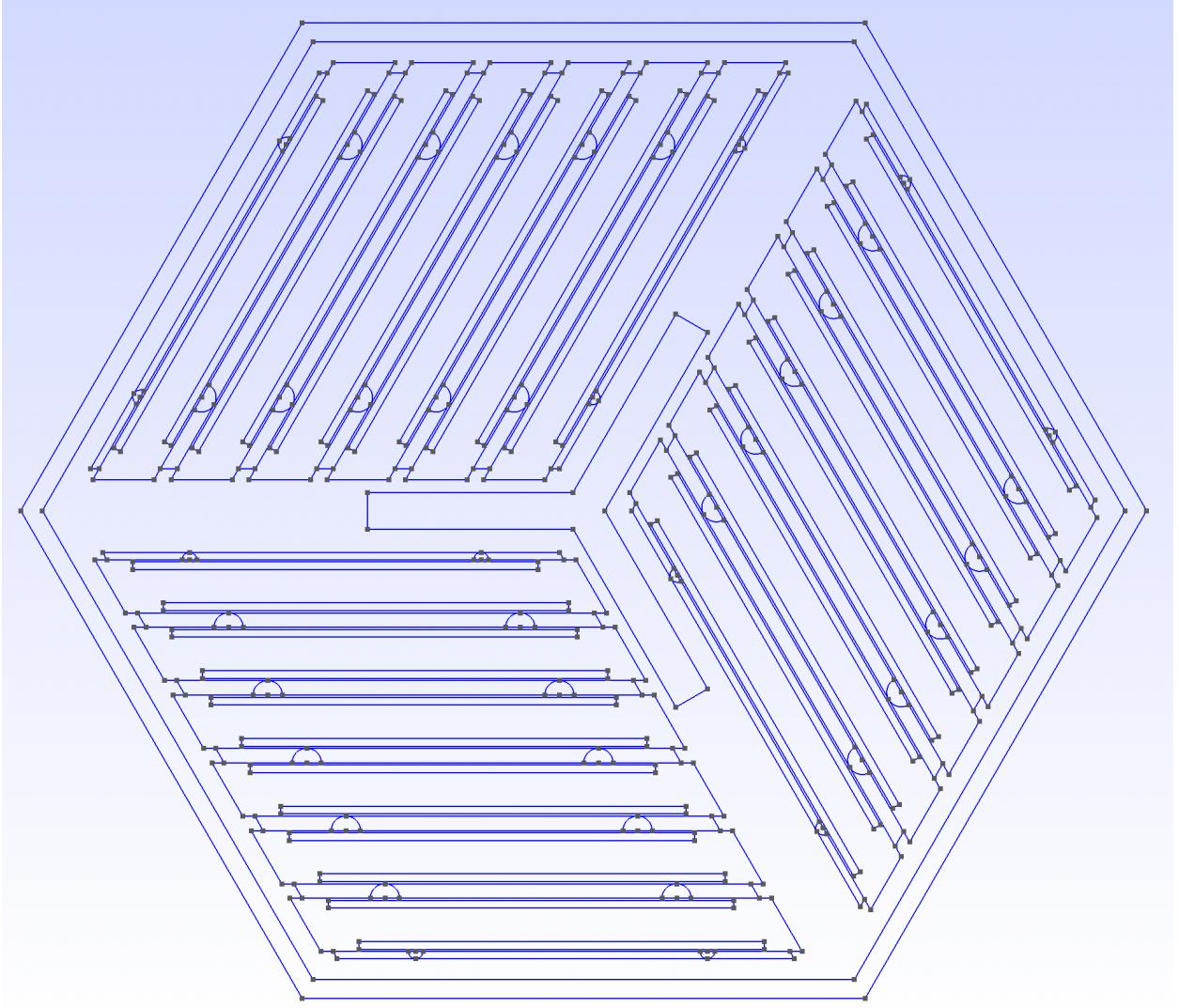


Figure 3.15: Advanced High-Temperature Reactor (AHTR) assembly’s Gmsh rendered geometry file. I meshed the geometry file using Gmsh [91]. The mesh file is used in the AHTR assembly Moltres temperature model.

Mesh Refinement Study

I performed a mesh refinement study on the Moltres AHTR full assembly temperature model to ensure that the geometry mesh inputs are sufficiently converged. Table 3.8 reports the average, maximum, and normalized 2-norm of discretized temperature difference between refinement steps (Equation 3.11):

$$\|\Delta T_k\|_N = \frac{1}{N} \sqrt{\sum_{i=1}^N (T_{k-1,i} - T_{k,i})^2} \quad (3.11)$$

where

$\|\Delta T_k\|_N$ = normalized 2-norm discretized temperature difference between refinement steps [K]

N = total number of discretized points

k = refinement step

T_{k-1} = temperatures from previous refinement step [K]

T = temperatures from current refinement step [K]

Table 3.8: Advanced High-Temperature Reactor (AHTR) full assembly's Moltres temperature model's mesh refinement study.

Refinement	Max Assembly Temp [K]	Diff [K]	Ave Assembly Temp [K]	Diff [K]	$\ \Delta T_k\ _N$ [K]
1	1040.931	-	960.619	-	-
2	1048.941	+8.010	962.854	+2.235	0.175
3	1057.403	+8.463	963.948	+1.093	0.067
4	1061.599	+4.196	965.020	+1.073	0.053

The mesh refinement study shows $\|\Delta T_k\|_N$ convergence with more refinement steps. I used x4 mesh refinement for the Moltres AHTR full assembly temperature model because $\|\Delta T_k\|_N = 0.053K$, which I evaluated as sufficiently converged.

3.4.3 Temperature Model Setup and Results

The Moltres AHTR steady-state temperature model is a 2D x-y cross-section model of the AHTR assembly. The Moltres input file is set up using the assumptions and constants described in the previous subsection. The Moltres temperature model first solves the AHTR neutronics and uses that to solve the AHTR's temperature distribution for a defined power. The temperature model assumes conductive heat transfer throughout the domain and heat removal by uniform salt flow in the coolant regions. These assumptions ignore turbulent effects that would most likely be present. However, an in-depth AHTR Moltres model that includes turbulence modeling is out of scope for this dissertation.

Temperature Model Setup

Moltres solves the four-group diffusion equations (Equation 2.1) as a steady-state eigenvalue problem to find k_{eff} for the static AHTR models. In the 2D cross-sectional AHTR steady-state temperature model, I ignore the time-dependent and velocity-dependent terms from Moltres' temperature governing equation (Equation 2.2) as I am running a steady-state model with no moving fuel, resulting in Equation 3.12:

$$-\nabla \cdot (k_i \nabla T) = Q_i \quad (3.12)$$

where

k_i = thermal conductivity of material i

T = temperature in the model

Q_i = source or sink term in material i

I use insulated temperature boundary conditions. Table 3.9 shows the thermal conductivity values used for each AHTR material. Equation 3.13 defines the fuel cells' fission source term (Q_f).

Table 3.9: Advanced High-Temperature Reactor (AHTR) materials' thermal conductivities used in Moltres temperature models, taken from [92].

Material	Thermal Conductivity [W cm ⁻¹ K ⁻¹]
FLiBe	0.01
Graphite	0.15
Fuel	0.099

$$Q_f = \sum_{g=1}^G \epsilon_{f,g} \Sigma_{f,g} \phi_g \quad (3.13)$$

where

$$Q_f = \text{source term } \left[\frac{\text{MeV}}{\text{cm}^3 \text{s}} \right]$$

G = number of discrete groups, g [-]

$\epsilon_{f,g}$ = heat produced per fission [MeV]

$\Sigma_{f,g}$ = macroscopic cross section for fission due to neutrons in group g [$\frac{1}{cm}$]

ϕ_g = flux of neutrons in group g [$\frac{n}{cm^2 s}$]

Equation 3.14 defines the heat removal from the AHTR model.

$$Q = h \cdot A \cdot (T(\vec{r}) - T_{ref}) \quad (3.14)$$

where

Q = heat removal rate for 1cm thin slice of the AHTR model [W/cm]

h = heat transfer coefficient [$\frac{W}{cm^3 \cdot K}$]

A = coolant area [cm^2]

$T(\vec{r})$ = temperature at point \vec{r} [K]

T_{ref} = reference temperature [K]

Table 3.10 shows reference temperature and heat transfer coefficient values for the convective heat transfer process.

Table 3.10: Advanced High-Temperature Reactor (AHTR) assembly heat transfer constants for Moltres temperature model.

Constant	Value	Units	Notes
h_{assem}	611	$\frac{W}{cm \cdot K}$	Calculated in Eq. 3.15
T_{ref}	923	K	AHTR Inlet Temperature [92]

I calculated the heat transfer coefficient (h) for the AHTR assembly using Equation 3.15 with the following assumptions: AHTR models generate constant amount of power, which is all removed by the coolant, and that there is linear increase in temperature from inlet to outlet.

$$h_{assem} = \frac{P_{dz}}{V_{coolant} \cdot \Delta T} \quad (3.15)$$

$$\begin{aligned}
&= \frac{26223W/cm}{471cm^3 \cdot 0.0909K \cot cm^{-1}} \\
&= 611Wcm^{-3}K^{-1}
\end{aligned}$$

and

$$\Delta T = \frac{T_{total}}{H} = \frac{50K}{550cm} = 0.0909K/cm \quad (3.16)$$

where

h_{assem} = assembly's heat transfer coefficient [$Wcm^{-3}K^{-1}$]

P_{dz} = power produced in 1cm AHTR assembly Δz slice [$W \cdot cm^{-1}$]

$V_{coolant}$ = coolant volume in AHTR assembly [cm^3]

ΔT = temperature change across 1cm AHTR assembly Δz slice [K] (3.17)

T_{total} = total temperature change from inlet to outlet [K]

H = AHTR height from inlet to outlet [cm]

The power produced by the AHTR assembly model is calculated based on the FHR benchmark model's specific power of $200 WgU^{-1}$ and the FHR benchmark's TRISO packing fractions in a 1cm thick slice of the assembly.

Temperature Model Results

Figure 3.16 shows the 2D temperature distribution in the AHTR assembly. The average temperature distribution across the fuel planks are similar at approximately 1025K, while the graphite structure has an average temperature of approximately 935K. As defined in the temperature model setup, the FLiBe coolant channel (inter-plank, inter-assembly, control rod slot) regions remain at 923K. The temperature peaks at 1062K, which occur in the fuel stripes near the spacers. This could be due to the extra moderation provided by the graphite spacers. This highlights the significance of the spacers' material composition and location in causing higher temperature peaks.

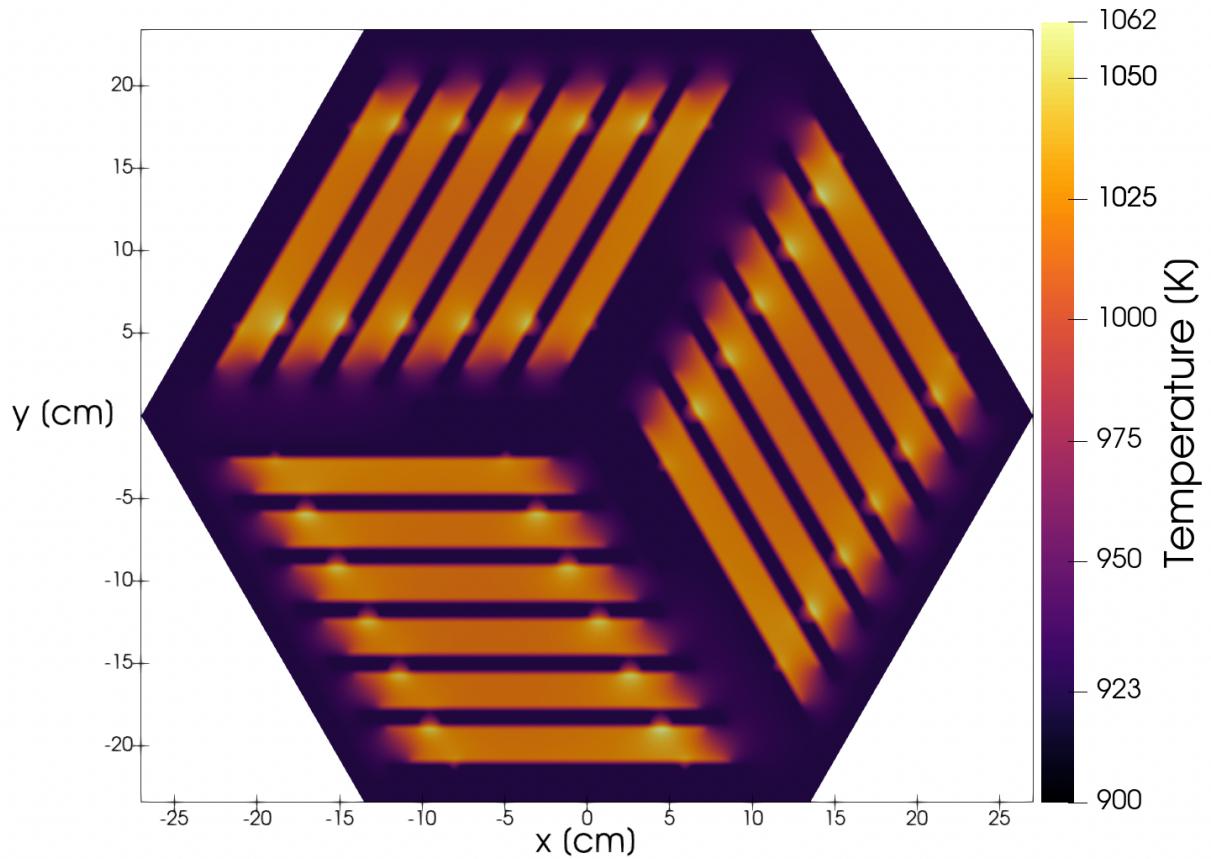


Figure 3.16: 2D temperature distribution in the Advanced High-Temperature Reactor (AHTR) full assembly generated by Moltres.

3.5 Summary

This chapter described the Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark specifications, Advanced High-Temperature Reactor (AHTR) design, Phase I-A and I-B results obtained by the University of Illinois at Urbana-Champaign (UIUC) team, and the FHR temperature model's results. The benchmark results highlight the AHTR's passive safety behavior with negative temperature coefficients. Results such as a lower k_{eff} for the AHTR configuration with higher heavy metal loading demonstrated that increased fuel packing does not always correspond with increased k_{eff} due to self-shielding effects. These results hint at the possibility of minimizing fuel required by optimizing for heterogenous fuel distributions within the core. This will be further explored in the later chapters. The temperature model demonstrates that the AHTR's temperature peaks are in the fuel stripes near the spacers, highlighting to reactor designers that spacer material and location in the AHTR geometry impact temperature peaks.

Chapter 4

ROLLO: Reactor evOLutionary aLgorithm Optimizer

In this chapter, I introduce the Reactor evOLutionary aLgorithm Optimizer (ROLLO) framework developed for this dissertation. ROLLO is a Python package that applies evolutionary algorithm techniques to optimize nuclear reactor design. Applying evolutionary algorithms to nuclear design problems is not new, as discussed in Section 2.3. Reactor designers have individually customized available evolutionary algorithm packages for their reactor design optimization problems. However, the evolutionary algorithm setup is highly customizable with an assortment of genetic algorithm designs and related operators. A well-performing evolutionary algorithm needs to balance the extent of exploration and exploitation. Thus, a reactor designer unfamiliar with evolutionary algorithms will have to go through the cumbersome process of customizing a genetic algorithm for their needs and determine which operators and hyperparameters work best for their problem. Furthermore, computing fitness values with nuclear software is computationally expensive, necessitating using supercomputers. Reactor designers have to set up parallelization to use the genetic algorithm in concert with nuclear software.

Therefore, the motivation for ROLLO is to limit these inconveniences and facilitate using evolutionary algorithms for reactor design optimization. ROLLO provides a general genetic algorithm framework, sets up parallelization for the user, and promotes usability with an input file that only exposes mandatory parameters. ROLLO strives to be effective, flexible, open-source, parallel, reproducible, and usable. I briefly summarize how ROLLO achieves these goals:

- Effective: ROLLO is well documented and tested.
- Flexible: This dissertation uses ROLLO to explore arbitrary reactor geometries and heterogeneous fuel distributions. However, future users might want to utilize ROLLO to explore other arbitrary design parameters. Thus, I designed the ROLLO framework accordingly. The

user can vary any imaginable parameter because ROLLO uses a templating method to edit the input file of the coupled software.

- Open-source: ROLLO is open-source and version-controlled on Github [12], benefitting from the innovation of the open-source community. ROLLO also utilizes the well-documented, open-source Distributed Evolutionary Algorithms in Python (DEAP) [93] Python package to drive the evolutionary algorithm optimization process.
- Parallel: Users have the option to run ROLLO in parallel so that they may effectively use the computational resources available.
- Reproducible: Data from every ROLLO run saves into a unique, pickled checkpoint file (pickle is a Python module that serializes Python objects). The checkpoint file contains the ROLLO input file and coupled evaluator's scripts enabling results replication. The checkpoint file also acts as a restart file for partially completed simulations.

ROLLO provides a framework to couple an evolutionary algorithm driver with nuclear software, such as neutron transport and thermal-hydraulics codes. ROLLO is nuclear code-agnostic and does not have any hard dependencies on any nuclear software. Figure 2.7 from Chapter 2 outlined a general evolutionary algorithm iterative problem solving process. I modified Figure 2.7 to produce Figure 4.1, which depicts how the nuclear transport and thermal-hydraulics software fit within ROLLO's evolutionary algorithm optimization process. ROLLO initially reads and validates the JSON input file, initializes the DEAP [93] genetic algorithm hyperparameters and operators, and finally runs the genetic algorithm following the flow chart in Figure 4.1, in which the nuclear software evaluates each individual reactor model's fitness.

The subsequent sections describe ROLLO's evolutionary algorithm driver software DEAP (Section 4.1), input file structure (Section 4.2), software architecture (Section 4.3), verification studies (Section 4.4), and convergence criteria (Section 4.5.1). Appendix A lists all the data and analysis related to this chapter to enable the reproduction of all the simulations.

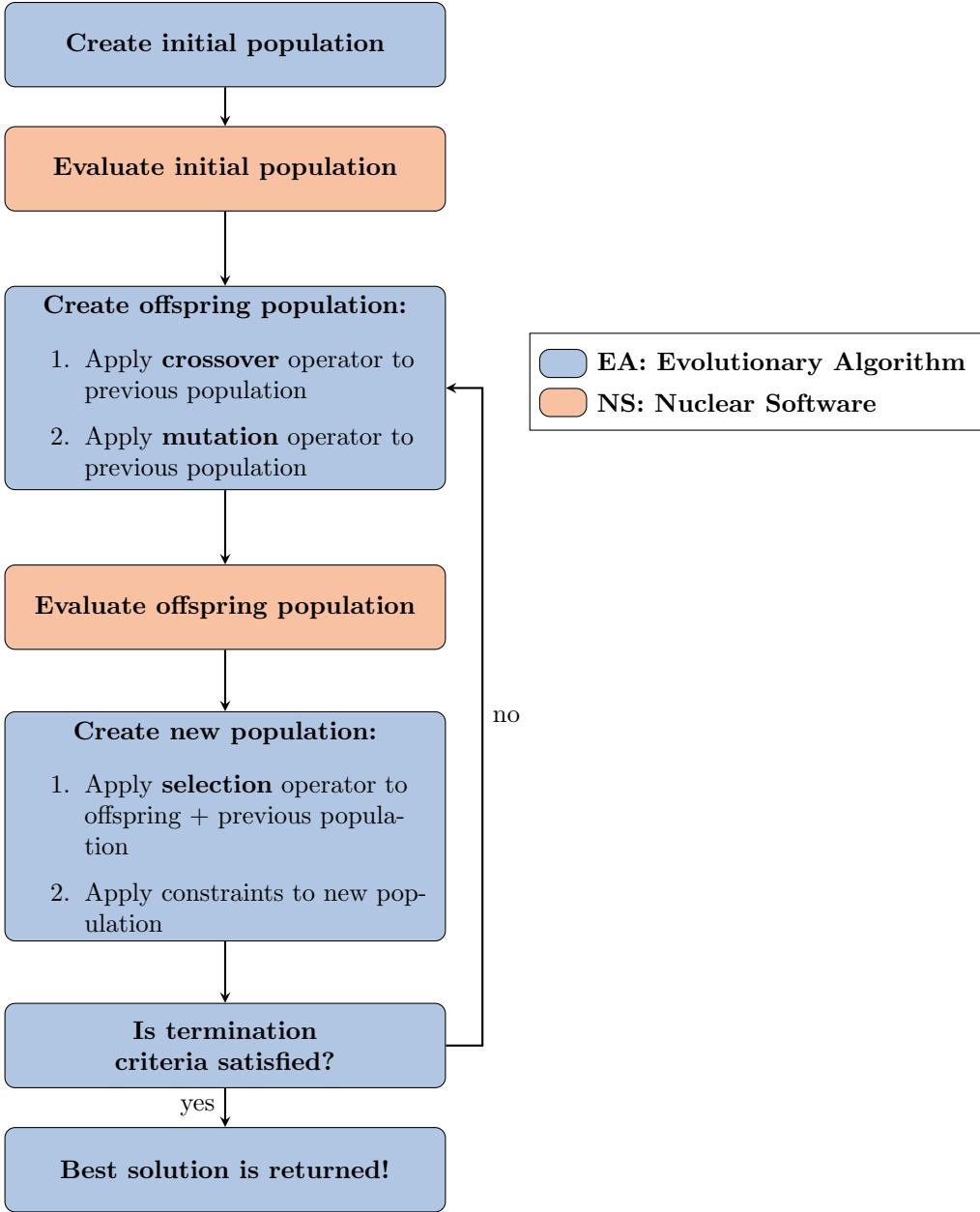


Figure 4.1: Process of finding optimal solutions for a problem with a genetic algorithm. Nuclear software evaluates each new population.

4.1 Evolutionary Algorithm Driver

Evolutionary algorithm computation uses sophisticated, diverse techniques and mechanisms, resulting in even the most well-designed software frameworks being complicated under the hood. Utilizing an existing evolutionary algorithm framework presents implementation challenges as the user must edit the framework's source code to customize it for their application and hyperparam-

eters [93]. Therefore, a computation framework that gives the user the capability to build custom evolutionary algorithms is ideal for this project.

Many evolutionary algorithm computation packages exist: DEAP [93], Pymoo [94], inspyred [95], Pyevolve [96], and OpenBEAGLE [97]. DEAP is the newest package and places a high value on code compactness and clarity [93]. DEAP is the only framework that allows the user to prototype evolutionary algorithms rapidly and define custom algorithms without digging deep into the source code to modify hyperparameters and their application methods. Accordingly, I chose DEAP to drive the ROLLO framework’s evolutionary algorithm component. DEAP provides building blocks for each optimizer function and allows the user to customize a specialized algorithm to fit their project [93].

4.1.1 Distributed Evolutionary Algorithms in Python

DEAP is composed of two structures: a *creator* and a *toolbox*. The *creator* module allows the run-time creation of classes via inheritance and composition, enabling individual and population creation from any data structure: lists, sets, dictionaries, trees, etc [93]. The *toolbox* is a container that the user manually populates. In the *toolbox*, the user defines the selection, crossover, and mutation operator types and their respective hyperparameters. For example, the user registers a crossover operator under the ‘mate’ alias and a selection operator under the ‘select’ alias. Then, the evolutionary algorithm uses these aliased operators from the *toolbox*. If the user wants to change the crossover operator, they update the ‘mate’ alias in the *toolbox* while keeping the evolutionary algorithm unchanged [93].

Figure 4.2 illustrates DEAP’s usage of the *creator* and *toolbox* modules. Line 2 creates a single-objective fitness class, `Objective`. The first argument defines the derived class’s name; the second argument specifies the inherited base class, `base.fitness`; the third argument indicates the objective fitness (-1.0 indicates a minimum objective, +1.0 indicates a maximum objective). Line 3 derives an `Individual` class from the standard Python list type and defines its fitness attribute as the newly created `Objective` object. Lines 5-9 initialize the DEAP toolbox, register `variable_1` and `variable_2` with their upper and lower bounds, and defines the `individual_creator` function to return an `Individual` initialized with `variable_1` and `variable_2`. Lines 10-11 and 14-17 are

```

1   from deap import creator, base, tools, algorithms
2   creator.create("Objective", base.Fitness, weights=(-1.0,)) # minimum
3   creator.create("Individual", list, fitness=creator.Objective)
4
5   toolbox = base.Toolbox()
6   toolbox.register("variable_1", random.uniform, 0.0, 10.0)
7   toolbox.register("variable_2", random.uniform, -1.0, 0.0)
8   def individual_creator():
9       return creator.Individual([toolbox.variable_1(), toolbox.variable_2()])
10  toolbox.register("individual", individual_creator())
11  toolbox.register("population", tools.initRepeat, list, toolbox.individual)
12  def evaluator_fn(individual):
13      return tuple([sum(individual)])
14  toolbox.register("evaluate", evaluator_fn)
15  toolbox.register("select", tools.selBest, k=5)
16  toolbox.register("mutate", tools.mutPolynomialBounded, eta=0.5, low=[0, -1], up=[-1, 0])
17  toolbox.register("mate", tools.cxOnePoint)

```

Figure 4.2: DEAP sample code demonstrating the usage of the *creator* and *toolbox* modules to initialize the genetic algorithm. In ROLLO, DEAP’s *creator* and *toolbox* modules are initialized in the source code based on the genetic algorithm parameters defined by the user in the ROLLO input file.

aliases for initializing individuals and populations, specifying variation operators (`select`, `mutate`, `mate`), and evaluating individual fitness (`evaluate`) [93]. Lines 12-13 define the evaluation function that returns the fitness values.

ROLLO initializes DEAP’s *creator* and *toolbox* modules based on the genetic algorithm parameters defined by the user in the ROLLO input file. The evaluation function runs the nuclear software and returns user-defined fitness values.

4.1.2 ROLLO Evolutionary Algorithm Implementation

DEAP creators provided variations of a classical genetic algorithm exposing different explicitness levels [93]. The high-level examples use the built-in DEAP genetic algorithms, whereas the low-level example completely unpacks the genetic algorithm to expose a generational loop. I included an unpacked genetic algorithm in ROLLO’s ***Algorithm*** class, depicted in Figure 4.1. The algorithm begins by initializing the starting population and evaluating each individual’s fitness value. Then, it enters a generational loop. During each iteration, the mating and mutation operators are applied

to the population to create an offspring population, and all offspring individuals are evaluated. The selection operator is then applied to both the previous and offspring populations to select the best individuals to create the new population. Finally, the constraints are applied, and the results are saved. Applying the selection operator to the combined previous and offspring populations expands the population, ensuring the effectiveness of elitism selection operators, such as NSGA-II, which work well for multi-objective optimization.

4.2 ROLLO Input File

ROLLO's input file is in JSON format. There are four sections that the user must define: `control_variables`, `evaluators`, `constraints`, and `algorithm`. Figure 4.3 shows an example ROLLO input file. In this example, ROLLO uses a genetic algorithm with user-defined hyper-parameters, defined in the `algorithm` section, to minimize the `output1` parameter. The OpenMC evaluator accepts input parameters: `variable1` and `variable2`, and calculates the `output1` parameter. Next, I describe how to define each section of a ROLLO input file. The ROLLO documentation [98] provides further descriptions for setting up a ROLLO input file.

4.2.1 Control Variables

I use the term control variables to refer to parameters the genetic algorithm will vary (conventionally, a control variable is held constant in a research study however this is not the case for ROLLO). The user must specify the minimum and maximum values for each control variable. For example, Lines 2 to 5 in Figure 4.3 demonstrate that the control variables, `variable1` and `variable2`, will be varied from 0 to 10 and -1 to 0, respectively. For example, in traditional reactor design, a control variable might be fuel enrichment.

4.2.2 Evaluators

Evaluators are the nuclear software ROLLO utilizes to calculate objective functions. ROLLO is nuclear code-agnostic and does not have hard dependencies on any nuclear software. Thus, it is up to the user to ensure that the nuclear software and their corresponding executables are correctly installed. In a single ROLLO input file, users may define any number of evaluators.

```

1      {
2          "control_variables": {
3              "variable1": {"min": 0.0, "max": 10.0},
4              "variable2": {"min": -1.0, "max": 0.0}
5          },
6          "evaluators": {
7              "openmc": {
8                  "order": 0,
9                  "inputs": ["variable1", "variable2"],
10                 "input_script": ["python", "openmc_inp.py"],
11                 "execute": [{"openmc"}],
12                 "outputs": ["output1", "output2"]
13                 "output_script": ["python", "openmc_output.py"],
14             }
15         },
16         "constraints": {
17             "output1": {"operator": [">=", "<"], "constrained_val": [1.0, 1.5]}
18         },
19         "algorithm": {
20             "objective": ["min"],
21             "weight": [1.0],
22             "optimized_variable": ["output1"],
23             "pop_size": 100,
24             "generations": 10,
25             "parallel": "job_control",
26             "keep_files": "all",
27             "mutation_probability": 0.23,
28             "mating_probability": 0.46,
29             "selection_operator": {"operator": "selTournament", "tournsize": 5},
30             "mutation_operator": {
31                 "operator": "mutPolynomialBounded",
32                 "indpb": 0.23,
33                 "eta": 0.23
34             },
35             "mating_operator": {"operator": "cxBlend", "alpha": 0.46}
36         }
37     }

```

Figure 4.3: Reactor evOLutionary aLgorithm Optimizer (ROLLO) sample JSON input file.

For each evaluator, mandatory parameters are `order`, `input_script`, `inputs`, and `outputs`, and the optional parameters are `execute` and `output_script`. Table 4.1 describes each evaluator's mandatory and optional parameters in the ROLLO evaluators' input file section.

ROLLO utilizes jinja2 templating [99] to insert the control variable values into the `input_script`.

Table 4.1: Reactor evOLutionary aLgorithm Optimizer (ROLLO) **evaluator**: Input Parameter Descriptions.

	Parameter	Type	Description
Mandatory Parameters	order	int	evaluator's operational order compared to other evaluators (indexed by 0)
	inputs	list of strings	control variables to be placed in the input script template
	input_script	list of strings (2-element)	1st element: executable to run input script, 2nd element: input script template
	outputs	list of strings	output variables that the evaluator will return to the genetic algorithm
Optional Parameters	execute	list of 2-element lists	enables users to run other executables or files beyond the input and output scripts. 1st element: executable to run file, 2nd element: file to run
	output_script	list of strings (2-element)	1st element: executable to run output script, 2nd element: output script template

Users must include each evaluator's input file template in the same directory as the ROLLO input file. Users must also ensure the template variables correspond to the **inputs** defined in the corresponding evaluator's section in the ROLLO input file. Lines 6 to 13 in the ROLLO input file (Figure 4.3) demonstrate that **variable1** and **variable2** are **inputs** into the `openmc_inp.py` **input_script**. Figure 4.4 shows the template and templated openmc script; once the `openmc_inp.py` **input_script** is templated, `{{variable1}}` and `{{variable2}}` on Lines 3 and 4 will be replaced with values selected by the ROLLO genetic algorithm.

<pre> 1 import openmc 2 # templating 3 variable1 = {{variable1}} 4 variable2 = {{variable2}} 5 # run openmc 6 ... </pre>	<pre> 1 import openmc 2 # templating 3 variable1 = 3.212 4 variable2 = -0.765 5 # run openmc 6 ... </pre>
--	---

Figure 4.4: `openmc_inp.py` input script template (left). Templated `openmc_inp.py` with **variable1** and **variable2** values defined (right).

ROLLO uses two methods to return an output variable to the genetic algorithm. First, ROLLO will automatically return the input parameter's value if the output parameter is also an input parameter. Second, the user may include an **output_script** that returns the desired output parameter. The **output_script** must include a line that prints a dictionary containing the output

parameters' names and their corresponding value as key-value pairs.

4.2.3 Constraints

In the constraints section, the user can define constraints on any output parameter. Any individual that does not meet the defined constraints is removed from the population, encouraging the proliferation of individuals that meet the constraints. For each constrained output parameter, the user lists the `operators` and `constrained_vals` as in Line 17 of the ROLLO input file (Figure 4.3). Thus, for this ROLLO simulation, `output_1` is constrained to be ≥ 1.0 and < 1.5 . For example, for reactor safety, a constraint might be the maximum temperature in the core.

4.2.4 Algorithm

In the algorithm section, users define the simulation's general settings and the genetic algorithm's hyperparameters. The mandatory input parameters include `optimized_variable`, `objective`, `pop_size`, and `generations`. The optional input parameters include `parallel`, `keep_files`, `mutation_probability`, `mating_probability`, `selection_operator`, `mutation_operator`, and `mating_operator`. Lines 17 to 31 in the example ROLLO input file (Figure 4.3) demonstrate `algorithm` specifications. Table 4.2 describes these input parameters in detail. The hyperparameter default values are selected based on hyperparameter studies conducted for a single objective FHR optimization. Section 5.5 describes the hyperparameter studies.

As mentioned previously in Section 2.4.1, it is important to select genetic algorithm hyperparameters that balance the extent of exploration and exploitation. For each operator, users choose from a list of operators and define each of their required hyperparameters. Table 4.3 shows the available operators and their respective hyperparameters. Descriptions for each operator can be found in Section 2.4.1.

Table 4.2: Reactor evOLutionary aLgorithm Optimizer (ROLLO) algorithm: Input Parameter Descriptions.

	Parameter	Type	Description	Default
Mandatory Parameters	<code>optimized_variable</code>	list of strings	variables to be optimized	-
	<code>objective</code>	list of strings	options include: min or max, each objective corresponds to a variable in <code>optimized_variable</code>	-
	<code>pop_size</code>	int	population size	-
	<code>generations</code>	int	number of generations	-
Optional Parameters	<code>parallel</code>	string	options include: <code>none</code> , <code>multiprocessing</code> , <code>job_control</code>	<code>none</code>
	<code>keep_files</code>	string	options include: <code>none</code> , <code>only_final</code> , <code>all</code>	<code>none</code>
	<code>mutation_probability</code>	float	mutation probability	0.23
	<code>mating_probability</code>	float	mating probability	0.47
	<code>selection_operator</code>	dict	see Table 4.3	<code>"operator": "selTournament", "tournsize": 5</code>
	<code>mutation_operator</code>	dict	see Table 4.3	<code>"operator": "mutPolynomialBounded", "eta": 0.23, "indpb": 0.23</code>
	<code>mating_operator</code>	dict	see Table 4.3	<code>"operator": "cxBlend", "alpha": 0.46</code>

Table 4.3: Selection, mutation, and mating operators available in Reactor evOLutionary aLgorithm Optimizer (ROLLO) and their corresponding hyperparameters. n/a indicates that the operator does not have associated hyperparameters.

Operator	Available Options	Hyperparameters
Selection	<code>selTournament</code>	<code>tournsize</code> : no. of individuals in each tournament
	<code>selNSGA2</code>	n/a
	<code>selBest</code>	n/a
Mutation	<code>mutPolynomialBounded</code>	<code>eta</code> : crowding degree of the mutation <code>indpb</code> : independent probability for each attribute to be mutated
Mating	<code>cxOnePoint</code>	n/a
	<code>cxUniform</code>	<code>indpb</code> : independent probability for each attribute to be exchanged
	<code>cxBinaria</code>	<code>alpha</code> : Extent of the interval that the new values can be drawn for each attribute on both sides of the parents' attributes

4.3 ROLLO Software Architecture

This section describes the ROLLO v1.0 software architecture and how each part contributes the reactor design optimization process. Table 4.4 outlines the classes in the ROLLO software and describes each class's purpose. Figure 4.5 depicts the ROLLO software architecture. When the

Table 4.4: Classes that comprise the ROLLO architecture.

Class	Description
<i>InputValidation</i>	The <i>InputValidation</i> class contains methods to read and validate the JSON ROLLO input file to ensure the user defined all key parameters. If they did not, ROLLO raises an exception to tell the user which parameters are missing.
<i>Evaluation</i>	DEAP's fitness evaluator (as mentioned in Section 4.1.1) requires an evaluation function to evaluate each individual's fitness values. The <i>Evaluation</i> class contains a method that creates an evaluation function that runs the nuclear software and returns the required fitness values defined in the input file.
<i>ToolboxGenerator</i>	The <i>ToolboxGenerator</i> class initializes DEAP's <i>toolbox</i> and <i>creator</i> modules with genetic algorithm hyperparameters defined in the input file.
<i>Constraints</i>	The <i>Constraints</i> class contains methods to initialize constraints defined in the input file and applies the constraints by removing individuals that do not meet the constraint.
<i>BackEnd</i>	The <i>BackEnd</i> class contains methods to save genetic algorithm population results into a pickled checkpoint file and to restart a partially completed genetic algorithm from the checkpoint file.
<i>Algorithm</i>	The <i>Algorithm</i> class contains methods to initialize and execute the genetic algorithm. It executes a general genetic algorithm framework that uses the hyperparameters defined in the <i>ToolboxGenerator</i> , applies constraints defined in <i>Constraints</i> , evaluates fitness values using the evaluation function produced by <i>Evaluation</i> , and saves all the results with <i>BackEnd</i> .
<i>Executor</i>	The <i>Executor</i> class drives the ROLLO code execution with the following steps: 1) User input file validation with <i>InputValidation</i> 2) Evaluation function generation with <i>Evaluation</i> 3) DEAP toolbox initialization with <i>ToolboxGenerator</i> 4) Constraint initialization with <i>Constraints</i> 5) Genetic algorithm execution with <i>Algorithm</i>

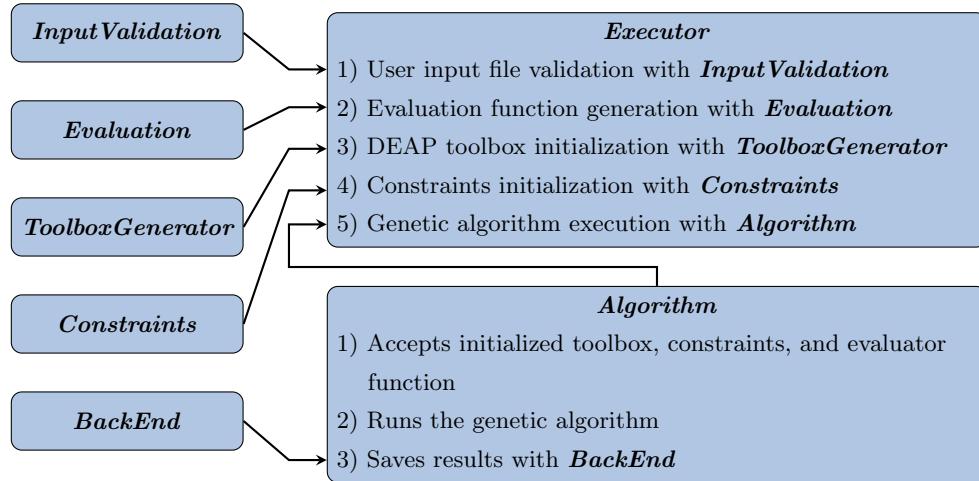


Figure 4.5: Visualization of ROLLO architecture.

user runs a ROLLO input file, the ***Executor*** class drives ROLLO’s execution from beginning to end. The ***Executor*** calls ***InputValidation*** to parse the input file to ensure that the user defined all mandatory parameters and used the correct formatting. Next, it initializes an ***Evaluation*** object based on the **evaluators** specifications in the input file. It uses the ***Evaluation*** object to create a function that will run each evaluator software with the desired input parameters and return the output parameters calculated by the evaluator software. Next, it uses the ***ToolboxGenerator*** to create an initialized DEAP toolbox object based on the input file’s **algorithm** specifications. The ***ToolboxGenerator*** object accepts the ***Evaluation*** object and registers it as the toolbox’s ‘evaluate’ tool. Then, it initializes a ***Constraints*** object to contain **constraints** specified in the input file. Next, the ***Executor*** initializes an ***Algorithm*** object that accepts the initialized DEAP toolbox and ***Constraints*** object. Finally, the ***Executor*** class uses a method in the ***Algorithm*** object to run the general genetic algorithm. The ***Executor*** class uses the hyperparameters from the DEAP toolbox, applies constraints defined in the ***Constraints*** object, and calculates objective functions using the evaluation function created by the ***Evaluation*** object; all the while saving the results using the ***Backend*** class.

In the ROLLO Github repository [12], I include a tests directory that contains unit tests for all methods in the classes described above. These tests serve both to document how to use ROLLO as well as to ensure ROLLO has feature stability.

4.3.1 Installing and Running ROLLO

There are two ways to install ROLLO. First, a user may use a package manager such as The Python Package Index (PyPI) to install ROLLO: `python -m pip install rollo` [12]. Second, a user can download the ROLLO repository [12] from Github and install it from source.

Users run ROLLO from the command line interface. A user must first set up the ROLLO JSON input file and evaluator scripts in a directory. When running ROLLO from the command line, there is one mandatory argument and two optional arguments. The mandatory argument is the input file (`-i`). The optional arguments are the checkpoint file (`-c`) and verbosity flag (`-v`). The checkpoint file holds the results from the ROLLO simulation. The checkpoint file also acts as a restart file. If a ROLLO simulation ends prematurely, the checkpoint file can be used to restart

the simulation from the most recent generation. The verbosity flag turns on verbose output. The structure of a command line input for running ROLLO is:

```
python -m rollo -i <input file name> -c <checkpoint file name> -v
```

The checkpoint file holds the ROLLO simulation results and acts as a restart file. Thus, if a ROLLO simulation ends prematurely, users can use the checkpoint file to restart the code from the most recent population and continue the simulation.

4.3.2 ROLLO Parallelization

During a ROLLO simulation, ROLLO runs nuclear reactor modeling software to calculate the fitness values for each reactor model in each generation. For example, a ROLLO simulation that runs for 5 generations with 100 reactor models will require 500 separate reactor modeling software runs. Computing hundreds of fitness values with reactor modeling software is computationally expensive, necessitating ROLLO to run in parallel on supercomputers.

ROLLO has a serial run mode and two modes for parallelization. The serial mode runs each reactor model in each generation serially and utilizes a map() function to run the nuclear software for each reactor model. The map function accepts a function and iterable and returns a map object of the results by applying the function to each item of the iterable. In ROLLO, the ***Evaluation*** class generates the function that accepts the reactor model's control variables and runs the nuclear software; and the iterable is a list containing the control variables for each reactor model. The first parallel mode, *multiprocessing*, replaces the default map function with the `multiprocessing_on_dill` map [100]. The `multiprocessing_on_dill` map splits the iterable into several chunks which it submits to the process pool as separate tasks. This *multiprocessing* mode is useful for parallelizing runs on a local machine or a single node on a computer cluster. However, it is unable to parallelize across distributed memory systems.

The second parallel mode, `job_control`, uses the Unix system's job control features to give users more flexibility with parallelization setup. This flexibility enables parallelization across distributed memory systems such as clusters and supercomputers. The second parallel mode does not use the map() function. Like the *multiprocessing* mode, the `job_control` mode generates a function using the ***Evaluation*** class that accepts the reactor model's control variables and runs

the nuclear software. ROLLO will then generate a combined bash command that launches multiple evaluation function calls for the different reactor models by backgrounding each command. For example, a ROLLO simulation with a population size of 2 has a combined command that looks like this:

```
cd 0_0
aprun -n 2 python program1.py &
sleep 1
cd ../0_1
aprun -n 2 python program1.py &
sleep 1
wait
```

Figure 4.6: Example combined bash command generated by ROLLO’s parallel mode `job_control1`. Directories `0_0` and `0_1` refer to the first generation’s first and second reactor model (indexed by 0).

Users define each command in ROLLO’s input file; thus, the `job_control` mode enables more control over the parallelization settings of each command. Many nuclear software, such as OpenMC and MOOSE, use Message Passing Interface (MPI) or OpenMP to parallelize their runs. With control over each command, users can continue to utilize the parallel versions of the nuclear software, thus, having two layers of parallelization. The first layer is the individual software’s parallelization, and the second layer is ROLLO’s parallelization. For example, running four reactor models using ROLLO across 16 nodes on a cluster and assigning each reactor model to run in parallel across four nodes. Users can find in-depth details about the parallelization setup in the ROLLO documentation [98].

4.3.3 ROLLO Results Analysis

The `Backend` class manages results from each ROLLO simulation. `Backend` saves all the results in a checkpoint file which will output into the same directory as the input file at the end of the simulation. The checkpoint file can be loaded into a Jupyter notebook and organized to produce desired plots. Users can find examples of ROLLO results analysis in the ROLLO documentation [98].

The evaluation function creates a new directory for each generation and individual, where it then stores all the evaluators' templated input files and output files associated with that particular run. The generation and individual values are indexed by zero. For example, the directory containing files associated with the tenth individual in the genetic algorithm's third generation will be named: `2_9`.

4.4 ROLLO Verification

I conducted multiple verification studies to verify ROLLO's optimization capabilities: commonly used evolutionary algorithm single and multi-objective benchmark problems and a nuclear reactor-specific problem. They prove that ROLLO's evolutionary algorithm is both correctly implemented and suitable for conducting nuclear reactor optimization problems.

4.4.1 Ackley Function

The Ackley function, shown in Equation 4.1, is a function with a large number of local minima but only one global minimum, thus, commonly used as a performance test for single-objective optimization algorithms [101]. An effective single-objective optimization algorithm should find the Ackley function's global minimum point.

$$f(x) = -a \cdot \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + a + \exp(1) \quad (4.1)$$

The recommended variable values are $a = 20$, $b = 0.2$, and $c = 2\pi$ [102]. The Ackley function's global minimum point is $f(0, 0) = 0$. Figure 4.7 shows the resulting two-variable Ackley function.

I added an integration test to ROLLO that checks that a default ROLLO simulation will find the Ackley function's global minimum point. If ROLLO performed sub-optimally, it would return one of the Ackley function's many local minimums.

4.4.2 Binh and Korn Function

The Binh and Korn function [103], shown in Equation 4.2, is a two-objective function with a widely spread out Pareto front. An effective multi-objective optimization algorithm should successfully

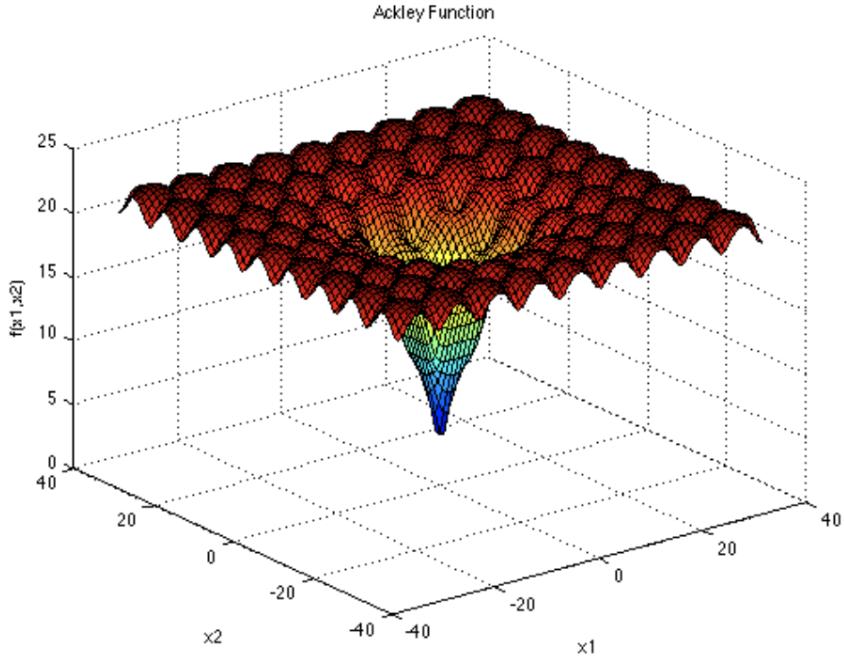


Figure 4.7: Two variable Ackley function. Figure reproduced from [102].

find widely spread solutions in the obtained Pareto front [66], thus, I used the Binh and Korn function as a performance test for ROLLO's multi-objective optimization capabilities.

$$\begin{aligned}
 \text{Minimize} = & \begin{cases} f_1(x_1, x_2) = 4x_1^2 + 4x_2^2 \\ f_2(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 5)^2 \end{cases} \\
 \text{Such that} = & \begin{cases} (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7 \\ 0 \leq x_1 \leq 5, 0 \leq x_2 \leq 3 \end{cases}
 \end{aligned} \tag{4.2}$$

Figure 4.8 shows the Binh and Korn function's Pareto front. I use the hypervolume indicator to quantify the Pareto front's quality. The hypervolume indicator measures the volume (in the objective space) covered by non-dominated solutions for problems in which all objectives are to be minimized [66]. Section 4.5.1 details the hypervolume indicator calculation. I added an integration test that checks that a ROLLO simulation will find a hypervolume comparable to the ideal Pareto front when minimizing the Binh and Korn function.

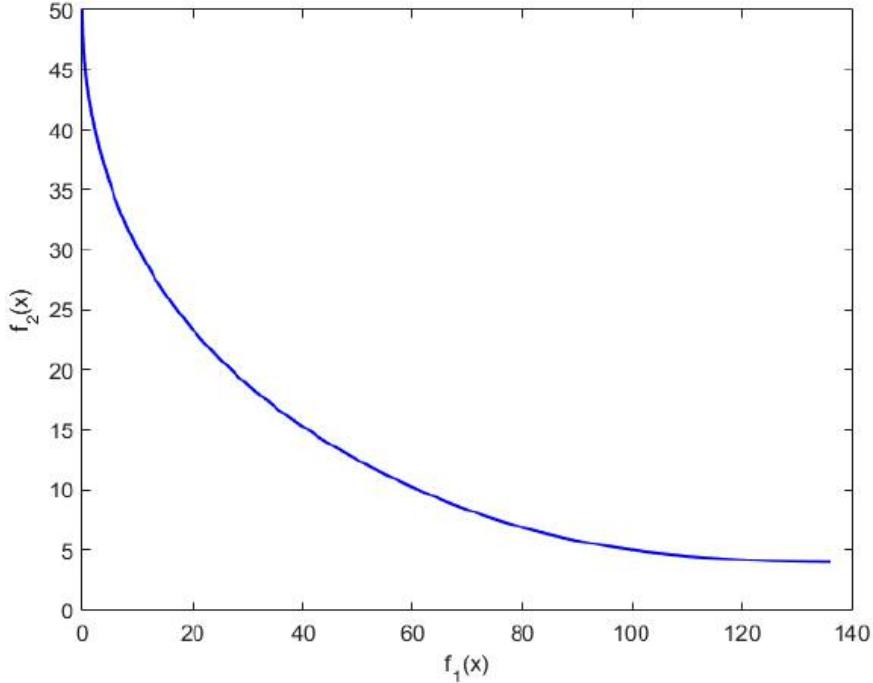


Figure 4.8: Pareto front of the Binh and Korn test function. Figure reproduced from [104].

4.4.3 Pu-239 Critical Bare Sphere

I ran a neutronics verification problem: finding the critical radius for a ^{239}Pu bare sphere, to ensure successful coupling between ROLLO and reactor neutronics modeling software, OpenMC [13]. The solution to this problem is well studied and readily available in the literature [105]. Blanchard et al reported that with MCNP4b code and ENDF/B-VI data library, the critical mass of a Pu239 bare sphere is 10.00 kg which corresponds to a diameter of 9.9cm [105]. Figure 4.9 shows the ROLLO input file for this verification problem. In the input file, I vary the radius between 1.0 and 8.0cm, with the objective of minimizing the radius while constraining the problem to have $k_{eff} \geq 1.0$. Figure 4.10 shows the OpenMC template used in the ROLLO simulation.

```

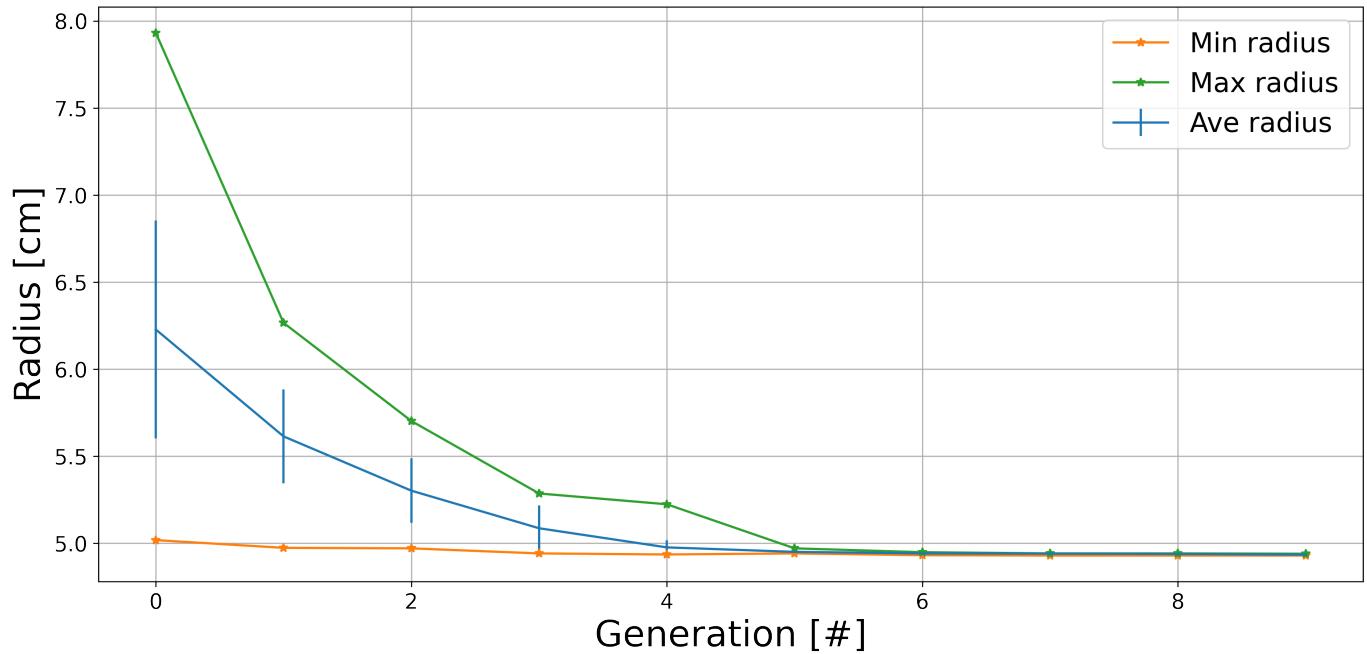
1      {"control_variables": {
2          "radius": {"min": 1.0, "max": 8.0}
3      },
4      "evaluators": {
5          "openmc": {
6              "order": 0,
7              "input_script": ["python", "critical_sphere.py"],
8              "inputs": ["radius"],
9              "outputs": ["keff", "radius"],
10             "output_script": ["python", "get_sphere_keff.py"]
11         }
12     },
13     "constraints": {"keff": {"operator": [">="],
14                         "constrained_val": [1.0]}},
15     "algorithm": {
16         "parallel": "multiprocessing",
17         "keep_files": "none",
18         "objective": ["min"],
19         "optimized_variable": ["radius"],
20         "pop_size": 80,
21         "generations": 10
22     }
23 }
```

Figure 4.9: Reactor evOLutionary aLgorithm Optimizer (ROLLO) JSON input file for finding the minimum radius for a ^{239}Pu Critical Bare Sphere.

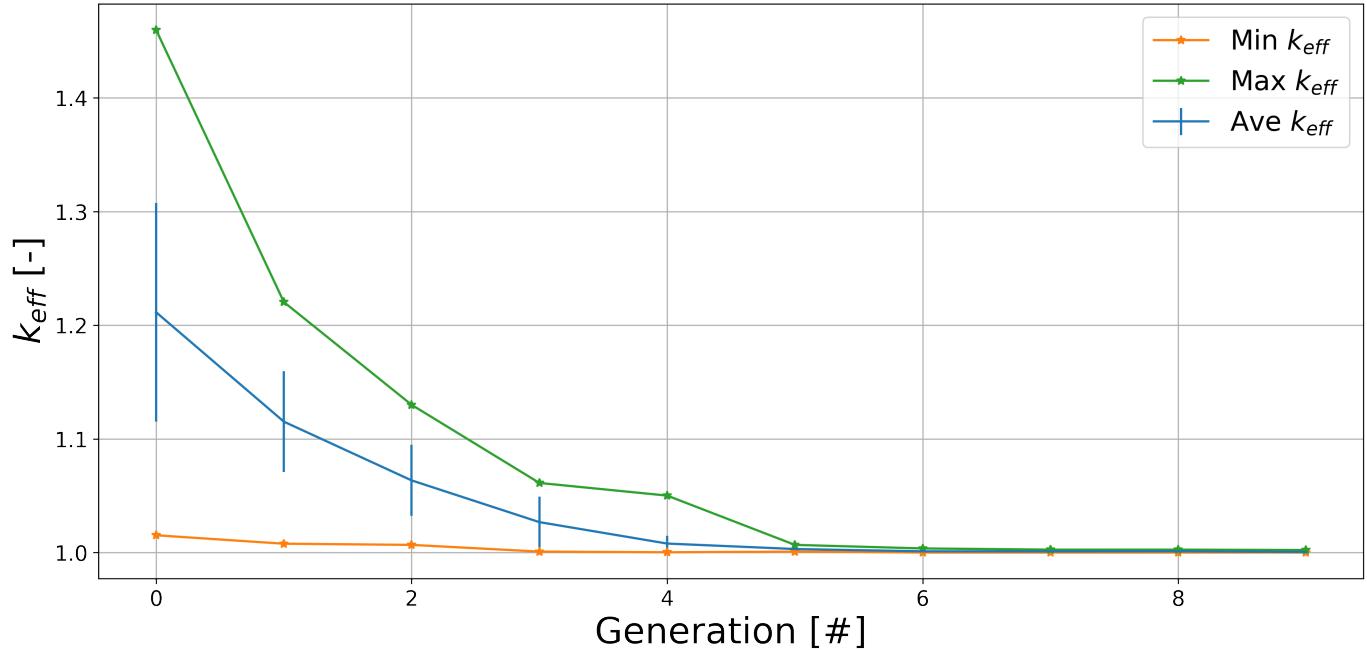
ROLLO successfully finds the critical radius of the ^{239}Pu bare sphere to be 4.9856cm which corresponds to approximately a 9.9cm diameter. The critical sphere's k_{eff} value is 1.000919 ± 0.000048 . Figure 4.11a and 4.11b show the radius and k_{eff} evolution through the evolutionary algorithm's generations. They demonstrate how the average radius converges towards the critical radius while constraining $k_{eff} \geq 1$ and improving with each generation.

```
1      import openmc
2      import numpy as np
3
4      pu = openmc.Material()
5      pu.set_density("g/cm3", 19.84)
6      pu.add_nuclide("Pu239", 1)
7      mats = openmc.Materials([pu])
8
9      radius = {{radius}}
10
11     fuel_sphere = openmc.Sphere(r=radius, boundary_type='vacuum')
12     fuel_cell = openmc.Cell(fill=pu, region=-fuel_sphere)
13     univ = openmc.Universe(cells=[fuel_cell])
14     geom = openmc.Geometry(univ)
15
16     settings = openmc.Settings()
17     settings.batches = 100
18     settings.inactive = 20
19     settings.particles = 20000
20     settings.temperature = {"multipole": True, "method": "interpolation"}
21
22     mats.export_to_xml()
23     geom.export_to_xml()
24     settings.export_to_xml()
25     openmc.run()
```

Figure 4.10: OpenMC template input file used in ROLLO simulation to find the minimum radius for a ^{239}Pu Critical Bare Sphere.



(a) Minimum, average, and maximum radius values evolution.



(b) Minimum, average, and maximum k_{eff} values evolution.

Figure 4.11: Results for each generation for ROLLO's genetic algorithm optimization to the find the critical radius of a ^{239}Pu bare sphere.

4.5 ROLLO Convergence Criteria

The `generations` input parameter defines ROLLO’s evolutionary algorithm convergence criterion. Each nuclear reactor model’s evaluation is computationally intensive. Users will most likely be constrained by the total available compute time. Therefore, rather than setting a results-based stopping criterion, ROLLO enables users to define the `generations` and `pop_size` based on the amount of compute time available to them:

$$t_{total} = t_{eval} \times gen \times pop \quad (4.3)$$

where

t_{total} = Total compute time available

t_{eval} = compute time per nuclear reactor model evaluation

gen = total number of generations in optimization process

pop = population size in optimization process

4.5.1 Convergence

ROLLO does not utilize a mathematical expression to evaluate problem convergence. The complexity of reactor design optimization means that no single indicator determines if convergence is met. Instead, ROLLO’s purpose is to help the human reactor designer narrow down the design space. When a ROLLO run completes, the user plots the objectives’ convergence and Pareto front (for multi-objective simulations) to evaluate if they are confident about the final solution set.

For a single-objective optimization problem, a reactor designer can determine if convergence criteria is met by checking if the objective’s average and minimum values are no longer changing. For a multi-objective optimization problem, a reactor designer can determine if convergence criteria is met by evaluating if the difference between generations’ hypervolume values are getting smaller. The hypervolume indicator is the most used set-quality indicator for assessing stochastic multi-objective optimizers [106]. The hypervolume indicator measures the volume (in the objective space)

covered by non-dominated solutions for problems in which all objectives are to be minimized [66]. Figure 4.12 illustrates the hypervolume enclosed by the non-dominated solutions (A, B, C, D, E) and reference point, W, in the hatched region for a two-dimensional problem. The hypervolume is calculated by finding the volume between the reference point and the objective values of the Pareto front's solutions. The reference point must be selected to ensure that every solution falls within the hypervolume. If the simulation is not converged, the user can easily restart the optimization simulation using the `checkpoint.pkl` file and run the problem for a few more generations.

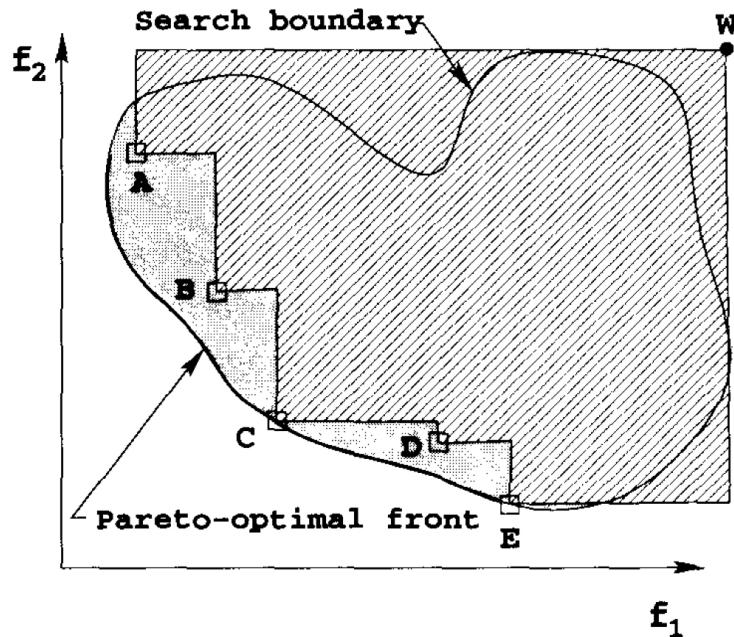


Figure 4.12: Example hypervolume enclosed by non-dominated solutions. Figure reproduced from [66]. The f_1 and f_2 axes correspond to the optimization objectives in a ROLLO simulation.

4.6 Summary

This chapter described the Reactor evOLutionary aLgorithm Optimizer (ROLLO) framework developed for this dissertation. ROLLO is a Python package that applies evolutionary algorithm optimization techniques to nuclear reactor design using the Distributed Evolutionary Algorithms in Python (DEAP) module. The motivation for ROLLO is to enable reactor designers to utilize robust evolutionary algorithm optimization methods without going through the cumbersome process of setting up a genetic algorithm framework, selecting appropriate hyperparameters, and setting up its parallelization. I designed ROLLO to be effective, flexible, open-source, parallel, reproducible, and usable. ROLLO is hosted on Github [12].

Chapter 5

AHTR Modeling and Optimization Methodology

In this chapter, I describe the modeling methodology of the Advanced High-Temperature Reactor (AHTR) plank and one-third assembly optimization for non-conventional geometries and parameters conducted using the Reactor evOLutionary aLgorithm Optimizer (ROLLO) software. The optimization tool should enable the placement of fuel, moderator, and coolant material in any possible location, within physical limits, to wholly explore the design space enabled by additive manufacturing. As exploring the design space of non-conventional geometries and parameters on reactors has been applied in very limited circumstances (previous attempts described in Section 2.3.2), this dissertation must systematically explore and constrain the large design space. As this work is a proof of concept, this dissertation is only an intermediate step towards developing a truly arbitrary geometry expression.

The subsequent sections define the optimization problem (Section 5.1), describe the AHTR geometries (Section 5.2), outline the AHTR modeling workflow (Section 5.3), verify the AHTR models (Section 5.4), and describe the process of tuning ROLLO's hyperparameters (Section 5.5). Appendix A lists all the data and analysis related to this chapter to enable the reproduction of all the simulations.

5.1 ROLLO Optimization Problem Definitions

In an effort toward optimizing the AHTR design for non-conventional geometries and parameters, I chose to vary the following AHTR parameters:

- Tristructural Isotropic (TRISO) fuel packing fraction distribution ($\rho_{TRISO}(\vec{r})$),
- total fuel packing fraction (PF_{total}), and

- coolant channel shape.

Varying the TRISO fuel packing fraction distribution allows us to explore how heterogeneous fuel distributions impact reactor performance. In Section 3.3.1, the Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark results demonstrated that increased fuel packing does not always correspond with increased k_{eff} due to self-shielding effects. Varying total fuel packing fraction and TRISO distribution enables synergistic exploration of how heterogenous TRISO distribution could minimize self-shielding and thus, reduce the fuel required for a reactor design. The coolant channel shape variation enables exploration of how non-uniform channel shapes impacts heat transfer.

I selected three key AHTR optimization objectives that address contrasting reactor core qualities. Table 5.1 describes each objective, how I quantified them, and the motivation.

Table 5.1: Reactor evOLutionary aLgorithm Optimizer (ROLLO) Advanced High-Temperature Reactor (AHTR) optimization problem objectives with their quantification descriptions and motivation.

Objective	Quantification	Motivation
Minimize fuel amount	Minimize total fuel packing fraction	Cost savings, Non-proliferation
Maximize heat transfer	Minimize maximum temperature	Minimize thermal stress in the fuel
Minimize power peaking	Minimize power peaking factor normalized by fuel distribution	Efficient fuel utilization, longer core life, safety

To optimize the objectives, I will vary the parameters in the AHTR plank and one-third assembly models. The optimization results for the AHTR plank and one-third assembly models are reported in Chapters 6 and 7, respectively. The AHTR plank optimization is a preliminary study to inform the more complex AHTR one-third assembly optimization setup. In the next section, I will describe both geometries.

5.2 AHTR Geometry for Optimization Problem

I apply the optimization process to the AHTR plank and AHTR one-third assembly geometries. The geometries are adapted from the FHR benchmark's AHTR design, outlined in Chapter 3 (see Figure 3.2). The main differences occur in the fuel plank region. In the FHR benchmark,

the TRISO particles are arranged in rectangular lattices within two fuel stripes in the fuel plank region, shown in Figure 5.1.

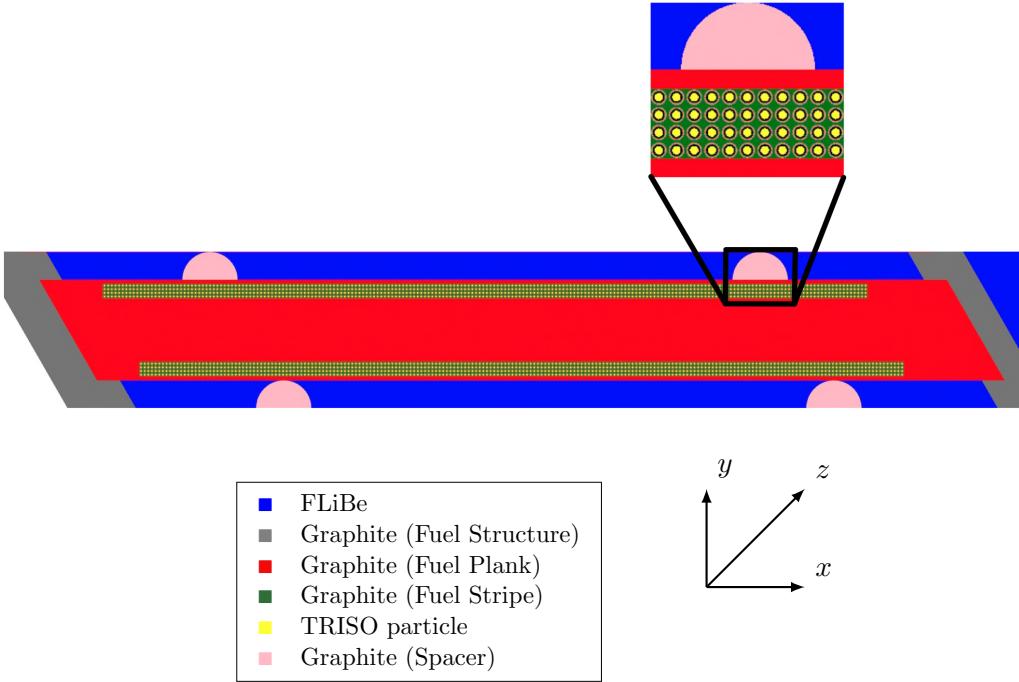


Figure 5.1: Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark’s Advanced High-Temperature Reactor (AHTR) design’s fuel plank, with the magnification of a spacer and segment of the fuel stripe with embedded TRISO particles.

For the AHTR plank and AHTR one-third assembly optimization geometries, I discretized each plank into ten cells with random TRISO packing and an individually controlled packing fraction. Thus, the AHTR plank has ten fuel cells, and the AHTR one-third assembly has 60 fuel cells (10 cells for 6 planks). The subsequent subsections describe each geometry in further detail. I also omit the graphite spacers.

5.2.1 AHTR Plank Geometry

The AHTR plank is a single graphite fuel plank model from the AHTR design (Figure 3.2). I modified the fuel plank to be straightened with perpendicular sides instead of slanted as in Figure 3.4 for ease of modeling. The one-third assembly optimization uses the original slanted AHTR planks. Figure 5.2 illustrates the straightened fuel plank with ten fuel cells with random TRISO packing in each cell. The plank has $27.1 \times 3.25 \times 1.85 \text{ cm}^3$ in the x, y, and z dimensions, respectively,

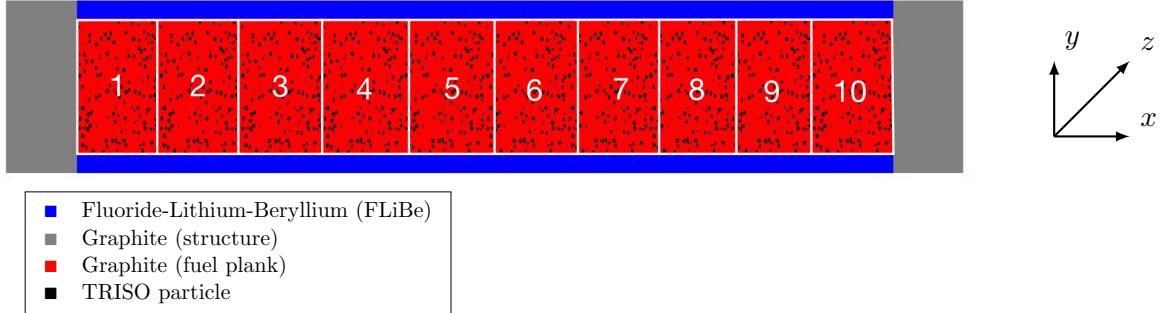


Figure 5.2: Straightened Advanced High-Temperature Reactor (AHTR) fuel plank with 10 fuel cells with random TRISO packing, graphite buffers, and straight FLiBe coolant channels. This geometry is used for AHTR plank optimization.

and reflective boundary conditions.

I used the same materials as in the FHR benchmark (Chapter 3), except that I homogenized each TRISO particle's four outer layers (porous carbon buffer, inner pyrolytic carbon, silicon carbide layer, and the outer pyrolytic carbon) into a single layer. The TRISO kernel and outer radius dimensions remain the same. Table 5.2 reports OpenMC's k_{eff} for the straightened AHTR configuration with and without the four outer layers TRISO homogenization. The TRISO particle outer four-layer homogenization resulted in a 30% speed-up without compromising accuracy with k_{eff} values within each other's uncertainty. As a result, the homogenized models are used for all subsequent optimization efforts.

Table 5.2: Straightened Advanced High-Temperature Reactor (AHTR) fuel plank k_{eff} for the case with no TRISO homogenization and case with homogenization of the four outer layers. Both simulations were run on one BlueWaters supercomputer XE Node [87] using OpenMC [13] with 80 active cycles, 20 inactive cycles, and 8000 particles.

TRISO Homogenization	k_{eff} [-]	Simulation Time [s]
None	1.38548 ± 0.00124	233
Four outer layers	1.38625 ± 0.00109	168

5.2.2 AHTR One-Third Assembly Geometry

The AHTR one-third assembly is one-diamond shape sector of the AHTR assembly (Figure 3.2). The one-third assembly contains six graphite fuel planks. Each graphite fuel plank has graphite buffers and ten rectangular prism fuel cells with random TRISO packing and individually controlled packing fractions. Figure 5.3 shows the one-third AHTR assembly with 10 x 6 fuel cells with random

TRISO packing. The one-third assembly model has reflective boundary conditions and uses the TRISO particle outer four-layer homogenization described in Section 5.2.1.

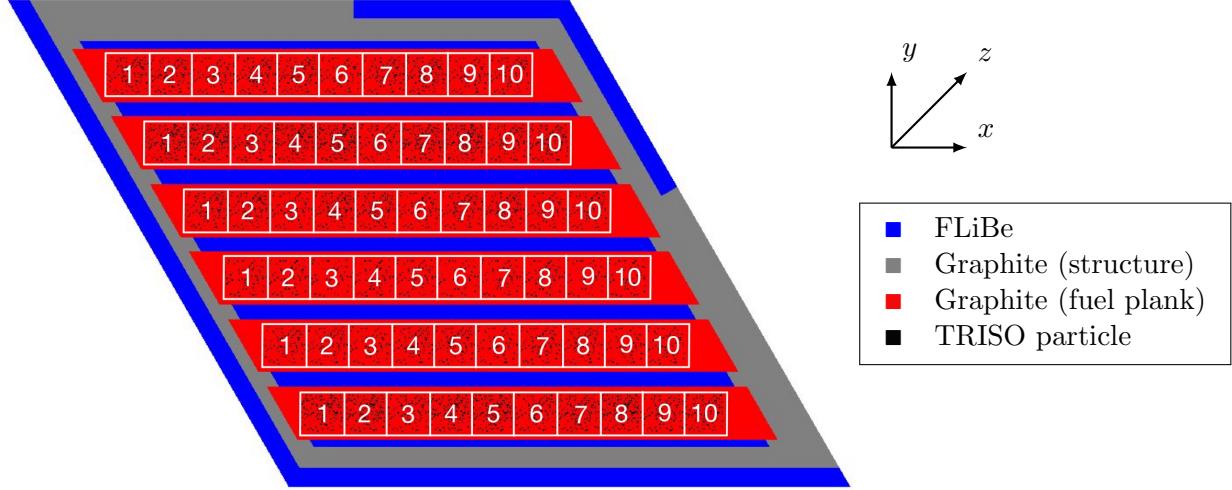


Figure 5.3: Advanced High-Temperature Reactor (AHTR) one-third assembly with ten randomly packed fuel cells in each graphite plank, graphite structure, and FLiBe coolant.

5.3 AHTR Model Workflow

The ROLLO software drives the evolutionary algorithm optimization process, depicted in Figure 4.1. In a ROLLO input file, I define control variables that the genetic algorithm uses to vary the AHTR parameters described in Section 5.1. The control variables are input into the nuclear software templates to model different AHTR geometries. The nuclear software will then run the AHTR models and calculate the optimization objective and constraint values. In this work, I use OpenMC [13] to model AHTR’s neutronics and Moltres [14] to model the AHTR’s multi-physics. Descriptions of each software can be found in Section 2.1.4.

In the following subsections, I describe the AHTR modeling workflow: the AHTR input parameter variations, the OpenMC and Moltres models, and the output and constraint values calculations. The AHTR modeling workflow described in this section falls within the nuclear software evaluation orange blocks in Figure 4.1’s genetic algorithm flow chart. Each AHTR modeling workflow models

a single reactor model. Figure 5.4 illustrates the AHTR modeling workflow.

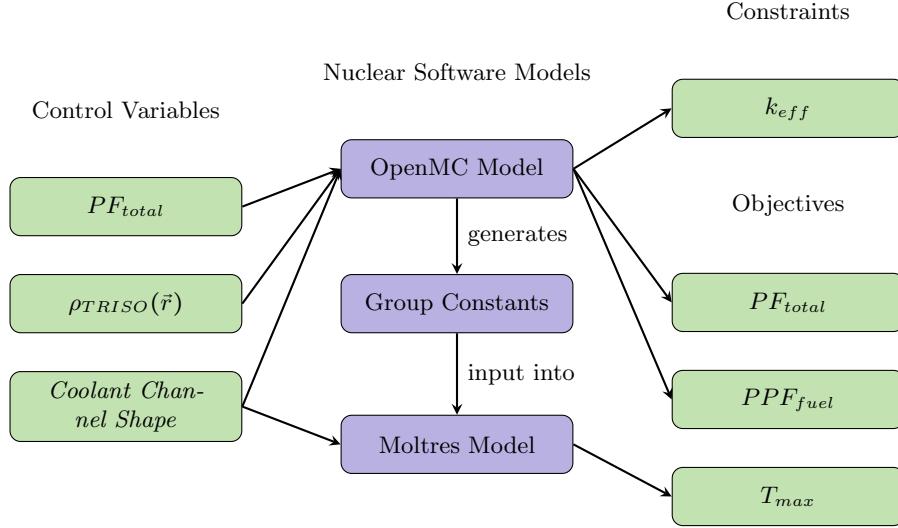


Figure 5.4: Advanced High-Temperature Reactor (AHTR) modeling workflow in Reactor evOLutionary aLgorithm Optimizer (ROLLO) optimization. For each reactor model in the optimization process, the modeling workflow begins with ROLLO selecting the control variables within the user-defined range. ROLLO inserts control variables and runs the templated OpenMC and Moltres input files. ROLLO analyzes the OpenMC and Moltres output files to determine the constraint and objective values. PF_{total} : Total fuel packing fraction, $\rho_{TRISO}(\vec{r})$: TRISO packing fraction distribution, PPF_{fuel} : normalized power peaking factor, T_{max} : model's maximum temperature.

5.3.1 Input Parameter Modeling

This subsection describes how I model the AHTR input parameter variations: total fuel packing fraction (PF_{total}), TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), and coolant channel shape. For both the AHTR plank and one-third assembly models, the PF_{total} parameter is a single numerical input. Following this, I describe how I modeled the other two parameters for the AHTR plank and one-third assembly models.

Input Parameter Modeling: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$)

I utilize sine distributions for both the AHTR plank and one-third assembly models to govern the $\rho_{TRISO}(\vec{r})$. First, I impose an overall sine distribution to represent fuel packing in the model. The model calculates each fuel cell's packing fraction based on the midpoint x-value of the sine distribution in the cell. Then, I use OpenMC's `pack_spheres` function to randomly disperse the packing

fraction in each fuel cell. For the AHTR plank, one sine distribution governs the $\rho_{TRISO}(\vec{x})$ across the AHTR plank's x-direction. For the AHTR one-third assembly, two sine distributions govern the $\rho_{TRISO}(\vec{x}, \vec{y})$ across each fuel plank's x-direction and the assembly's y-direction, respectively.

For the AHTR plank, the sine distribution that governs the TRISO packing fraction distribution across the ten fuel cells in the x-direction is given by Equation 5.1.

$$\rho_{TRISO}(\vec{x}) = (a \cdot \sin(b \cdot x + c) + 2) \cdot NF \quad (5.1)$$

where

$\rho_{TRISO}(\vec{x})$ = TRISO packing fraction distribution across ten cells [-]

a = amplitude, peak deviation of the function from zero [-]

b = angular frequency, rate of change of the function argument [$radians \cdot cm^{-1}$]

c = phase, the position in its cycle the oscillation is at $t = 0$ [$radians$]

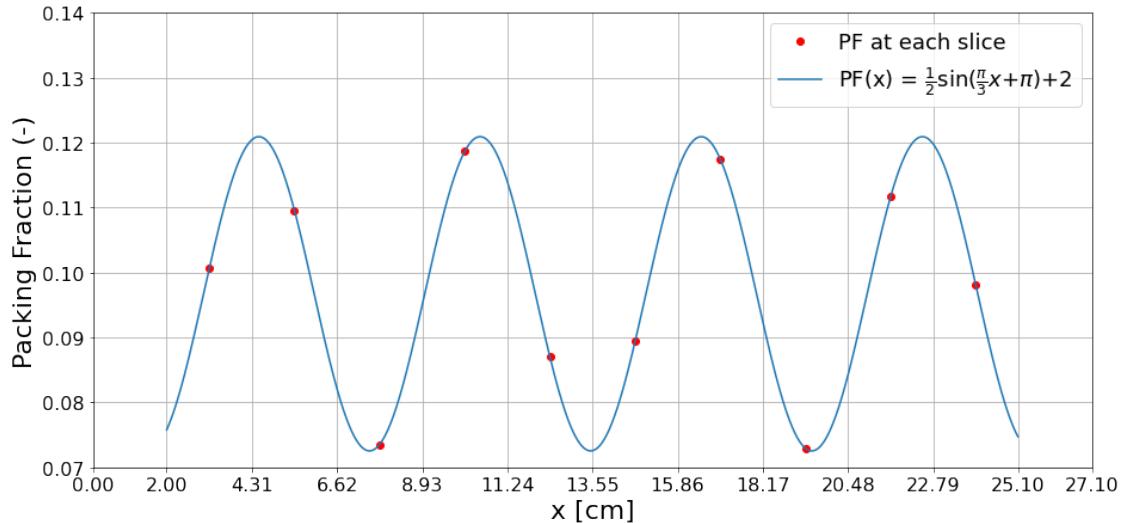
x = midpoint x-value for each cell [cm]

NF = normalization factor [-]

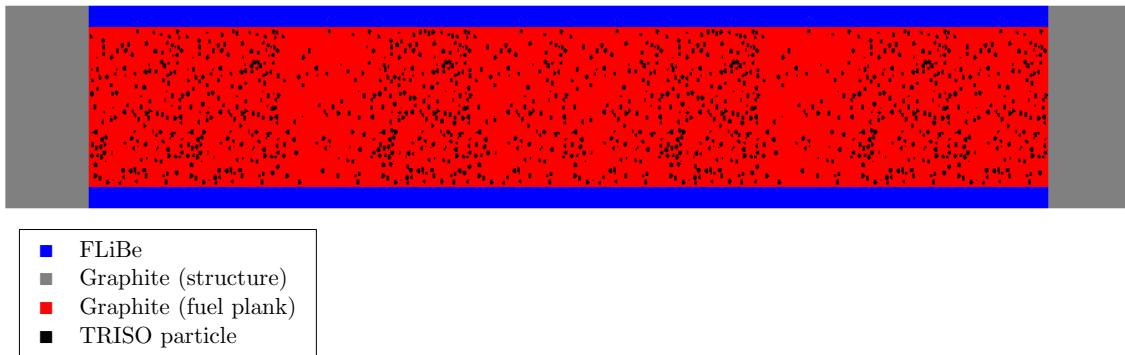
Figure 5.2 depicted the ten fuel cells. The sine distribution's coefficients, a b c , are the control variables ROLLO utilizes to manipulate the plank's $\rho_{TRISO}(\vec{x})$. Thus, ROLLO will vary the a , b , c variables to find an optimal $\rho_{TRISO}(\vec{x})$ that optimizes objectives. The normalization factor normalizes each fuel cell to ensure that the total volume of TRISO particles in the plank corresponds to PF_{total} .

An AHTR plank with $PF_{total} = 0.0979$, and $\rho_{TRISO}(\vec{x}) = (0.5 \cdot \sin(\frac{\pi}{3} \cdot x + \pi) + 2) \cdot NF$, results in the following packing fractions for the ten cells, respectively: 0.103, 0.120, 0.049, 0.138, 0.076, 0.081, 0.136, 0.048, 0.125, and 0.098. Figure 5.5b shows this sine distribution, highlights the packing fraction at the respective midpoints, and displays the plank's x-y axis view with the packing fraction varying based on this sine distribution.

For the AHTR one-third assembly, Equation 5.2's two sine distributions govern TRISO packing



(a) TRISO packing fraction distribution values.



(b) TRISO distribution in plank model.

Figure 5.5: Straightened Advanced High-Temperature Reactor (AHTR) fuel plank with varying TRISO particle distribution across 10 cells based on the sine distribution:
 $\rho_{TRISO}(\vec{x}) = (0.5 \sin(\frac{\pi}{3}x + \pi) + 2) \times NF$. Figure 5.5a's packing fraction values correspond to Figure 5.5b's TRISO distribution in the plank model.

fraction distributions in the x and y direction for the 10 x 6 fuel cells:

$$\rho_{TRISO}(\vec{x}, \vec{y}) = (a \cdot \sin(b \cdot x + c) + 2) \cdot (d \cdot \sin(e \cdot y + f) + 2) \cdot NF \quad (5.2)$$

where

$\rho_{TRISO}(\vec{x}, \vec{y})$ = TRISO packing fraction distribution across 60 cells [-]

a, d = amplitude, peak deviation of the function from zero [-]

b, e = angular frequency, rate of change of the function argument [radians · cm⁻¹]

c, f = phase, the position in its cycle the oscillation is at t = 0 [radians]

x = midpoint value for each x-direction fuel cell [cm]

y = midpoint value for each fuel plank [cm]

NF = normalization factor [-]

Figure 5.3 depicts the 10 x 6 fuel cells. The sine distribution's coefficients, $a b c d e f$, are the control variables ROLLO utilizes to manipulate the $\rho_{TRISO}(\vec{x}, \vec{y})$ in the one-third assembly. Thus, ROLLO will vary $a b c d e f$ constants to find an optimal $\rho_{TRISO}(\vec{x}, \vec{y})$ that optimizes objectives. The normalization factor normalizes each fuel cell to ensure that the total volume of TRISO particles in the one-third assembly corresponds to PF_{total} .

An AHTR one-third assembly with $PF_{total} = 0.1$ and $\rho_{TRISO}(\vec{x}, \vec{y}) = (0.5 \cdot \sin(1 \cdot x + 1) + 2) \cdot (0.7 \cdot \sin(1.5 \cdot y + 2) + 2) \cdot NF$ results in the packing fraction distribution shown in Figure 5.6a, which corresponds to the one-third assembly with varying TRISO distribution in its fuel cells, shown in Figure 5.6b.

Input Parameter Modeling: Coolant Channel Shape

For both the AHTR plank and one-third assembly models, I simulate the variation in coolant channel shape with a sinusoidal pattern using OpenMC's cylinder surfaces functionality. By varying the cylinders' radii, the coolant channel shapes' depth and frequency mimic a sinusoidal pattern. I vary the FLiBe coolant channel shape while holding the total coolant volume constant. Holding

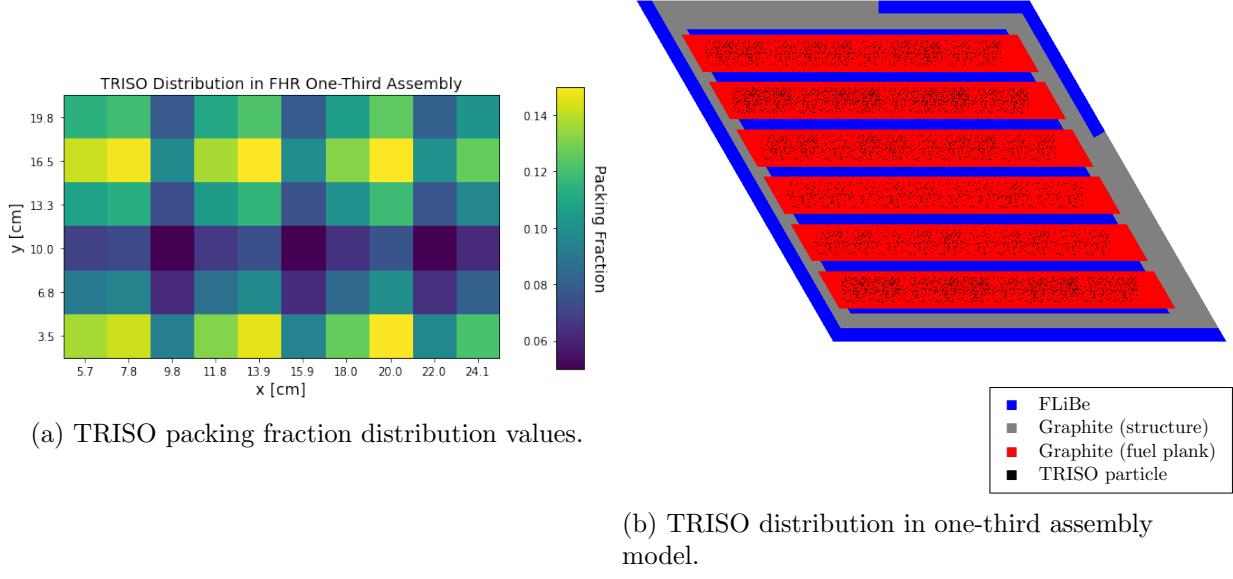


Figure 5.6: Advanced High-Temperature Reactor (AHTR) one-third assembly with varying TRISO particle distribution across 10×6 cells based on the sine distributions:
 $\rho_{TRISO}(\bar{x}, \bar{y}) = (0.5 \cdot \sin(1 \cdot x + 1) + 2) \cdot (0.7 \cdot \sin(1.5 \cdot y + 2) + 2) \cdot NF$. Figure 5.6a's packing fraction values correspond to Figure 5.6b's TRISO distribution in the one-third assembly model.

the coolant volume constant throughout the coolant channel optimization process enables the use of the same heat transfer coefficient (h) for all the AHTR Moltres temperature models of varying coolant channel shapes.

For the AHTR plank, the r_{top} and r_{bot} variables control the coolant channel shape on the top and bottom FLiBe channels. Figure 5.7 shows the AHTR plank's coolant channel shape for $r_{top} = 0.2$ and $r_{bot} = 0.3$. Thus, ROLLO will vary r_{top} and r_{bot} to find optimal coolant channel shapes that optimizes objectives.

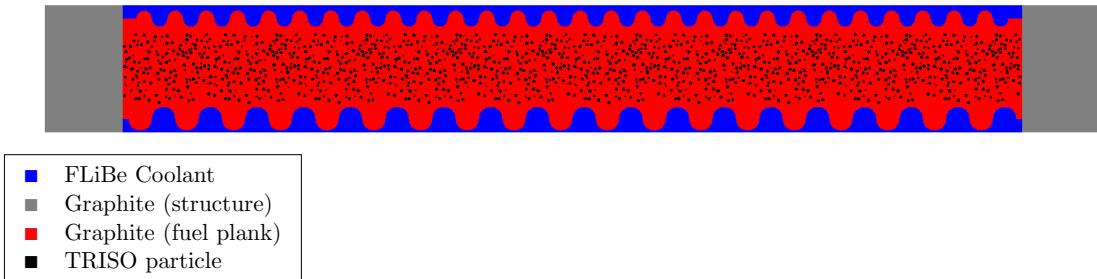


Figure 5.7: Advanced High-Temperature Reactor (AHTR) plank with coolant channel shape variation, $r_{top} = 0.2\text{cm}$ and $r_{bot} = 0.3\text{cm}$.

For the AHTR one-third assembly, r_1, r_2, r_3, r_4, r_5 variables control the coolant channel shape in the inter-plank FLiBe. Figure 5.8 shows the AHTR one-third assembly's inter-plank coolant channel shapes for $r_1, r_2, r_3, r_4, r_5 = 0.3, 0.2, 0.1, 0.2, 0.3$. Thus, ROLLO will vary r_1, r_2, r_3, r_4, r_5 to find optimal coolant channel shapes for the AHTR one-third assembly model. I did not vary the shape of the one-third assembly's top and bottom coolant channels, since they have half the thickness of the inner coolant channels. r_1, r_2, r_3, r_4, r_5 are sufficient to show the effect of changing the coolant channel shape.

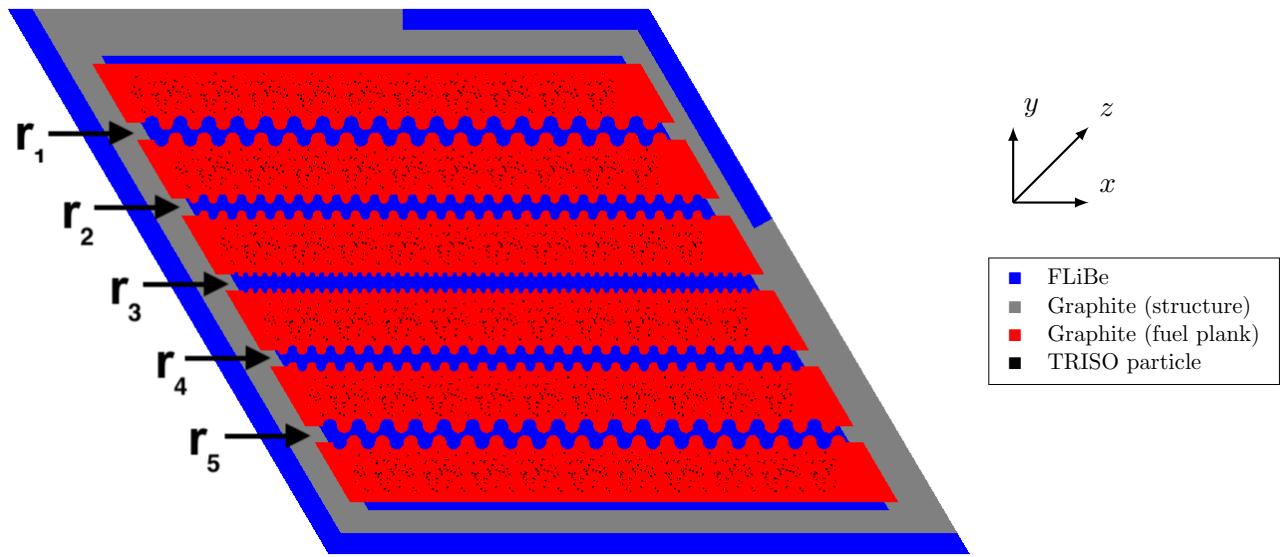


Figure 5.8: Advanced High-Temperature Reactor (AHTR) one-third assembly with coolant channel shape variation, $r_1, r_2, r_3, r_4, r_5 = 0.3\text{cm}, 0.2\text{cm}, 0.1\text{cm}, 0.2\text{cm}, 0.3\text{cm}$.

5.3.2 AHTR OpenMC and Moltres Models

The input parameters outlined in the previous section are inputs for the OpenMC neutronics and Moltres temperature models. The Moltres model relies on group constant data produced by the OpenMC model. The OpenMC and Moltres models' workflow are:

1. Run transport with OpenMC neutronics model to calculate k_{eff} and fuel-normalized power peaking factor (PPF_{fuel}) objective value

2. Produce group constant data for the Moltres model with the OpenMC neutronics model
3. Generate mesh for Moltres model
4. Run Moltres temperature model (accepts group constant data and mesh) to calculate maximum temperature (T_{max}) objective value

In the subsequent subsections, I describe details and assumptions for each step of the OpenMC and Moltres models' workflow.

AHTR OpenMC Neutronics Model

The OpenMC model template generates an AHTR model from the following ROLLO input parameters: total fuel packing fraction (PF_{total}), TRISO distribution (a, b, c, d, e, f), and coolant channel shapes ($r_{top}, r_{bot}, r_1, r_2, r_3, r_4, r_5$). ROLLO takes these input parameters and generates an OpenMC AHTR model with TRISO distribution and Flibe coolant channel shape variation. The OpenMC simulations are run with 80 active cycles, 20 inactive cycles, and 8000 particles resulting in an uncertainty between $\sim 100 - 150 \text{ ppcm}$. After the transport simulation is complete, a separate OpenMC file analyzes the OpenMC model's output files and calculates the k_{eff} constraint and the normalized power peaking factor (PPF_{fuel}) objective value. Section 5.3.3 describes the calculation for each output and constraint value.

AHTR Moltres Group Constant Generation

Unlike the OpenMC model, Moltres does not explicitly model each TRISO particle. This is because a TRISO-level fidelity mesh file is impractical for this simulation and results in intractable runtimes. Instead, Moltres relies on the OpenMC model to generate group constant data for the Moltres' multigroup neutron diffusion calculations. Previously, Moltres could only generate group constant data from Serpent [40] or SCALE [90] output files. I implemented functionality in Moltres for group constant data generation with OpenMC [89].

To enable successful Moltres AHTR temperature model simulations, I must establish suitable spatial and energy homogenization for group constant generation. These homogenizations must preserve accuracy while maintaining an acceptable runtime. I compared the key neutronics param-

eters (k_{eff} , reactivity coefficients, flux distribution, neutron energy spectrum) for the continuous OpenMC and multigroup Moltres simulations (that utilized OpenMC generated group constants) for both the AHTR plank and one-third assembly to ensure that the selected spatial and energy homogenizations preserve accuracy. Section 5.4 reports the verification study’s results. Next, I describe the energy and spatial homogenizations used.

For both the AHTR plank and one-third assembly models, I used eight precursor groups and a 4-group energy structure derived by Gentry et al. [39] for AHTR geometries. Table 5.3 defines the 4-group energy boundaries.

Table 5.3: 4-group energy structures for Advanced High-Temperature Reactor (AHTR) geometry derived by Gentry et al. [39].

Group Boundaries [MeV]		
Group #	Upper Bound	Lower Bound
1	2.0000×10^1	9.1188×10^{-3}
2	9.1188×10^{-3}	2.9023×10^{-5}
3	2.9023×10^{-5}	1.8554×10^{-6}
4	1.8554×10^{-6}	1.0000×10^{-12}

For spatial homogenization of the straightened AHTR plank and one-third assembly, I used OpenMC’s cell domain type to compute multigroup cross sections for different cells. For the AHTR plank, I discretized the plank into 13 cells: FLiBe, left graphite, right graphite, and ten fuel cells (where each cell has a different packing fraction and thus a different material definition). Figure 5.9 illustrates the AHTR plank’s spatial homogenization for the OpenMC multigroup calculation for Moltres group constant generation.

For the AHTR one-third assembly, I discretized the one-third assembly into 70 *cells*: inter-assembly FLiBe, Y-shaped graphite structure, control rod slot FLiBe, inter-plank FLiBe, each graphite plank (6), and ten fuel cells per plank (60). Figure 5.10 illustrates the AHTR one-third assembly’s spatial homogenization for the OpenMC multigroup calculation for group constant generation.

AHTR Moltres Mesh Generation

I wrote Python scripts for the AHTR plank and one-third assembly models that accept the coolant channel shape input parameters ($r_{top}, r_{bot}, r_1, r_2, r_3, r_4, r_5$) to generate a geometry script file (.geo)

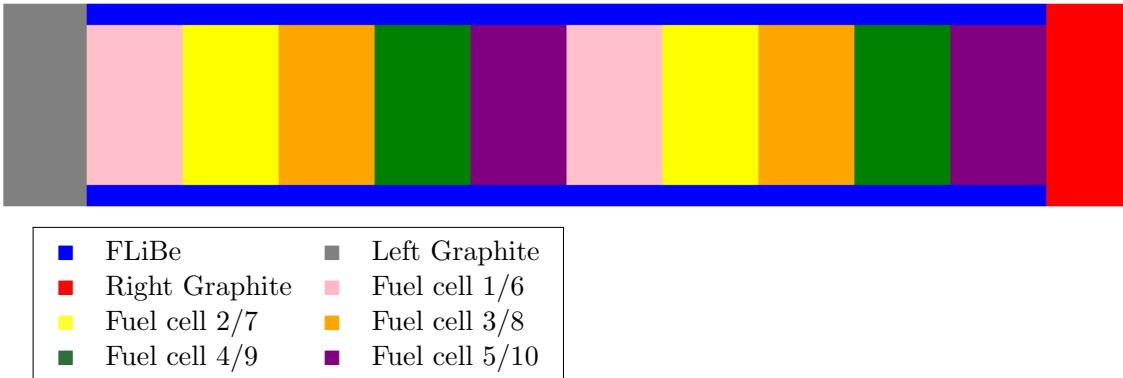


Figure 5.9: Straightened Advanced High-Temperature Reactor (AHTR) fuel plank spatially discretized into 13 *cells* for OpenMC multigroup calculation to produce group constants data for the Moltres model. 13 *cells*: FLiBe, left graphite, right graphite, and ten fuel cells (each cell has a different packing fraction).

based on the spatial homogenizations outlined in the previous subsection. The AHTR geometry mesh is then generated from the geometry script file using Gmsh [91]. I used Gmsh’s *refine by splitting* functionality to refine the mesh. Figures 5.11a and 5.11b show Gmsh rendered geometry file examples generated by the geometry scripts for the AHTR plank and one-third assembly, respectively.

AHTR Moltres Temperature Model

The Moltres AHTR steady-state temperature models are a 2D x-y cross-section model of the AHTR plank and one-third assembly. The plank and one-third assembly spatially homogenized geometries are depicted in Figure 5.9 and Figure 5.10. Both Moltres temperature models first solve the AHTR neutronics and use the outputs to solve the AHTR’s temperature distribution for a defined power. The temperature models assume conductive heat transfer throughout the domain and heat removal by uniform salt flow in the coolant region. These assumptions ignore turbulent effects that would most likely be present. However, in-depth AHTR Moltres models that includes turbulence modeling is a fruitful avenue for future work beyond this dissertation.

Moltres solves the 4-group diffusion equations (Equation 2.1) as a steady-state eigenvalue problem to find k_{eff} for the static AHTR models. In the 2D cross-sectional AHTR steady-state temperature models, I ignore the time-dependent and velocity-dependent terms from Moltres’ tem-

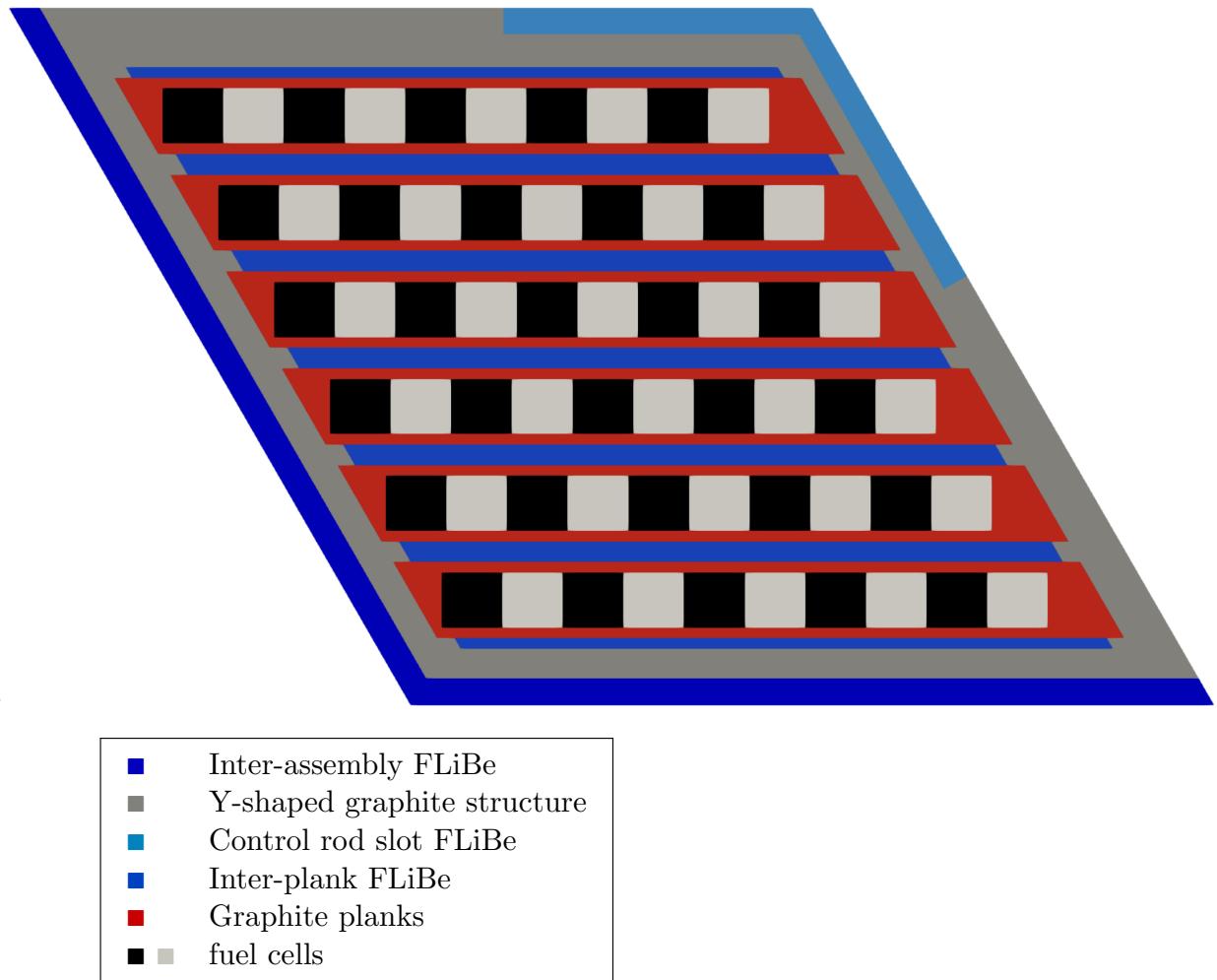
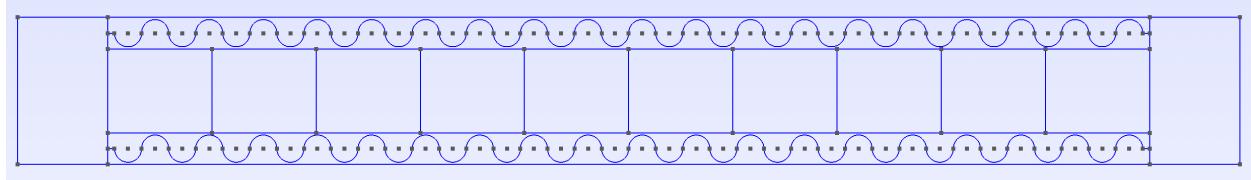
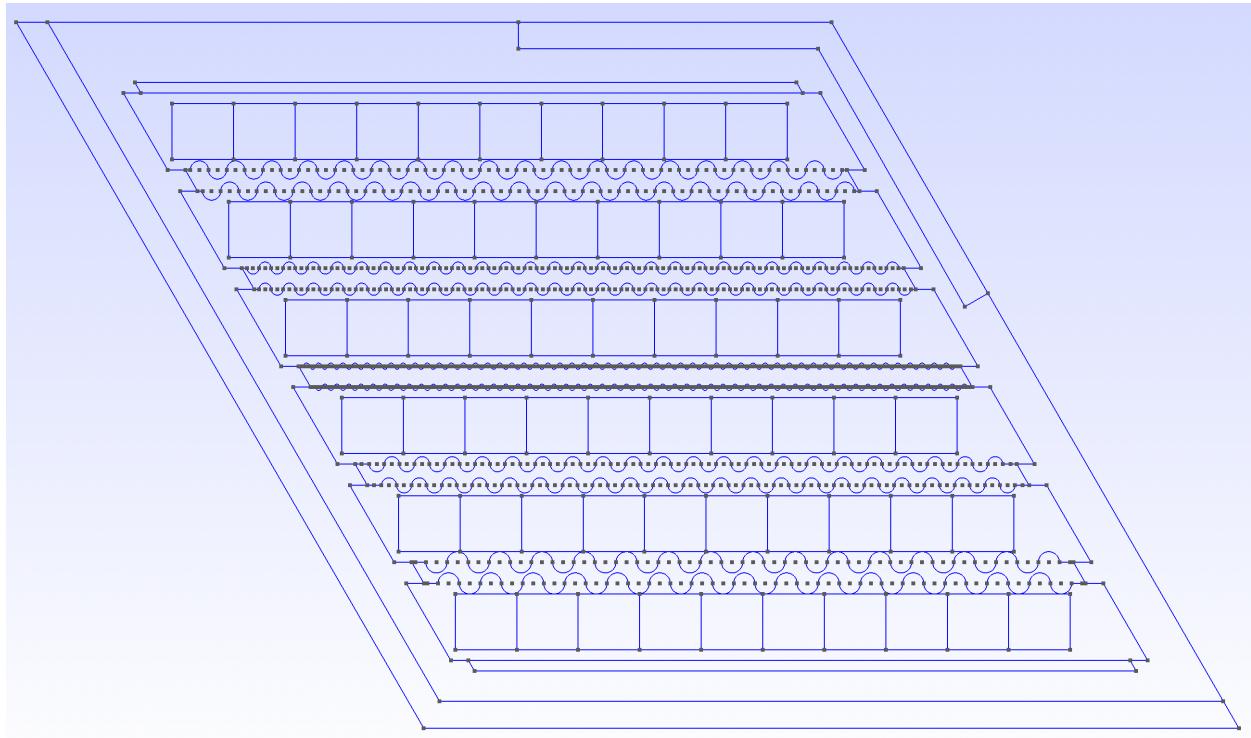


Figure 5.10: Advanced High-Temperature Reactor (AHTR) one-third assembly spatially discretized into 70 *cells* for OpenMC multigroup calculation to produce group constants data for Moltres model. 70 *cells*: inter-assembly FLiBe, Y-shaped graphite structure, control rod slot FLiBe, inter-plank FLiBe, each graphite plank (6), and ten fuel cells per plank (60).



(a) Advanced High-Temperature Reactor (AHTR) plank geometry file



(b) Advanced High-Temperature Reactor (AHTR) one-third assembly geometry file

Figure 5.11: Examples of AHTR plank (Figure 5.11a) and one-third assembly (Figure 5.11b) Gmsh rendered geometry files generated by the geometry scripts. These geometry files are meshed using Gmsh, and the mesh files are used in the Moltres temperature models.

perature governing equation, shown in Equation 5.3, as they are steady-state models with no moving fuel, resulting in Equation 5.4.

$$\rho c_p \frac{\partial T}{\partial t} = -\rho c_p \vec{u} \cdot \nabla T + \nabla \cdot (k_i \nabla T) + Q_f \quad (5.3)$$

$$-\nabla \cdot (k_i \nabla T) = Q_i \quad (5.4)$$

where

k_i = thermal conductivity of material i [$Wcm^{-1}K^{-1}$]

T = temperature in the model [K]

Q_i = source or sink term in material i [Wcm^{-2}]

I use insulated temperature boundary conditions, which mimics an insulator at the boundary. Table 5.4 shows the thermal conductivity values used for each AHTR material.

Table 5.4: Advanced High-Temperature Reactor (AHTR) materials' thermal conductivities used in Moltres temperature models, taken from [92].

Material	Thermal Conductivity [$Wcm^{-1}K^{-1}$]
FLiBe	0.01
Graphite	0.15
Fuel	0.099

Equation 5.5 defines the fuel cells' fission source term (Q_f):

$$Q_f = \sum_{g=1}^G \epsilon_{f,g} \Sigma_{f,g} \phi_g \quad (5.5)$$

where

Q_f = source term [Wcm^{-3}]

G = number of discrete groups, g [-]

$\epsilon_{f,g}$ = heat produced per fission in group g [J]

$\Sigma_{f,g}$ = macroscopic cross section for fission due to neutrons in group g [cm^{-1}]

ϕ_g = flux of neutrons in group g [$n \cdot cm^{-2} s^{-1}$]

Equation 5.6 defines the heat removal from the AHTR models:

$$Q = h \cdot A \cdot (T(\vec{r}) - T_{ref}) \quad (5.6)$$

where

Q = heat removal rate for 1cm thin slice of the AHTR model [W]

h = heat transfer coefficient [$W cm^{-2} K^{-1}$]

A = coolant area [cm^2]

$T(r)$ = temperature at point r [K]

T_{ref} = reference temperature [K]

Table 5.5 shows reference temperature and heat transfer coefficient values for the convective heat transfer process.

Table 5.5: Advanced High-Temperature Reactor (AHTR) plank and one-third assembly heat transfer constants.

Model	Constant	Value	Units	Notes
Plank	h_{plank}	990	$W cm^{-2} K^{-1}$	Calculated in Eq. 5.8
One-Third Assembly	h_{assem}	611	$W cm^{-2} K^{-1}$	Calculated in Eq. 5.9
Both	T_{ref}	923	K	AHTR Inlet Temperature [92]

I calculated the heat transfer coefficient (h) for the AHTR plank and one-third assembly using Equation 5.8, 5.9 with the following assumptions:

- the AHTR modeled generates a constant amount of power, which is all removed by the coolant
- a linear increase in temperature from inlet to outlet

The linear temperature change is given by:

$$\Delta T = \frac{T_{total}}{H} = \frac{50 \text{ K}}{550 \text{ cm}} \times 1 \text{ cm} = 0.0909 \text{ K} \quad (5.7)$$

For the plank, h is calculated by:

$$\begin{aligned} h_{plank} &= \frac{P_{dz}}{V_{coolant} \cdot \Delta T} \\ &= \frac{1456 \text{ W}}{(23.1 \text{ cm} \times 0.35 \text{ cm} \times 2) \cdot 0.0909 \text{ K}} \\ &= 990 \text{ W cm}^{-2} \text{ K}^{-1} \end{aligned} \quad (5.8)$$

and for the one-third assembly:

$$\begin{aligned} h_{assem} &= \frac{P_{dz}}{V_{coolant} \cdot \Delta T} \\ &= \frac{8741 \text{ W}}{157 \text{ cm}^2 \cdot 0.0909 \text{ K}} \\ &= 611 \text{ W cm}^{-1} \text{ K}^{-1} \end{aligned} \quad (5.9)$$

The relevant parameters for each equation are:

h_{plank} = plank's heat transfer coefficient [$\text{W cm}^{-2} \text{ K}^{-1}$]

h_{assem} = one-third assembly's heat transfer coefficient [$\text{W cm}^{-2} \text{ K}^{-1}$]

P_{dz} = power produced in 1cm AHTR plank/one-third assem Δz slice [W]

$A_{coolant}$ = coolant area in AHTR plank/one-third assem [cm^2]

ΔT = temperature change across 1cm AHTR Δz slice [K]

T_{total} = total temperature change from inlet to outlet [K]

H = AHTR height from inlet to outlet [cm]

The power produced by the AHTR plank and one-third models are calculated based on the FHR benchmark model's specific power of 200 W gU^{-1} and the FHR benchmark's TRISO packing frac-

tions in the plank and one-third assembly. During the optimization simulations in which I vary the PF_{total} , I assume that power remains constant.

In the ROLLO optimization simulations, I vary the FLiBe coolant channel shape. During the coolant channel shape variation, I hold the coolant volume constant. Since the coolant volume is held constant throughout the coolant channel optimization process, I use the same heat transfer coefficient (h) for all the AHTR Moltres temperature models of varying coolant channel shapes.

5.3.3 Output Parameter Calculation

This section describes how I tallied the AHTR model outputs for the ROLLO optimization problem objectives (described in Table 5.1); that is, the total fuel packing fraction (PF_{total}), the maximum temperature (T_{max}), and the fuel-normalized power peaking factor (PPF_{fuel}).

ROLLO will automatically return the PF_{total} output parameter since it is also an input parameter. In the Moltres AHTR model, I defined a post processor object to return T_{max} . The PPF_{fuel} output parameter takes into account fuel density variations across the model. For the AHTR plank, and each fuel plank in the AHTR one-third fuel assembly, I discretized the fuel cell area of the plank into 10×5 blocks. I then use OpenMC to tally the fission energy production rate (`fission-q-recoverable`) in each section. `fission-q-recoverable`'s units are eV per source particle. Because the final PPF_{fuel} is a ratio, I did not normalize the score to calculate power. The fuel-normalized power peaking factor is given by Equation 5.10:

$$PPF_{fuel} = \max\left(\frac{fqr_j}{PF_j}\right) \div \text{avg}\left(\frac{fqr_j}{PF_j}\right) \quad (5.10)$$

where

PPF_{fuel} = fuel-normalized power peaking factor [-]

j = discretized fuel area j [-]

fqr_j = fission-q-recoverable at position j [$eVsrc^{-1}$]

PF_j = fuel packing fraction at position j [-]

5.4 AHTR Moltres Model Verification

This section verifies Moltres' ability to reproduce key neutronics parameters using group constant data from OpenMC, which is essential for accurate neutronics calculations in the Moltres temperature models. I set up a criticality eigenvalue problem in Moltres and calculate key neutronics parameters using group constant data from OpenMC. For both the AHTR plank and one-third assembly models, I compare the key neutronics parameters between two simulations:

1. OpenMC simulation with continuous energy and TRISO-level spatial fidelity
2. Moltres simulation with 4-group energy and spatial homogenization

Section 5.3.2 outlined the spatial homogenization used. The OpenMC simulation with TRISO-level fidelity generates the group constants for the energy and spatially homogenized Moltres simulation.

I compare the following key neutronics parameters for both the AHTR plank and one-third assembly models: effective multiplication factor, reactivity coefficients, flux distribution, and neutron energy spectrum. The comparisons are to verify that the Moltres model is replicating the OpenMC model's neutronics correctly. And of these, comparisons of the reactivity coefficients and flux distributions are key to ensuring that Moltres accurately calculates the AHTR's temperature distribution. The reactivity coefficients capture temperature reactivity feedback on the flux when the temperature varies with space in the Moltres model. The heat produced per fission, $\epsilon_{f,g}$, and macroscopic cross section for fission, $\Sigma_{f,g}$, terms in the Moltres source term (Equation 5.5) are provided to Moltres through the group constants generated by the transport software, OpenMC. Thus, differences in the source term between OpenMC and Moltres are dependent on the flux.

5.4.1 AHTR Plank: Key Neutronics Parameters Verification

I compare the following key neutronics parameters: effective multiplication factor, reactivity coefficients, flux distribution, and neutron energy spectrum for the AHTR plank model. For this verification study, I used an AHTR plank model with $PF_{total} = 0.0979$ and

$\rho_{TRISO}(\bar{x}) = (1.989 \cdot \sin(0.354 \cdot x + 3.143) + 2) \cdot NF$. Section 5.5.1 found an optimized hyperparameter set for ROLLO single-objective optimization simulations. I used the optimized hyperparameter set in a ROLLO simulation to maximize the k_{eff} in a AHTR plank model. Then, I used the AHTR

plank model with the largest k_{eff} in the optimization's final generation for this verification study. Since the same materials, uranium isotope, and general plank structure is used for the AHTR plank's optimization, conducting verification for one geometry is acceptable. Figure 5.12 shows the AHTR plank model with TRISO-level fidelity.

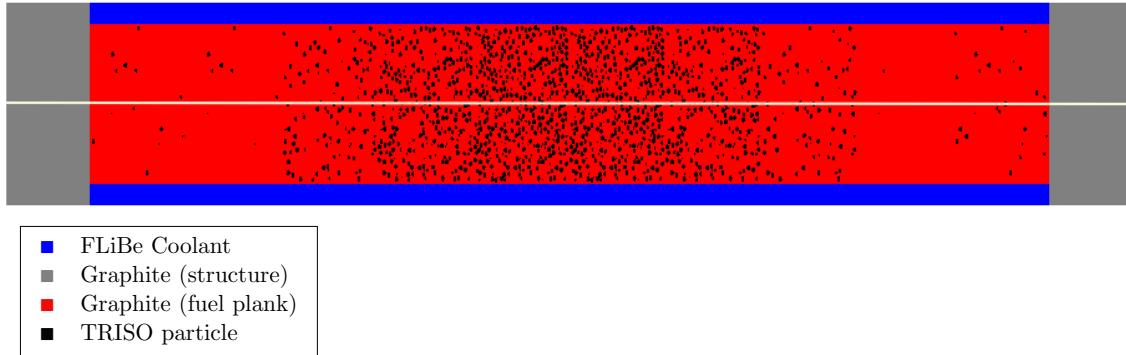


Figure 5.12: Advanced High-Temperature Reactor (AHTR) plank geometry and packing fraction distribution used for the AHTR plank's key neutronics parameters verification. $PF_{total} = 0.0979$ and $\rho_{TRISO}(\bar{x}) = (1.989 \cdot \sin(0.354 \cdot x + 3.143) + 2) \cdot NF$. The white line corresponds to the centerline where flux distribution is measured.

AHTR Plank: Effective Multiplication Factor

Comparing k_{eff} and reactivity produced by Moltres and OpenMC verify that the Moltres model replicates the OpenMC model's neutronics correctly. Table 5.6 compares the effective multiplication factor (k_{eff}) and reactivity (ρ) for the OpenMC simulation with continuous energy and TRISO-level spatial fidelity, the OpenMC simulation with 4-group energy and spatial homogenization, and the Moltres simulation with 4-group energy and spatial homogenization. I included results from the homogenized OpenMC simulation to distinguish between differences caused by spatial homogenization and energy discretization, or differing OpenMC and Moltres solve methods.

The 71pcm k_{eff} and 35pcm reactivity difference, that can be observed in Table 5.6, between continuous and homogenized OpenMC simulations are within uncertainty. This shows that the selected spatial homogenizations and energy discretizations are acceptable. However, the Moltres simulation shows a 706pcm difference in k_{eff} and 355pcm difference in reactivity. The summary at the end of this section explains the differences in these values.

Table 5.6: The Advanced High-Temperature Reactor (AHTR) plank's k_{eff} and reactivity values from the OpenMC simulation with continuous energy and TRISO-level spatial fidelity, the OpenMC simulation with 4-group energy and spatial homogenization, and the Moltres simulation with 4-group energy and spatial homogenization. All simulations are at 948K. Reported differences are with the OpenMC simulation with continuous energy and TRISO-level spatial fidelity (no homogenization).

Software	Homogenized?	k_{eff}	Diff [pcm]	Reactivity [pcm]	Reactivity Diff [pcm]
OpenMC	No	1.41402 ± 0.00140	-	29279 ± 70	-
OpenMC	Yes	1.41473 ± 0.00098	+71	29314 ± 49	+35
Moltres	Yes	1.40696	-706	28924	-355

AHTR Plank: Reactivity Coefficients

A comparison of reactivity coefficients produced by Moltres and OpenMC verifies that the Moltres model is replicating the OpenMC model's neutronics correctly and are also important to ensure that Moltres accurately calculates the temperature distribution. Moltres' delayed neutron fraction, β_{eff} , is calculated by taking the normalized difference between k_{eff} values with and without delayed neutron precursors (DNPs). Table 5.7 shows that the β_{eff} values from OpenMC and Moltres show excellent agreement with a discrepancy of 0.03pcm. I calculated the temperature reactivity coefficients using Equation 3.1. Table 5.7 also shows that the total temperature coefficients from OpenMC and Moltres have good agreement with a discrepancy of $0.23 \text{ pcm} \cdot K^{-1}$.

Table 5.7: Advanced High-Temperature Reactor (AHTR) fuel plank's β_{eff} values from OpenMC and Moltres simulations at 948K, and total reactivity coefficient values calculated from OpenMC and Moltres at 948K and 1100K. The OpenMC simulation has continuous energy and TRISO-level spatial fidelity and Moltres simulation has 4-group energy and spatial homogenization.

Software	Homogenized?	β_{eff} [pcm]	Diff [pcm]	Total $\frac{\Delta\rho}{\Delta T}$ [pcm $\cdot K^{-1}$]	Diff [pcm $\cdot K^{-1}$]
OpenMC	No	654.31	-	-4.26	-
Moltres	Yes	654.28	-0.03	-4.49	-0.23

AHTR Plank: Flux Distribution

A comparison of flux distributions produced by Moltres and OpenMC verifies that the Moltres model is replicating the OpenMC model's neutronics correctly and are also important to ensure that Moltres accurately calculates the temperature distribution. Figure 5.13 shows the 4-group spatial flux distributions for OpenMC and Moltres on the AHTR plank's x-axis centerline, at the

y-axis' midpoint (white line on Figure 5.12). Table 5.8 reports the 2-norm percentage difference

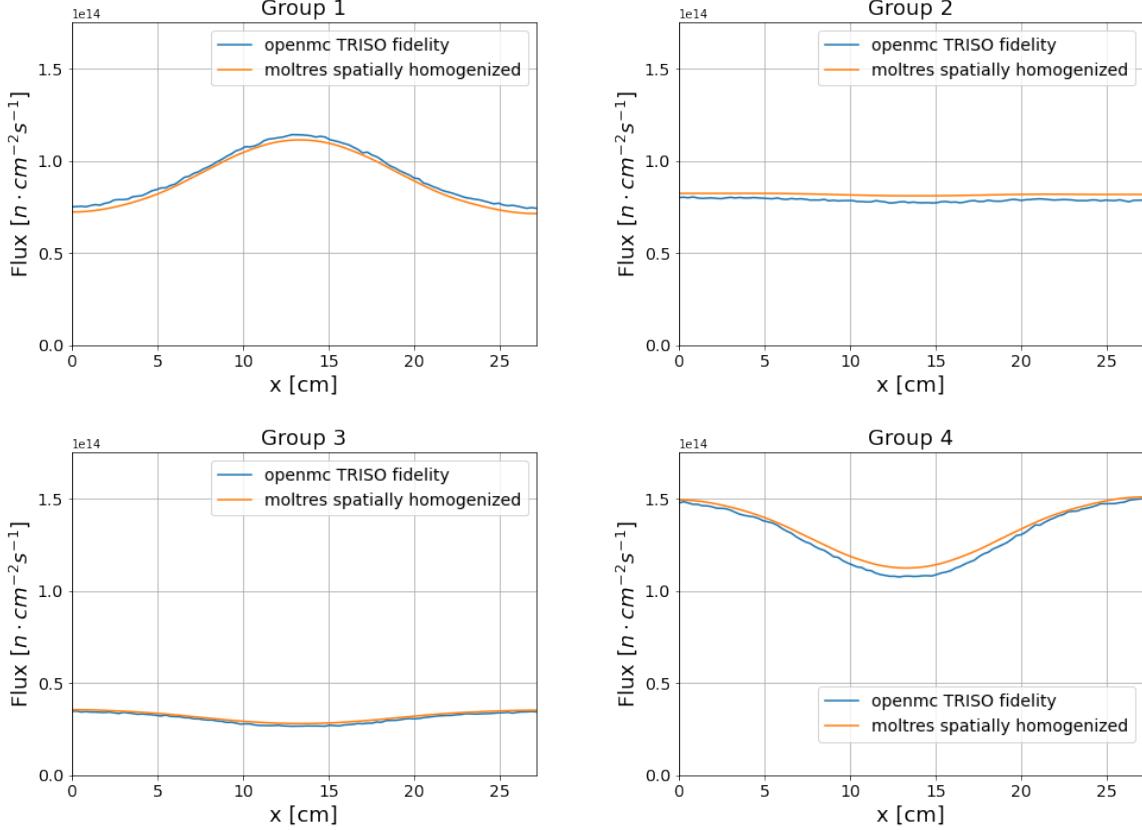


Figure 5.13: Advanced High-Temperature Reactor (AHTR) plank's centerline neutron flux distribution in 4 groups at 948K. Centerline is the white line in Figure 5.12. Comparison is between OpenMC simulation with continuous energy and TRISO-level spatial fidelity and Moltres simulation with 4-group energy and spatial homogenization. Energy Group 1: $E > 9.1188 \times 10^{-3}$ MeV, Energy Group 2: $2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$ MeV, Energy Group 3: $1.8556 \times 10^{-5} < E < 2.9023 \times 10^{-5}$ MeV, Energy Group 4: $1.0 \times 10^{-12} < E < 1.8554 \times 10^{-6}$ MeV.

(given by Equation 5.11) and maximum percentage difference between centerline flux values from OpenMC and Moltres models.

$$\|\Delta\phi\|_N = \frac{1}{N} \sqrt{\sum_{i=1}^N \left(\frac{\phi_{moltres,i} - \phi_{openmc,i}}{\phi_{openmc,i}} \times 100 \right)^2} \quad (5.11)$$

where

$\|\Delta\phi\|_N$ = normalized 2-norm flux percentage difference between Moltres and OpenMC [%]

N = total number of discretized points [#]

i = discretized point [-]

$\phi_{moltres}$ = Moltres model's centerline flux [$n \cdot cm^{-2}s^{-1}$]

ϕ_{openmc} = OpenMC model's centerline flux [$n \cdot cm^{-2}s^{-1}$]

Table 5.8: Advanced High-Temperature Reactor (AHTR) plank's centerline normalized 2-norm of flux percentage difference and maximum flux percentage difference. Centerline is the white line in Figure 5.12. The difference values are calculated from comparison between the OpenMC simulation with continuous energy and TRISO-level spatial fidelity and Moltres simulation with 4-group energy and spatial homogenization.

Energy Group	2-norm Diff [%]	Max Diff [%]
1	0.28	-4.90
2	0.39	+5.17
3	0.40	+6.82
4	0.27	+5.22

Figure 5.13 shows that the OpenMC model has higher flux in Group 1 and lower flux in Group 2 and 4 compared to the Moltres model. Table 5.8 shows that the 2-norm percentage differences between OpenMC and Moltres models' flux values are slight, demonstrating that there is a good overall agreement for each group's flux. The summary at the end of this section explains the differences in the flux's maximum percentage differences between the OpenMC and Moltres models.

AHTR Plank: Neutron Energy Spectrum

A comparison of the neutron energy spectrum verifies that the Moltres model is replicating the OpenMC model's neutronics correctly. Figure 5.14 shows the neutron energy spectrum of the OpenMC simulation for both 252 and 4 groups and the 4-group Moltres simulation. There is good agreement between OpenMC and Moltres with their 4-group spectrums.

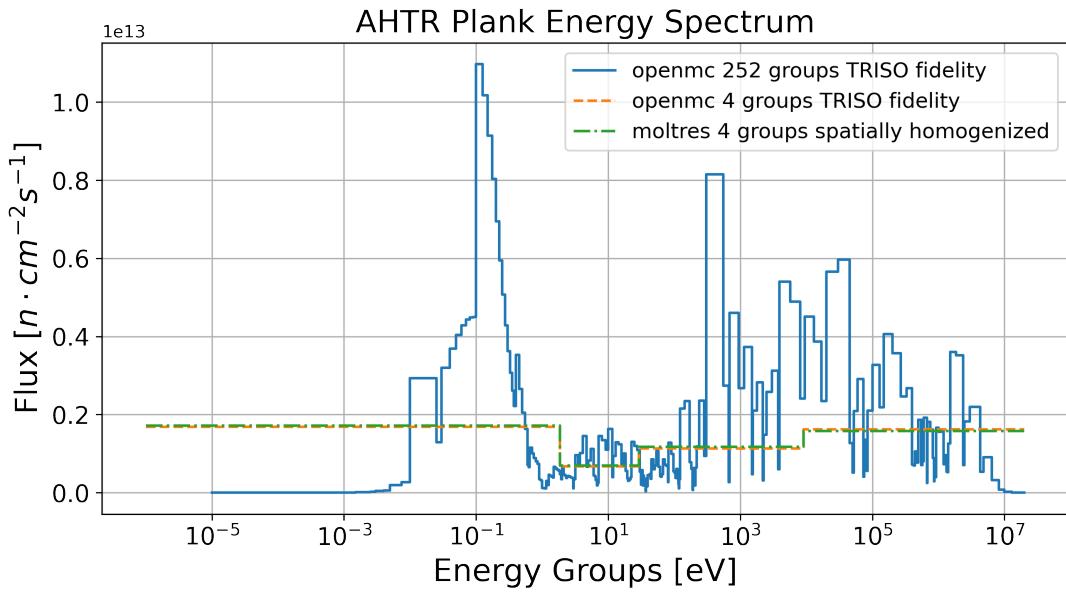


Figure 5.14: Advanced High-Temperature Reactor (AHTR) plank’s neutron spectrum. Spectrums include 252 and 4 group spectrums from OpenMC simulation with continuous energy and TRISO-level spatial fidelity and 4-group spectrum from Moltres simulation with 4-group energy and spatial homogenization.

AHTR Plank: Key Neutronics Parameters Verification Summary

The verification study found a 355 pcm reactivity difference between OpenMC and Moltres models and good agreement in their reactivity coefficients and 4-group neutron energy spectrum. The verification study also found good agreement in overall flux distributions (based on the 2-norm difference); however, there were larger flux differences at specific points. The reactivity and flux differences are due to Moltres utilizing the neutron diffusion method instead of neutron transport methods, resulting in flux not being reproduced well in small regions which are less than a few mean free paths in length (i.e., the FLiBe coolant channel). The differences in reactivity and flux at specific points might result in a slightly inaccurate absolute temperature distribution. However, since the reactivity coefficients and overall flux distribution are in agreement, Moltres accurately captures the relative temperature distributions. During the AHTR optimization, so long the relative maximum temperature between different AHTR models is captured correctly, ROLLO can find optimal reactor geometries accurately. Reactor designers can then use higher fidelity software for modeling the final optimal AHTR geometry. In summary, Moltres replicated the relevant neu-

tronics parameters with sufficient accuracy using OpenMC's group constant data for the AHTR plank model.

5.4.2 AHTR One-Third Assembly: Key Neutronics Parameters Verification

Key neutronics parameter verification is also necessary for the one-third assembly model since it has different ratios of materials compared to the plank model. However, successful AHTR plank model key neutronics parameter verification gives confidence for the one-third assembly model verification study. I compare the following key neutronics parameters: effective multiplication factor, reactivity coefficients, flux distribution, and neutron energy spectrum for the AHTR one-third assembly model. For this verification study, I used an AHTR one-third assembly model with a constant 0.0979 total packing fraction across all fuel cells. Since the same materials, uranium isotope, and general one-third assembly model structure is used for the AHTR one-third assembly's optimization, conducting verification for one geometry is acceptable. Figure 5.15 shows the AHTR one-third assembly model with TRISO-level fidelity.

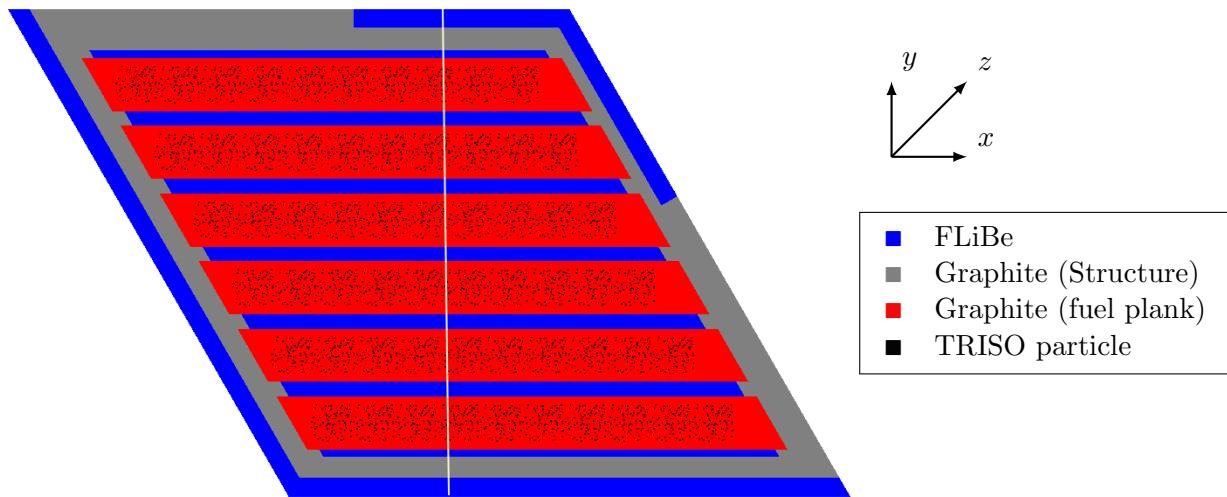


Figure 5.15: Advanced High-Temperature Reactor (AHTR) one-third assembly geometry with a constant 0.0979 total packing fraction across all fuel cells used for AHTR one-third assembly's key neutronics parameters verification. The white line corresponds to the centerline where flux distribution is measured.

AHTR One-Third Assembly: Effective Multiplication Factor

Comparing k_{eff} and reactivity produced by Moltres and OpenMC verify that the Moltres model replicates the OpenMC model's neutronics correctly. Table 5.9 compares the effective multiplication factor for OpenMC simulation with continuous energy and TRISO-level spatial fidelity, OpenMC simulation with 4-group energy and spatial homogenization, and Moltres simulation with 4-group energy and spatial homogenization. I included results from a homogenized OpenMC simulation to distinguish between differences caused by spatial homogenization and energy discretization, or differing OpenMC and Moltres solve methods.

Table 5.9: Advanced High-Temperature Reactor (AHTR) one-third assembly's k_{eff} values from the OpenMC simulation with continuous energy and TRISO-level spatial fidelity, OpenMC simulation with 4-group energy and spatial homogenization, and Moltres simulation with 4-group energy and spatial homogenization. All simulations are at 948K. The normalized difference is the pcm difference normalized by OpenMC non-homogenized model's k_{eff} .

Software	Homogenized?	k_{eff}	Diff [pcm]	Reactivity [pcm]	Reactivity Diff [pcm]
OpenMC	No	1.41657 ± 0.00131	-	29406 ± 65	-
OpenMC	Yes	1.41670 ± 0.00116	+13	29413 ± 58	+7
Moltres	Yes	1.40895	-762	29025	-381

The 13pcm k_{eff} and 7pcm reactivity difference between continuous and homogenized OpenMC simulations is within uncertainty, showing that the selected spatial homogenizations and energy discretizations are acceptable. However, the Moltres simulation shows a 762pcm difference in k_{eff} and 381pcm difference in reactivity, similar to the AHTR plank model's difference in Section 5.4.1. The summary at the end of this section explains the differences in these values.

AHTR One-Third Assembly: Reactivity Coefficients

A comparison of reactivity coefficients produced by Moltres and OpenMC verifies that the Moltres model is replicating the OpenMC model's neutronics correctly and are also important to ensure that Moltres accurately calculates the temperature distribution. Moltres' delayed neutron fraction, β_{eff} , is calculated by taking the normalized difference between k_{eff} values with and without DNP. The β_{eff} values from OpenMC and Moltres in Table 5.10 show excellent agreement with a discrepancy of 0.8pcm. I calculated the temperature reactivity coefficients with Equation 3.1. Table 5.10 shows that the total temperature coefficients from OpenMC and Moltres have good

agreement with a discrepancy of 0.2 pcm/K.

Table 5.10: Advanced High-Temperature Reactor (AHTR) one-third assembly's β_{eff} values from OpenMC and Moltres at 948K, and total reactivity coefficient values calculated from OpenMC and Moltres at 948K and 1100K, respectively. The OpenMC simulation has continuous energy and TRISO-level spatial fidelity. The Moltres simulation has 4-group energy and spatial homogenization.

Software	Homogenized?	β_{eff} [pcm]	Diff [pcm]	Total $\frac{\Delta\rho}{\Delta T}$ [pcm · K ⁻¹]	Diff [pcm · K ⁻¹]
OpenMC	No	652.3	-	-3.64	-
Moltres	Yes	651.5	-0.8	-3.44	+0.2

AHTR One-Third Assembly: Flux Distribution

A comparison of flux distributions produced by Moltres and OpenMC verifies that the Moltres model is replicating the OpenMC model's neutronics correctly and are also important to ensure that Moltres accurately calculates the temperature distribution. Figure 5.16 shows the 4-group flux distributions for OpenMC and Moltres on the AHTR one-third assembly's centerline, along the y-axis at the x-axis' midpoint (white line on Figure 5.15). Table 5.11 reports the 2-norm percentage difference (Equation 5.11) and maximum percentage difference between centerline flux values from OpenMC and Moltres models.

Table 5.11: Advanced High-Temperature Reactor (AHTR) one-third assembly's centerline 2-norm flux percentage difference and maximum flux percentage difference. Centerline is the white line in Figure 5.15. The difference values are calculated from comparison between the OpenMC simulation with continuous energy and TRISO-level spatial fidelity and the Moltres simulation with 4-group energy and spatial homogenization.

Energy Group	2-norm Diff [%]	Max Diff [%]
1	0.15	-5.36
2	0.29	+7.13
3	0.33	+11.66
4	0.26	+5.82

Figure 5.16 shows that the OpenMC model has lower flux in Group 2 and 4 compared to the Moltres model. Table 5.8 shows that the 2-norm percentage differences between OpenMC and Moltres models' flux values are slight, demonstrating that there is a good overall agreement for each group's flux. The summary at the end of this section explains the differences in the flux's maximum percentage differences between the OpenMC and Moltres models.

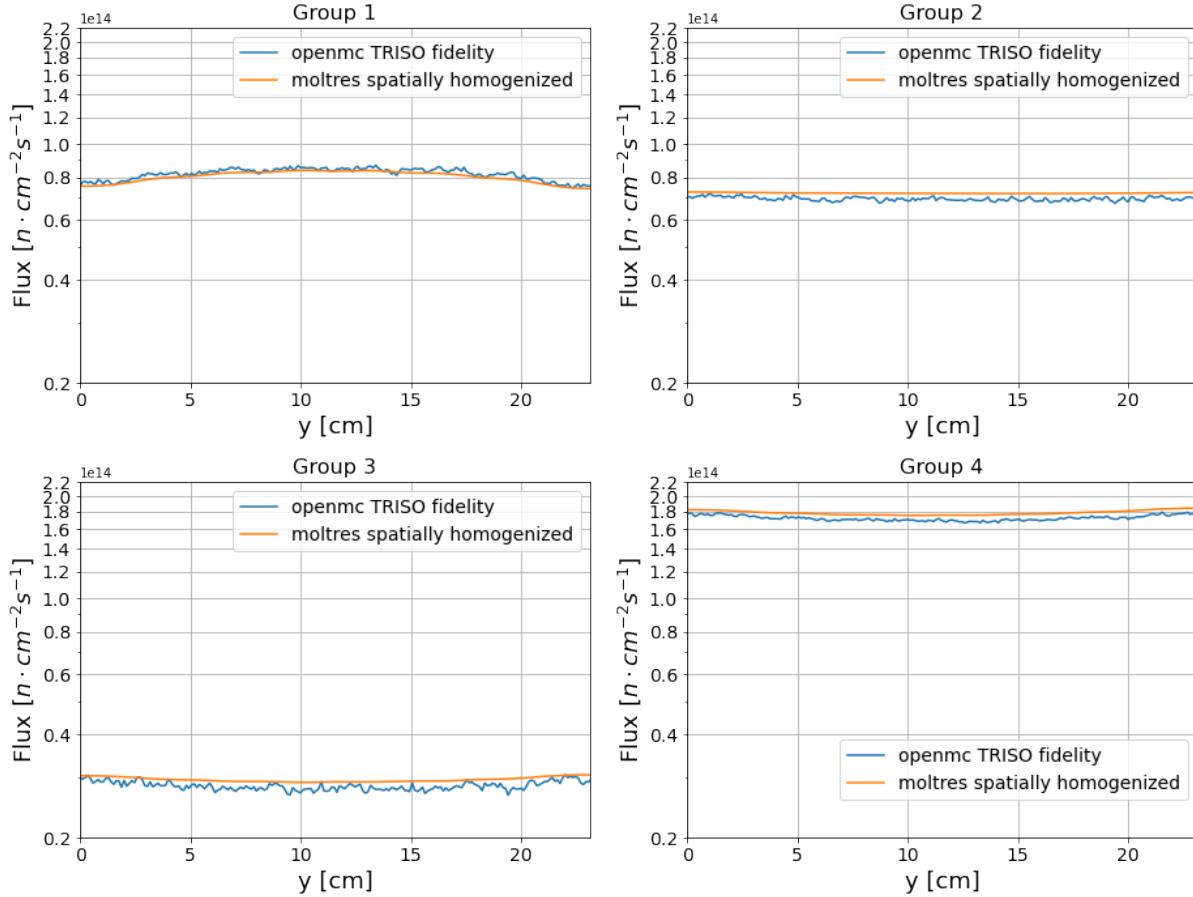


Figure 5.16: Advanced High-Temperature Reactor (AHTR) one-third assembly's centerline neutron flux distribution in 4 groups at 948K. Centerline is the white line in Figure 5.15. The comparison is between the OpenMC simulation with continuous energy and TRISO-level spatial fidelity and the Moltres simulation with 4-group energy and spatial homogenization. Energy Group 1: $E > 9.1188 \times 10^{-3}$ MeV, Energy Group 2: $2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$ MeV, Energy Group 3: $1.8556 \times 10^{-5} < E < 2.9023 \times 10^{-5}$ MeV, Energy Group 4: $1.0 \times 10^{-12} < E < 1.8554 \times 10^{-6}$ MeV.

AHTR One-Third Assembly: Neutron Energy Spectrum

A comparison of the neutron energy spectrum verifies that the Moltres model is replicating the OpenMC model's neutronics correctly. Figure 5.17 shows the one-third assembly's neutron spectrum of the OpenMC simulation for 252 and 4 groups and 4-group Moltres simulation. There is

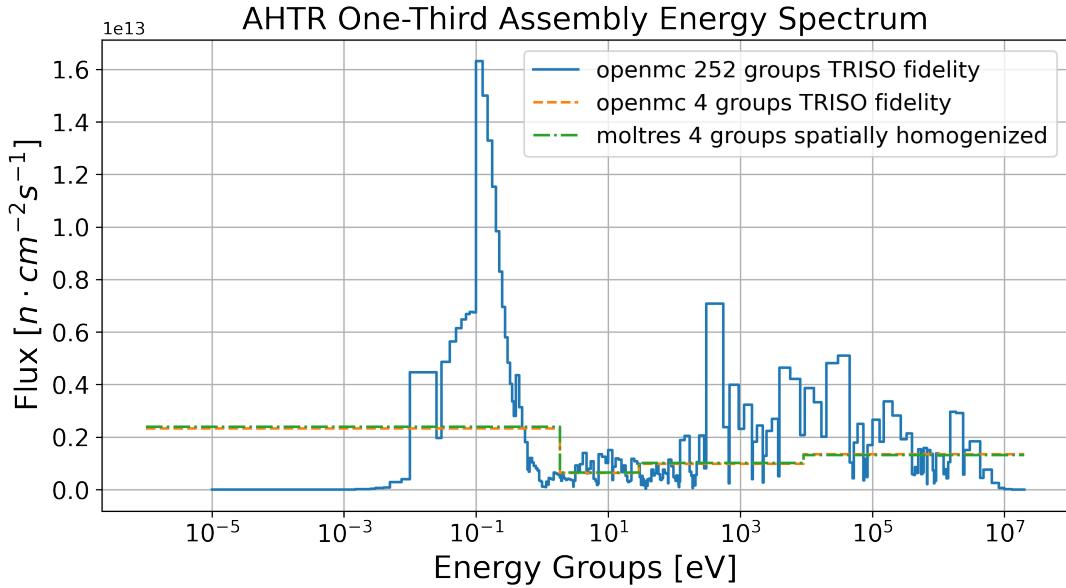


Figure 5.17: Advanced High-Temperature Reactor (AHTR) one-third assembly's neutron spectrum. Spectrums include 252 and 4 group spectrums from OpenMC simulation with continuous energy and TRISO-level spatial fidelity and 4 group spectrum from Moltres simulation with 4-group energy and spatial homogenization.

good agreement between OpenMC and Moltres models' 4-group spectrums.

AHTR One-Third Assembly: Key Neutronics Parameters Verification Summary

The one-third assembly verification study has similar results to the plank verification study (Section 5.4.1). The verification study found a 381 pcm reactivity difference between OpenMC and Moltres models and good agreement in their reactivity coefficients and 4-group neutron energy spectrum. The verification study also found good agreement in overall flux distributions (based on the 2-norm difference); however, there were larger flux differences at specific points. The reactivity and flux differences are due to Moltres utilizing the neutron diffusion method instead of neutron transport methods, resulting in flux not being reproduced well in small regions which are less than

a few mean free paths in length (i.e., the FLiBe coolant channel). The differences in reactivity and flux at specific points might result in a slightly inaccurate absolute temperature distribution. However, since the reactivity coefficients and overall flux distribution are in agreement, Moltres accurately captures the relative temperature distributions. During the AHTR optimization, so long the relative maximum temperature between different AHTR models is captured correctly, ROLLO can find optimal reactor geometries accurately. Reactor designers can then use higher fidelity software for modeling the final optimal AHTR geometry. In summary, Moltres replicated the relevant neutronics parameters with sufficient accuracy using OpenMC’s group constant data for the AHTR one-third assembly model.

5.4.3 AHTR Plank and One-Third Assembly: Mesh Refinement Studies

I performed mesh refinement studies on the AHTR plank and one-third assembly Moltres temperature models to ensure that their geometry mesh inputs are sufficiently converged. Tables 5.12 and 5.13 show the mesh refinement study results for the AHTR plank and one-third assembly. The mesh refinement studies report the average, maximum, and normalized 2-norm temperature difference between refinement steps. The 2-norm calculation is defined by Equation 5.12.

$$\|\Delta T_k\|_N = \frac{1}{N} \sqrt{\sum_{i=1}^N (T_{k-1,i} - T_{k,i})^2} \quad (5.12)$$

where

$\|\Delta T_k\|_N$ = normalized 2-norm temperature difference between refinement steps [K]

N = total number of discretized points [#]

i = discretized point [-]

k = refinement step [-]

T_{k-1} = temperatures from previous refinement step [K]

T_k = temperatures from current refinement step [K]

I used 100 discretized points (N) to calculate $\|\Delta T_k\|_N$ for all refinement steps.

Table 5.12: Advanced High-Temperature Reactor (AHTR) plank's Moltres temperature model mesh refinement study.

Refinement	Max Plank Temp [K]	Diff [K]	Avg Plank Temp [K]	Diff [K]	$\ \Delta T_k\ _N$ [K]
1	1126.219	-	1011.209	-	-
2	1127.711	+1.492	1017.630	+6.421	0.839
3	1128.434	+0.723	1020.021	+2.390	0.313

Table 5.13: Advanced High-Temperature Reactor (AHTR) one-third assembly's Moltres temperature model mesh refinement study.

Refinement	Max One-Third Assembly Temp [K]	Diff [K]	Avg One-Third Assembly Temp [K]	Diff [K]	$\ \Delta T_k\ _N$ [K]
1	1185.436	-	995.963	-	-
2	1186.045	+0.609	1001.751	+5.788	0.703
3	1185.994	-0.051	1003.625	+1.874	0.200

For each Moltres temperature model, I must balance convergence and computational cost, thus, I determine that mesh convergence is met if $\|\Delta T_k\|_N < 0.5K$. Both AHTR plank and one-third assembly show suitable $\|\Delta T_k\|_N$ convergence at x3 mesh refinement; thus I used x3 mesh refinement for all AHTR plank and one-third assembly Moltres temperature models. Further refinement is computationally unfeasible due to the large mesh size. I used the Theta supercomputer [107] for the optimization work. Each supercomputer node saves the entire mesh file, further refinement results in mesh sizes that are larger than each Theta node's memory size.

5.4.4 AHTR Plank and One-Third Assembly: Group Constant Temperature Study

During ROLLO optimization, each new reactor model results in the creation of a new Moltres temperature model. Correspondingly, a new set of group constant data needs to be created for each Moltres temperature model. Moltres relies on the OpenMC model to generate group constant data for the Moltres' multigroup neutron diffusion calculations. The Moltres model uses the group constant data by interpolating it for its required temperatures. Group constant data with multiple temperatures requires running multiple OpenMC neutronics simulations at the various temperatures, thus, increasing the total compute time required for each new reactor model. This section explores the effects of using multiple temperature group constant data compared to single temperature group constant data. I set up Moltres AHTR steady-state temperature models for the plank and one-third assembly with group constant data at each and all the of following temperatures:

948, 1024, 1100, and 1200K.

Tables 5.14 and 5.15 show the average and maximum temperature reported by the AHTR plank and one-third assembly Moltres models using group constant data from one temperature and their differences compared to the model with group constant data with all four temperatures. Both tables also show the normalized 2-norm of the centerline temperature difference between the AHTR models with single temperatures and the AHTR model with all four temperature (Equation 5.13):

$$\|\Delta T\|_N = \frac{1}{N} \sqrt{\sum_{i=1}^N (T_{all,i} - T_i)^2} \quad (5.13)$$

where

$\|\Delta T\|_N$ = normalized 2-norm temperature difference [K]

N = total number of discretized points [#]

i = discretized point [-]

T_{all} = temperatures from model with group constant data at all four temperatures [K]

T = temperatures from model with group constant data at one temperature [K]

The centerlines where the temperature (T) is measured in the model for the AHTR plank and one-third models are depicted in Figures 5.12 and 5.15, respectively.

Table 5.14: Advanced High-Temperature Reactor (AHTR) plank's average and maximum temperature and normalized 2-norm of the temperature difference across the plank's centerline for varying group constant temperature data. The difference values are calculated from comparison against the Moltres model using group constant data with all four temperatures (All).

Group Constant Data Temps [K]	Avg Plank Temp [K]	Avg Plank Temp Diff [K]	Max Plank Temp [K]	Max Plank Temp Diff [K]	$\ \Delta T\ _N$
All	1019.965	-	1128.306	-	-
948	1019.997	+0.032	1128.386	+0.080	0.007
1024	1019.989	+0.023	1128.640	+0.333	0.023
1100	1019.976	+0.011	1128.300	-0.006	0.029
1200	1019.943	-0.023	1128.052	-0.255	0.043

The temperature differences between the AHTR plank model using group constants at all

four temperatures and the AHTR plank models' using group constants at each single temperature are not significant. Since all the single temperature group constant data in Table 5.14 have non-significant temperature differences, any of them could be used for the Moltres temperature models. The 948K single temperature group constant data has the lowest 2-norm difference ($\|\Delta T\|_N$). Thus, I chose to use a 948K in the AHTR plank OpenMC neutronics model to generate the group constant data for the Moltres temperature models.

Table 5.15: Advanced High-Temperature Reactor (AHTR) one-third assembly's average and maximum temperature and 2-norm of the temperature difference across the slab's centerline for varying group constant temperature data. The difference values are calculated from comparison against the Moltres model using group constant data with all four temperatures (All).

Group Constant Data Temps [K]	Avg One-Third Assembly Temp [K]	Diff [K]	Max One-Third Assembly Temp [K]	Diff [K]	$\ \Delta T\ _N$
All	1004.366	-	1186.359	-	-
948	1004.708	+0.341	1187.658	+1.299	0.044
1024	1004.676	+0.310	1187.148	+0.788	0.041
1100	1004.622	+0.256	1186.085	-0.274	0.046
1200	1004.539	+0.172	1185.359	-1.000	0.047

The temperature differences between the AHTR one-third assembly model using group constants at all four temperatures and the AHTR plank models using group constants at a single temperature are small ($< 1K$). The 1024K single temperature group constant data has the lowest 2-norm difference ($\|\Delta T\|_N$). Thus, I use 1024K in the AHTR one-third assembly OpenMC neutronics model to generate the group constant data for the Moltres temperature models.

5.5 ROLLO Hyperparameter Tuning

Recall from Chapter 2, I discussed how genetic algorithms require a good hyperparameter set that guides the optimization process by balancing exploitation and exploration to find an optimal solution quickly and accurately [66]. Finding a good hyperparameter set requires a trial-and-error process [66]. In a ROLLO input file, the user defines hyperparameters for the genetic algorithm. The subsequent subsections describe the hyperparameter search I conducted for single-objective and multi-objective optimization.

5.5.1 ROLLO Single-Objective Optimization Hyperparameters

I performed the single-objective hyperparameter search with a coarse-to-fine random sampling scheme, whose advantages I previously discussed in Section 2.4.2. I used an AHTR plank OpenMC model for the hyperparameter search (Figure 5.2). The hyperparameters are population size, number of generations, mutation probability, mating probability, selection operator, selection operator's number of individuals, selection operator's tournament size, mutation operator, and mating operator. I started with 25 coarse experiments and fine-tuned the hyperparameters with 15 more experiments. For each genetic algorithm experiment, total number of OpenMC models run remained constant at 600. The number of evaluations correlates with the population size and the number of generations. I randomly sampled population size and used Equation 5.14 to calculate the number of generations:

$$\begin{aligned} gens &= \frac{evals}{pop_size} \\ &= \frac{600}{pop_size} \end{aligned} \tag{5.14}$$

where

gens = no. of generations

evals = no. of evaluations

pop-size = population size

Table 5.16 shows the lower and upper bounds used for randomly sampling each hyperparameter at each phase of the hyperparameter search.

The initial 25 coarse experiments sought to narrow down the hyperparameters to find a smaller set of hyperparameter bounds that produce higher k_{eff} values. Figure 5.18 shows the hyperparameters plotted against each other with a third color dimension representing the average k_{eff} value ($\overline{k_{eff}}$) in each experiment's final generation. Lighter scatter points indicate higher final population k_{eff} values, suggesting better hyperparameter sets. I plot the hyperparameters against each other to visualize the interdependence between hyperparameters. From the coarse hyperparameter search

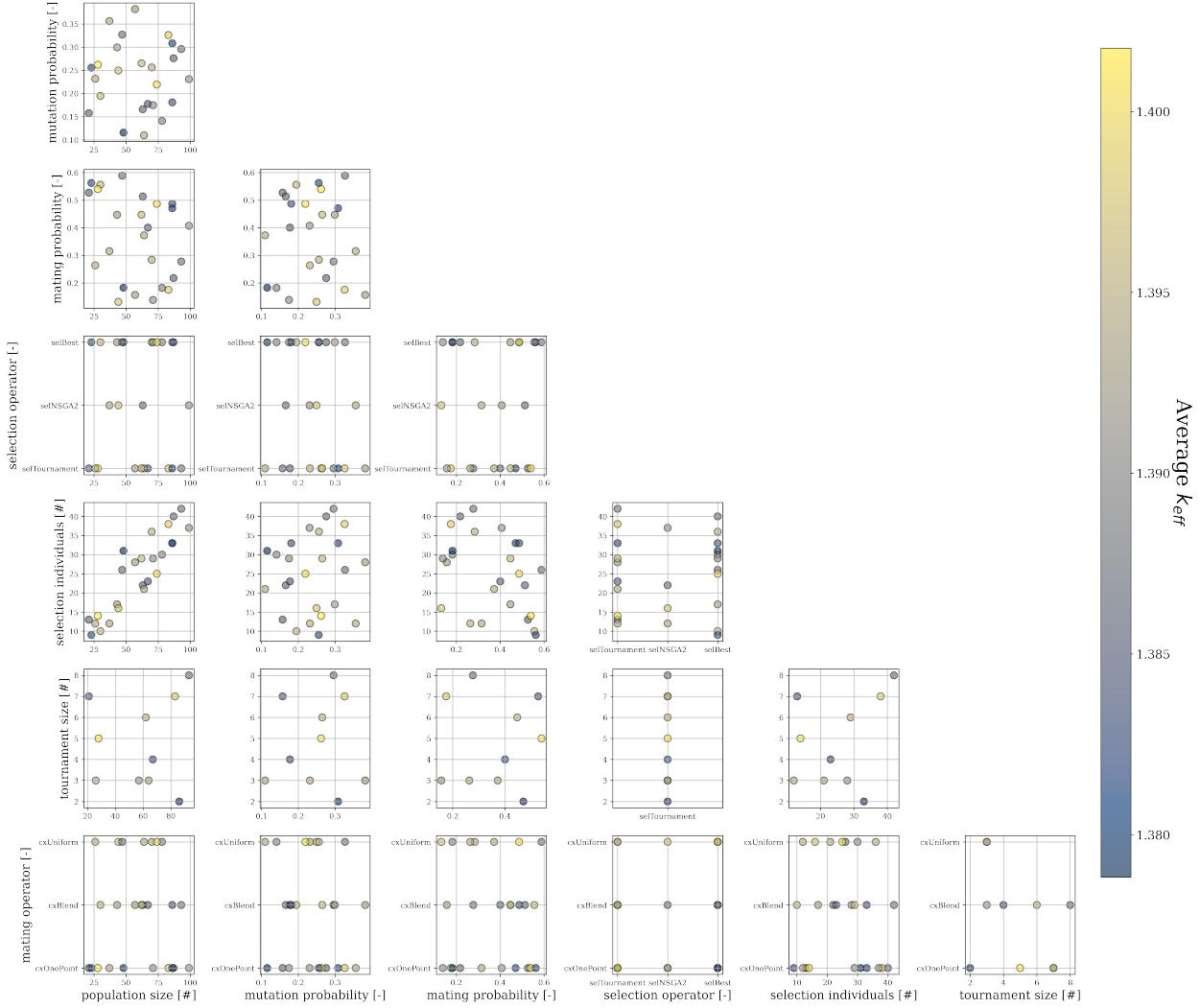


Figure 5.18: Coarse Reactor evOLutionary aLgorithm Optimizer (ROLLO) hyperparameters search's results. Hyperparameter values are plotted against each other with a third color dimension representing each experiment's final population's \bar{k}_{eff} .

Table 5.16: The hyperparameter search is conducted in three phases: *Coarse Search*, *Fine Search 1*, *Fine Search 2*. Each hyperparameter's lower and upper bounds for each search phase are listed.

Hyperparameter	Type	Coarse Search Bounds	Fine Search 1 Bounds	Fine Search 2 Bounds
Experiments [#]	-	0 to 24	24 to 34	35 to 39
Population size [#]	Continuous	$10 < x < 100$	$20 < x < 60$	60
Mutation probability [-]	Continuous	$0.1 < x < 0.4$	$0.2 < x < 0.4$	$0.2 < x < 0.3$
Mating probability [-]	Continuous	$0.1 < x < 0.6$	$0.1 < x < 0.3$	$0.45 < x < 0.6$
Selection operator [-]	Discrete	<code>SelTournament</code> , <code>SelBest</code> , <code>SelNSGA2</code>	<code>SelTournament</code> , <code>SelBest</code> , <code>SelNSGA2</code>	<code>SelTournament</code>
Selection individuals [#]	Continuous	$\frac{1}{3}pop < x < \frac{2}{3}pop$	$\frac{1}{3}pop < x < \frac{2}{3}pop$	15
Selection tournament size (only for SelTournament) [#]	Continuous	$2 < x < 8$	$2 < x < 8$	5
Mutation operator [-]	Discrete	<code>mutPolynomialBounded</code>	<code>mutPolynomialBounded</code>	<code>mutPolynomialBounded</code>
Mating operator [-]	Discrete	<code>cxOnePoint</code> , <code(cxuniform, code="" cxblend)<=""></code(cxuniform,>	<code>cxOnePoint</code> , <code>cxUniform, cxBlend</code>	<code>cxOnePoint, cxBlend</code>

(Figure 5.18), I noticed the following trends:

- Mutation probability has a higher $\overline{k_{eff}}$, between 0.2 and 0.4.
- Mating probability has a higher $\overline{k_{eff}}$, between 0.1 and 0.3.
- Population size has a higher $\overline{k_{eff}}$, between 20 and 60.
- No obvious interdependence between hyperparameters.

Next, I proceeded to the fine searches. From Figure 5.18, I narrowed down population size, mutation probability, and mating probability bounds, as shown in Table 5.16's *Fine Search 1 Bounds* column. I found no significant trends in the other hyperparameters, so I left them as is. I ran ten more experiments (25 to 34), sampling hyperparameters from the *Fine Search 1 Bounds*. From these results, I conducted a second fine search with five experiments (35 to 39) with further tuned hyperparameter bounds, as shown in Table 5.16's *Fine Search 2 Bounds* column. I determined these new hyperparameter bounds based on these reasons:

- Mutation probability has a higher $\overline{k_{eff}}$, between 0.2 and 0.3.
- I overlooked $\overline{k_{eff}}$ peaking at mating probability between 0.45 and 0.6 in the previous *Fine Search 1*; thus, I shifted the bounds.

- The highest $\overline{k_{eff}}$ occurred for **selTournament**.
- I narrowed down mating operator options to **cxBlend** and **cxOnePoint** since they had higher $\overline{k_{eff}}$ than **cxUniform**.
- I selected arbitrary numbers for population size, selection individuals, and tournament size since they did not correlate with $\overline{k_{eff}}$ values.

Figure 5.19 shows the relationship between hyperparameter values and a , b , c control variables, final generation k_{effmax} , and final generation $\overline{k_{eff}}$. The coarse experiments' scatter points are 50% transparent, while the fine experiments' scatter points are opaque. In Figure 5.19, on average, the

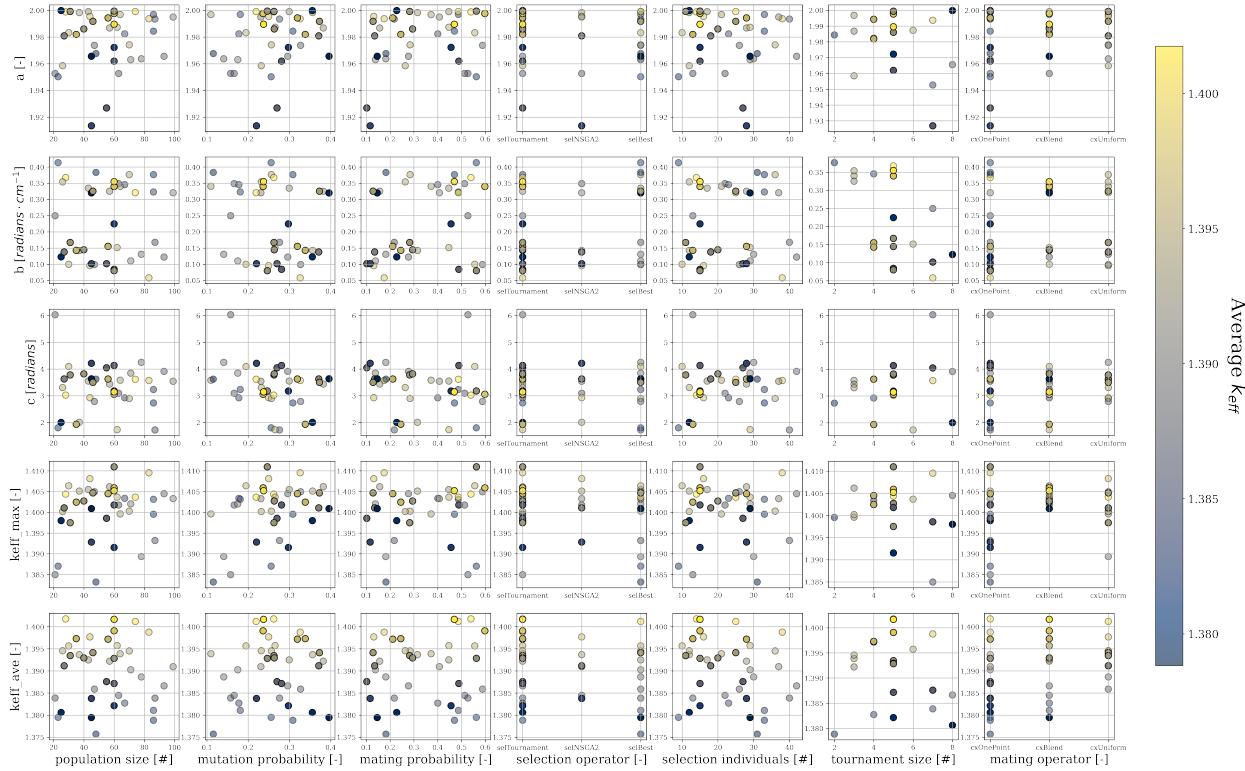


Figure 5.19: Reactor evOLutionary aLgorithm Optimizer (ROLLO) hyperparameter search results for all 40 experiments (coarse and fine). I plot the hyperparameters against: the a , b , c control variables, each experiment's final generation k_{effmax} , and the final generation $\overline{k_{eff}}$ with a third color dimension representing each experiment's final population's $\overline{k_{eff}}$ (color bar representing the $\overline{k_{eff}}$ values provided on the right side of the figure). Coarse experiments' (0 to 24) scatter points are 50% transparent, while the fine experiments' (24 to 39) scatter points are opaque.

fine experiments (opaque scatter points) have higher $\overline{k_{eff}}$, which indicates that the hyperparameter

search process met its objective of finding hyperparameter bounds that enable quicker and more accurate optimization.

I ran these hyperparameter search simulations on the BlueWaters supercomputer [87]. In each ROLLO simulation, each generation runs a population size number of individual OpenMC simulations. Each OpenMC simulation takes approximately 13 minutes to run on a single BlueWaters XE node. With approximately 600 OpenMC evaluations per ROLLO simulation, one ROLLO simulation takes about 130 BlueWaters node-hours. The hyperparameter search ran 40 ROLLO simulations, thus using approximately 5200 node-hours.

I define the best-performing hyperparameter set as the experiment that produces the highest $\overline{k_{eff}}$ in its final generation. Table 5.17 shows the hyperparameters for the five experiments with the highest final generation $\overline{k_{eff}}$. *Fine Search 2*'s experiment 39 produces the best performing hyperparameter set, with $\overline{k_{eff}} = 1.40165\beta$. Therefore, for single-objective ROLLO optimization simulations in this dissertation, I use the hyperparameter set from experiment 39.

Table 5.17: Control variables, k_{eff} results, and hyperparameter values for the five hyperparameter search experiments with the highest final generation $\overline{k_{eff}}$.

Control/Output Variables	Experiment No.				
	6	15	24	36	39
k_{eff} [-]	1.39876	1.40155	1.40118	1.39906	1.40165
k_{effmax} [-]	1.40954	1.40440	1.40365	1.40590	1.40519
a [-]	1.993	1.998	1.999	1.997	1.989
b [$radians \cdot cm^{-1}$]	0.057	0.367	0.320	0.339	0.354
c [$radians$]	3.571	3.022	3.615	3.053	3.143
Hyperparameter					
Population size [#]	83	28	74	60	60
Generations [#]	8	22	9	10	10
Mutation probability [-]	0.32	0.26	0.21	0.23	0.23
Mating probability [-]	0.17	0.53	0.48	0.59	0.46
Selection operator [-]	selTournament	selTournament	selBest	selTournament	selTournament
Selection individuals [#]	38	14	25	15	15
Selection tournament size [#]	7	5	-	5	5
Mutation operator [-]	mutPolynomialBounded	mutPolynomialBounded	mutPolynomialBounded	mutPolynomialBounded	mutPolynomialBounded
Mating operator [-]	cxOnePoint	cxOnePoint	cxUniform	cxBlend	cxBlend

5.5.2 ROLLO Multi-Objective Optimization Hyperparameters

Due to a lack of computational resources, I did not use an OpenMC example problem or a coarse-to-fine random sampling scheme for the multi-objective optimization hyperparameter search. I used the Binh and Korn function described in Section 4.4.2 as an example problem to conduct the hyperparameter search.

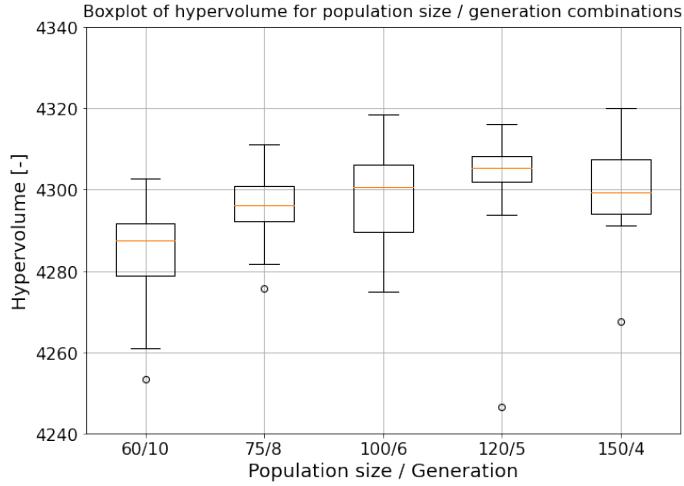
I conducted a simpler hyperparameter search for the multi-objective optimization case. I chose to use `SelNSGA2` for the selection operator, since it is an elitist operator that works well for multi-objective optimization. Next, I varied each of the following parameters individually and compared their hypervolume values: population size/generations, mutation probability, and mating probability. Section 4.4.2 described that the hypervolume indicator quantifies how well the multi-objective optimization’s Pareto front covers the multi-objective domain (bigger = better).

Figure 5.20 shows boxplots comparing hypervolumes for various hyperparameters. I ran each simulation 25 times to produce each box plot. Figure 5.20a shows the boxplot comparing hypervolume for different population size and generation combinations. Similar to Section 5.5.1, I held the total number of OpenMC evaluations constant at 600. Equation 5.14 shows the relationship between population size and generations. The combination of 120 population size and 5 generations performs the best with the highest hypervolume median (orange line).

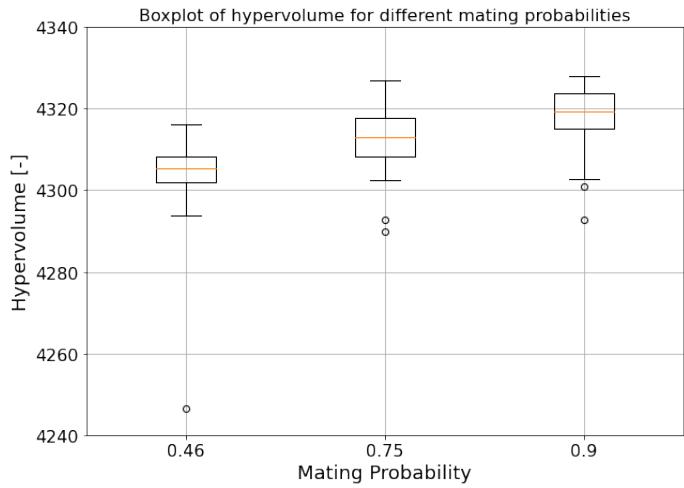
Figure 5.20b shows a boxplot comparing hypervolume for different mating probabilities. The 0.9 mutation probability performs the best with the highest hypervolume median (orange line), compared to the 0.46 from the best single-objective hyperparameter set (Section 5.5.1). Figure 5.20c shows a boxplot comparing hypervolume for different mutation probabilities. The 0.23 mating probability from the best single-objective hyperparameter set (Section 5.5.1) performs as well as the 0.1 mating probability.

Based on the box plot comparisons, I selected the combination of hyperparameters that maximized the hypervolume. Table 5.18 summarizes the hyperparameters I used for ROLLO multi-objective optimization.

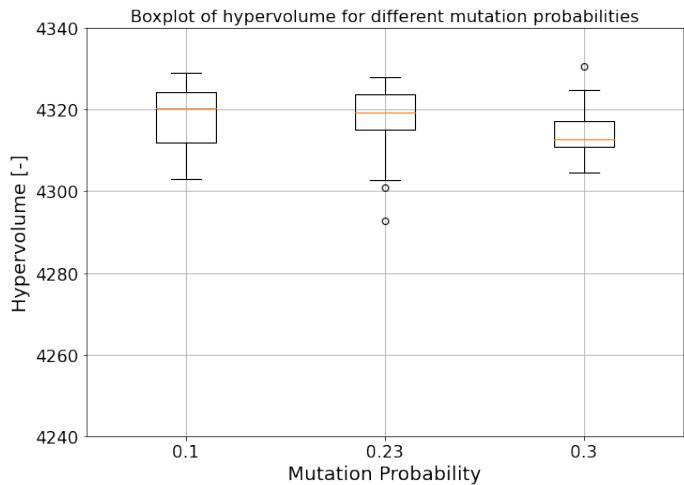
Reactor designers with ample computational power may use a reactor modeling problem to conduct their multi-objective optimization hyperparameter search. Using hyperparameters optimized for their reactor model could reduce the number of generations required to find a Pareto front



(a) Population size / Generation variation.



(b) Mating probability variation



(c) Mutation probability variation

Figure 5.20: Hypervolume comparisons for varying Reactor evOLutionary aLgorithm Optimizer (ROLLO) hyperparameters for the multi-objective optimization hyperparameter search.

Table 5.18: Reactor evOLutionary aLgorithm Optimizer (ROLLO) hyperparameters used for multi-objective optimization.

Hyperparameter	Value
Population size	120
Generations	5
Mutation probability	0.9
Mating probability	0.23
Selection operator	<code>SelNSGA2</code>
Mutation operator	<code>mutPolynomial_Bounded</code>
Mating operator	<code>cxBlend</code>

with a large hypervolume and also enable finding a more optimal Pareto front. However, using the multi-objective hyperparameters outlined in Table 5.18 is sufficient for any reactor designers that do not want to conduct a problem-specific hyperparameter search.

5.6 Summary

This chapter describes the modeling and optimization methodology of the Advanced High-Temperature Reactor (AHTR) plank and one-third assembly optimization for non-conventional geometries and parameters conducted using the Reactor evOLutionary aLgorithm Optimizer (ROLLO) software. I defined and described the AHTR input parameters varied for the optimization problems. The input parameters include Tristructural Isotropic (TRISO) packing distribution, total fuel packing fraction, and coolant channel shape. I defined and motivated the key optimization objectives: minimizing fuel amount, maximizing heat transfer, and minimizing power peaking (Table 5.1). I described the AHTR plank and one-third assembly geometries and detailed their modeling workflow: the AHTR input parameter variations, the OpenMC neutronics model, the Moltres temperature model, and the output constraint and objective value calculations. I conducted key neutronics parameter verifications, which confirmed that the spatial and energy homogenizations used for the AHTR plank and one-third assembly Moltres temperature models preserved accuracy. Finally, I described the ROLLO hyperparameter tuning process and reported the hyperparameters used for the single and multi-objective optimization problems.

Chapters 6 and 7 report the AHTR plank and AHTR one-third assembly optimization results that use this chapter's described methods.

Chapter 6

AHTR Plank Optimization Results

In this chapter, I report the Advanced High-Temperature Reactor (AHTR) plank's Reactor evOLutionary aLgorithm Optimizer (ROLLO) optimization results. I vary the following AHTR plank input parameters:

- Tristructural Isotropic (TRISO) packing fraction distribution ($\rho_{TRISO}(\vec{r})$)
- Total fuel packing fraction (PF_{total})
- Coolant channel shape (r_{top} and r_{bot})

Section 5.3.1 detailed how I vary these AHTR one-third assembly's input parameters. I optimize the AHTR plank for the following objectives:

- Minimize total fuel packing fraction (PF_{total})
- Minimize maximum plank temperature (T_{max})
- Minimize fuel-normalized power peaking factor (PPF_{fuel})

Table 6.1 outlines these objectives and their motivation. Chapter 5 detailed the methodology for AHTR plank modeling and ROLLO optimization.

Table 6.1: Reactor evOLutionary aLgorithm Optimizer (ROLLO) Advanced High-Temperature Reactor (AHTR) optimization problem objectives with their quantification descriptions and motivation.

Objective	Quantification	Motivation
Minimize fuel amount	Minimize total fuel packing fraction	Cost savings, Non-proliferation
Maximize heat transfer	Minimize maximum temperature	Minimize thermal stress in the fuel
Minimize power peaking	Minimize power peaking factor normalized by fuel distribution	Efficient fuel utilization, longer core life, safety

The subsequent sections outline the AHTR plank optimization simulations (Section 6.1), describe the single-objective (Section 6.2), double-objective (Section 6.3), and triple-objective (Section 6.4) ROLLO optimization simulation results, and report each simulation’s computational cost (Section 6.5). Appendix A lists all the data and analysis related to this chapter to enable the reproduction of all the simulations.

6.1 ROLLO AHTR Plank Optimization Simulations Overview

In this chapter, I first conduct single-objective, single input parameter ROLLO optimizations to understand the individual impacts of each objective on each input parameter. I then run multi-objective optimizations for combinations of the input parameters and objectives, whose setup is informed by the single-objective optimization results. Table 6.2 summarizes the details of each ROLLO optimization simulation conducted in this chapter.

k_{eff} strongly correlates with total fuel packing fraction (PF_{total}). The AHTR plank with the same PF_{total} as the Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark’s assembly model had a k_{eff} of 1.35. For optimization simulations that vary PF_{total} , I constrained $k_{eff} \geq 1.35$ to find optimal input parameters that achieve similar performance to the original benchmark TRISO distribution. For the optimization simulations that do not vary PF_{total} , I use $PF_{total} = 0.0979$, which corresponds to the FHR benchmark’s PF_{total} , and constrained $k_{eff} \geq 1.00$. Thus, the k_{eff} constraint values used are 1.35 and 1.00 for optimization simulations that do and do not vary total packing fraction, respectively.

Simulations are run on the BlueWaters supercomputer [87] and Theta supercomputer at the Argonne Leadership Computing Facility under the Director’s Discretionary Allocation Program [107]. Section 6.5 details the computational cost of each optimization simulation. Readers must consider the computational cost if they desire to reproduce this analysis.

Table 6.2: ROLLO simulations for optimizing AHTR plank. There are six single objective optimization simulations (p-1a to p-1f) that are used to understand the individual impacts of each objective on each input parameter. Their results are then used to inform the multi-objective optimization simulations (p-2a, p-2b, p-2c, p-3a, p-3b) setup. Relevant variables include: PF_{total} : total fuel packing fraction; T_{max} : maximum plank temperature; PPF_{fuel} : fuel-normalized power peaking factor; $\rho_{TRISO}(\vec{r})$: TRISO particle distribution.

Objs [#]	Sim	Objectives	Constraints	Varying Parameters	Simulation Software
1	p-1a	• $\min(PF_{total})$	• $k_{eff} \geq 1.35$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC
	p-1b	• $\min(T_{max})$	• $k_{eff} \geq 1.00$	• $\rho_{TRISO}(\vec{r})$	OpenMC, Moltres
	p-1c	• $\min(PPF_{fuel})$	• $k_{eff} \geq 1.00$	• $\rho_{TRISO}(\vec{r})$	OpenMC
	p-1d	• $\min(PF_{total})$	• $k_{eff} \geq 1.35$	• Coolant channel shape • PF_{total}	OpenMC
	p-1e	• $\min(T_{max})$	• $k_{eff} \geq 1.35$	• Coolant channel shape	OpenMC, Moltres
2	p-1f	• $\min(PPF_{fuel})$	• $k_{eff} \geq 1.35$	• Coolant channel shape	OpenMC
	p-2a	• $\min(PF_{total})$ • $\min(T_{max})$	• $k_{eff} \geq 1.35$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC, Moltres
	p-2b	• $\min(PF_{total})$ • $\min(PPF_{fuel})$	• $k_{eff} \geq 1.35$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC
	p-2c	• $\min(T_{max})$ • $\min(PPF_{fuel})$	• $k_{eff} \geq 1.00$	• $\rho_{TRISO}(\vec{r})$	OpenMC, Moltres
3	p-3a	• $\min(PF_{total})$ • $\min(PPF_{fuel})$ • $\min(T_{max})$	• $k_{eff} \geq 1.35$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC, Moltres
	p-3b	• $\min(PF_{total})$ • $\min(PPF_{fuel})$ • $\min(T_{max})$	• $k_{eff} \geq 1.35$	• $\rho_{TRISO}(\vec{r})$ • PF_{total} • Coolant channel shape	OpenMC, Moltres

6.2 AHTR Plank: Single-Objective Optimization Results

This section reports the AHTR plank’s ROLLO single-objective optimization results. Table 6.2 summarizes the parameters for the single-objective simulations that are presented in this section: p-1a, p-1b, p-1c, p-1d, p-1e, and p-1f. In the following subsections, I describe the single-objective simulation results grouped by the minimized objective (Sections 6.2.1, 6.2.2, and 6.2.3), and provide discussion about the single-objective simulations results (Section 6.2.4).

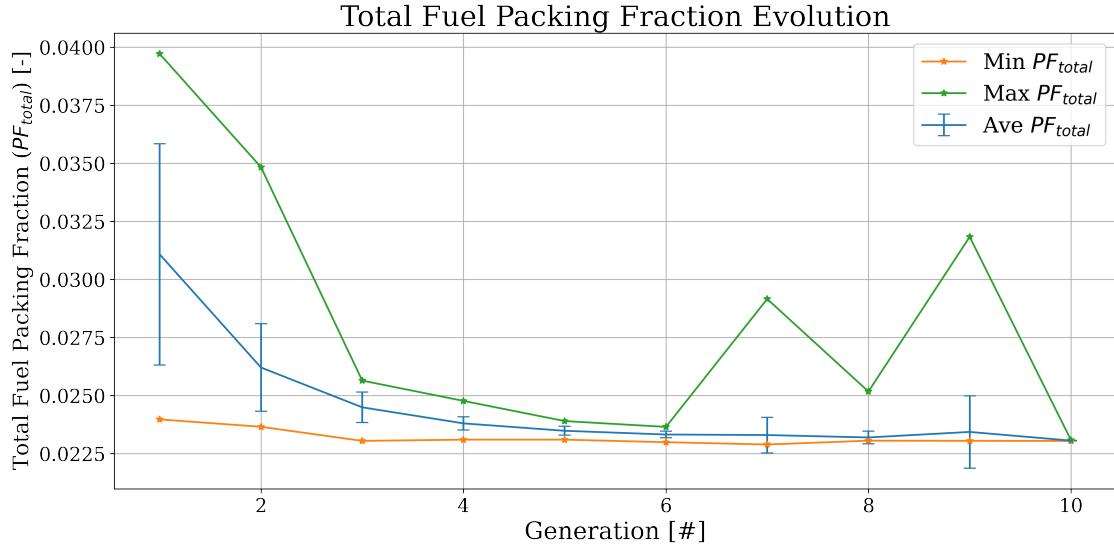
Section 4.5.1 described that for single-objective optimization, the reactor designer should plot the objective’s minimum and average values at each generation and when the values are no longer changing, the simulation is converged. If a single-objective optimization problem’s objective converges earlier than the five generations I intended to run (determined in Section 5.5.2), I stop the simulation at that generation to save computational resources as the solution will not change. However, if the problem’s objective does not converge by generation 5, I run the problem for a few more generations until satisfactory convergence is reached.

6.2.1 Objective: Minimize Total Packing Fraction (PF_{total})

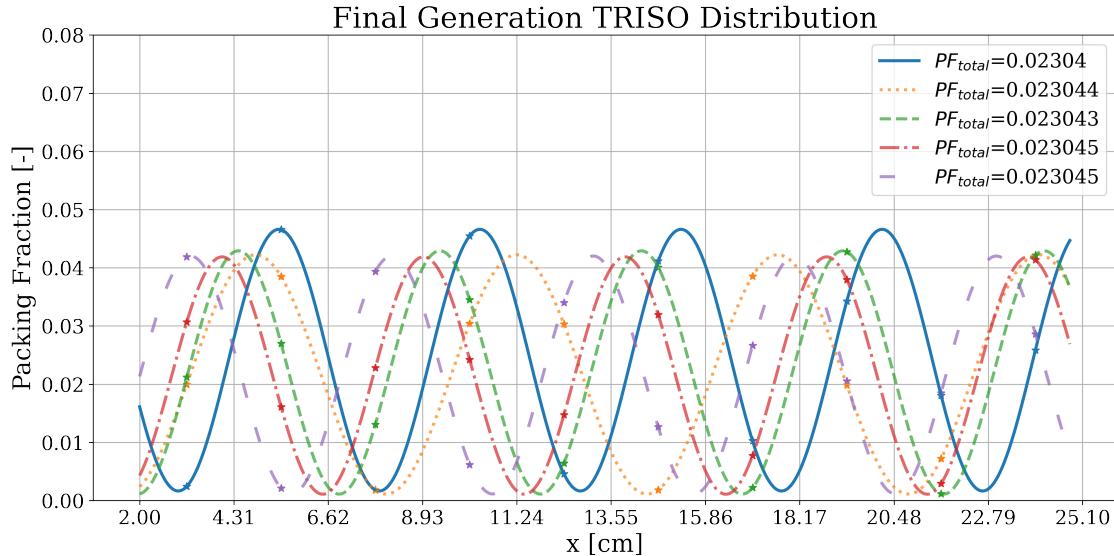
This section describes the single-objective p-1a and p-1d optimization simulation results. Both simulations minimize the total fuel packing fraction (PF_{total}) objective. The minimize PF_{total} objective is important because a reactor that uses less fuel with similar performance enables cost savings. Simulation p-1a varies the total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), while simulation p-1d varies the total fuel packing fraction (PF_{total}) and coolant channel shape.

Simulation p-1a: Variation of PF_{total} and $\rho_{TRISO}(\vec{r})$

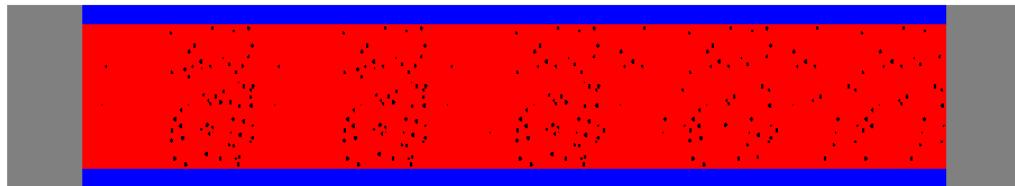
Table 6.3 summarizes the optimization problem parameters for simulation p-1a. Figure 6.1a shows simulation p-1a’s PF_{total} evolution, Figure 6.1b shows the five TRISO packing fraction distributions ($\rho_{TRISO}(\vec{r})$) in the final generation with the most-minimized PF_{total} , and Figure 6.1c illustrates the AHTR plank model with the most-minimized PF_{total} . Figure 6.1a shows that the minimum and average PF_{total} converged quickly. The final generation’s average PF_{total} values for all reactor models converged to approximately 0.023. In Figure 6.1b, the TRISO packing fraction distributions



(a) Minimum, average, and maximum PF_{total} evolution of the population in each generation.



(b) TRISO packing fraction distribution for the five reactor models with the smallest PF_{total} in the final generation.



(c) AHTR plank model with the most-minimized PF_{total} (corresponds to the blue solid distribution in Figure 6.1b).

Figure 6.1: Simulation p-1a – ROLLO single-objective optimization to minimize the total fuel packing fraction (PF_{total}). Input parameters varied: total fuel packing fraction (PF_{total}), TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

Table 6.3: Simulation p-1a optimization problem parameters.

Single Objective: Simulation p-1a	
Objectives	Minimize PF_{total}
Input parameter variations	$0.02 \leq PF_{total} \leq 0.04$ $\rho_{TRISO}(\vec{r})$: $0 < a < 2$ $\rho_{TRISO}(\vec{r})$: $0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r})$: $0 < c < 2\pi$
Constraints	$k_{eff} \geq 1.35$
Genetic algorithm parameters	Population size: 64 Generations: 10

are not the same but follow a similar oscillating TRISO distribution pattern. Because it is more productive to compare all of the single objective results to one another, Section 6.2.4 discusses the driving factors for the minimize PF_{total} objective and explains simulation p-1a's most-minimized PF_{total} oscillating TRISO distribution.

Simulation p-1d: Variation of PF_{total} and Coolant channel shape

Table 6.4 summarizes the optimization problem parameters for simulation p-1d.

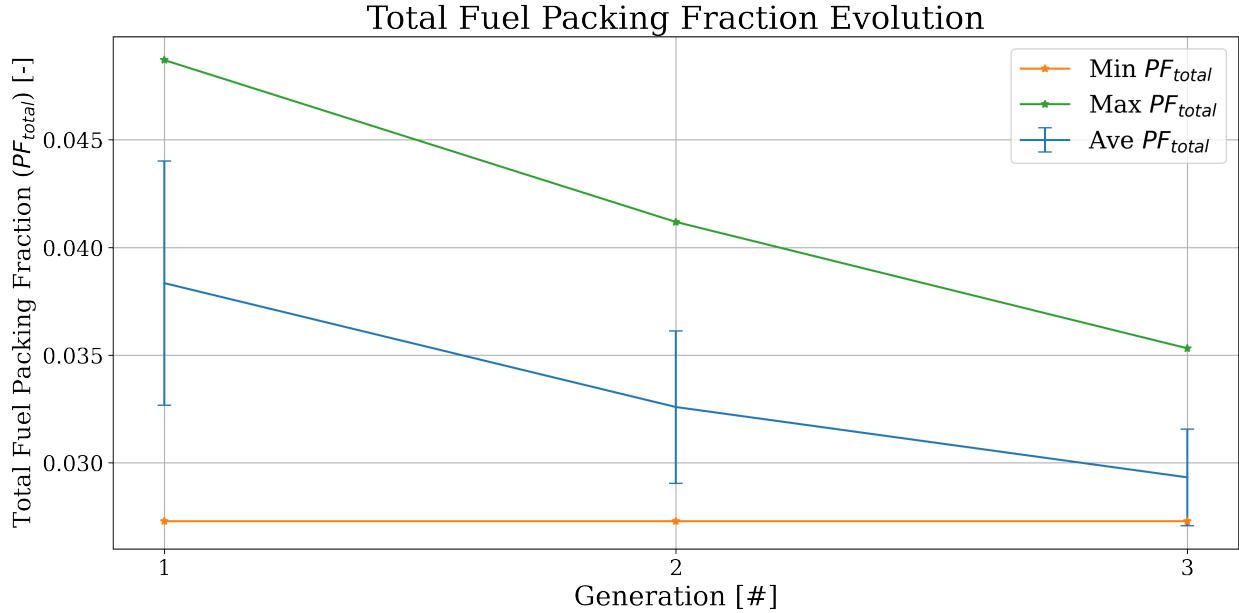
Table 6.4: Simulation p-1d optimization problem parameters.

Single Objective: Simulation p-1d	
Objectives	Minimize PF_{total}
Input parameter variations	$0.02 \leq PF_{total} \leq 0.04$ Coolant channel shape: $0.05 < r_{top} < 0.35$ Coolant channel shape: $0.05 < r_{bot} < 0.35$
Constraints	$k_{eff} \geq 1.35$
Genetic algorithm parameters	Population size: 64 Generations: 3

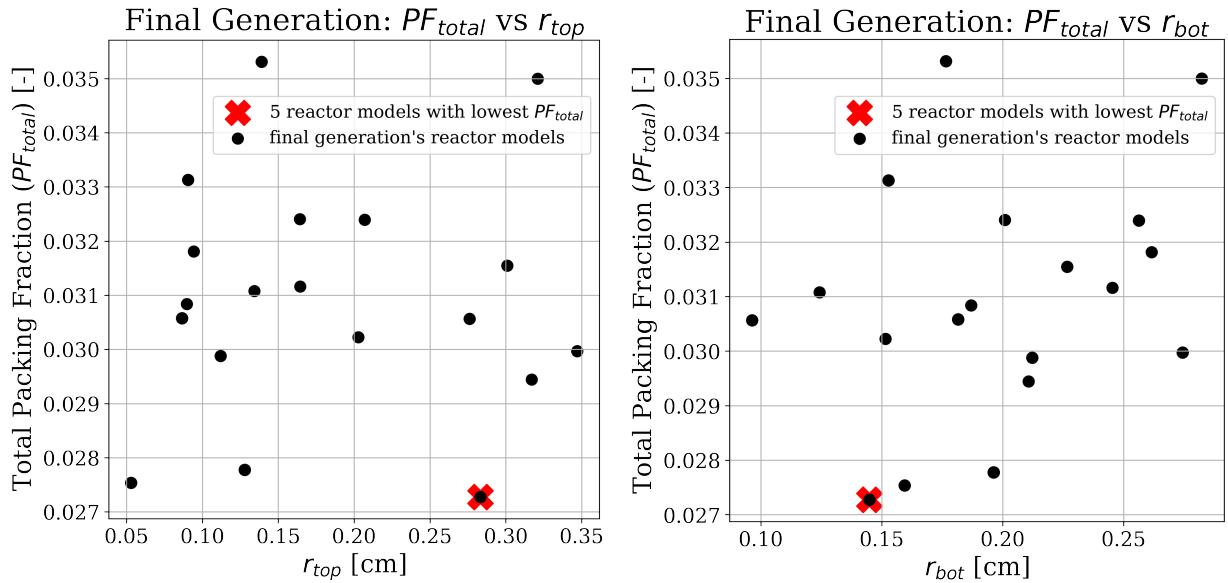
Figure 6.2a shows the PF_{total} evolution converges over three generations. Figure 6.2b shows plots of r_{top} and r_{bot} against PF_{total} . The random scattering of reactor model points in Figure 6.2b demonstrates that there is no correlation between PF_{total} , and r_{top} and r_{bot} .

6.2.2 Objective: Minimize Maximum Plank Temperature (T_{max})

This section describes the single-objective p-1b and p-1e optimization simulation results. Both simulations minimize the maximum plank temperature (T_{max}) objective. The minimize T_{max} objective is important because a reactor that has a lower peak tempeature minimizes thermal stresses



(a) Minimum, average, and maximum total PF_{total} evolution of the population in each generation.



(b) Plots of r_{top} and r_{bot} against PF_{total} . Red crosses indicate the five reactor models with the lowest PF_{total} . There is only one red cross shown in each plot because the five reactor models with the lowest PF_{total} are the same and overlap.

Figure 6.2: Simulation p-1d – ROLLO single-objective optimization to minimize the total fuel packing fraction (PF_{total}). Input parameters varied: total fuel packing fraction (PF_{total}), and coolant channel shape (r_{top}, r_{bot}).

in the fuel. Simulation p-1b varies the TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), while simulation p-1e varies the coolant channel shape.

Simulation p-1b: Variation of $\rho_{TRISO}(\vec{r})$

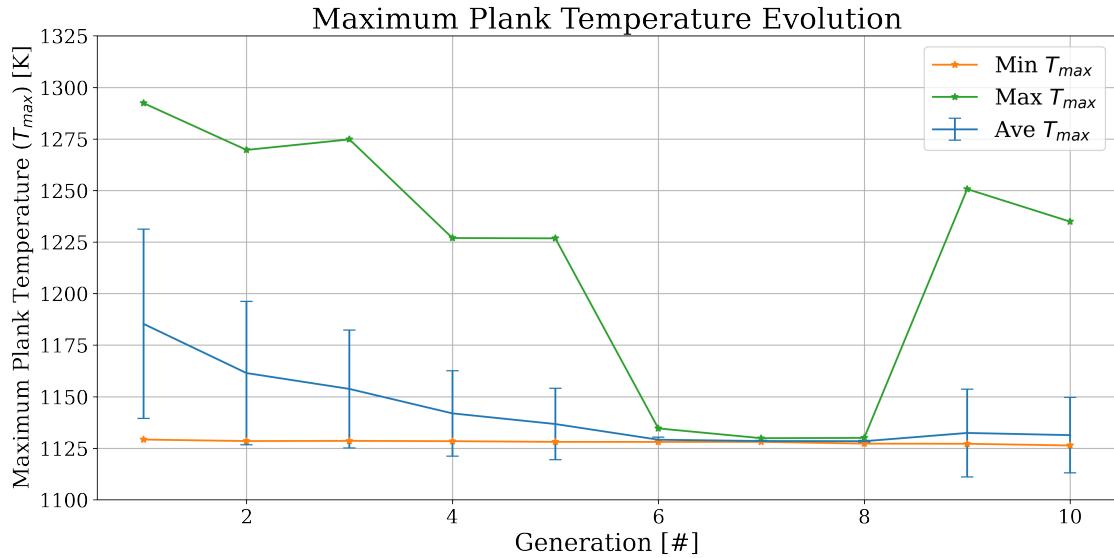
Table 6.5 summarizes the optimization problem parameters for simulation p-1b.

Table 6.5: Simulation p-1b optimization problem parameters.

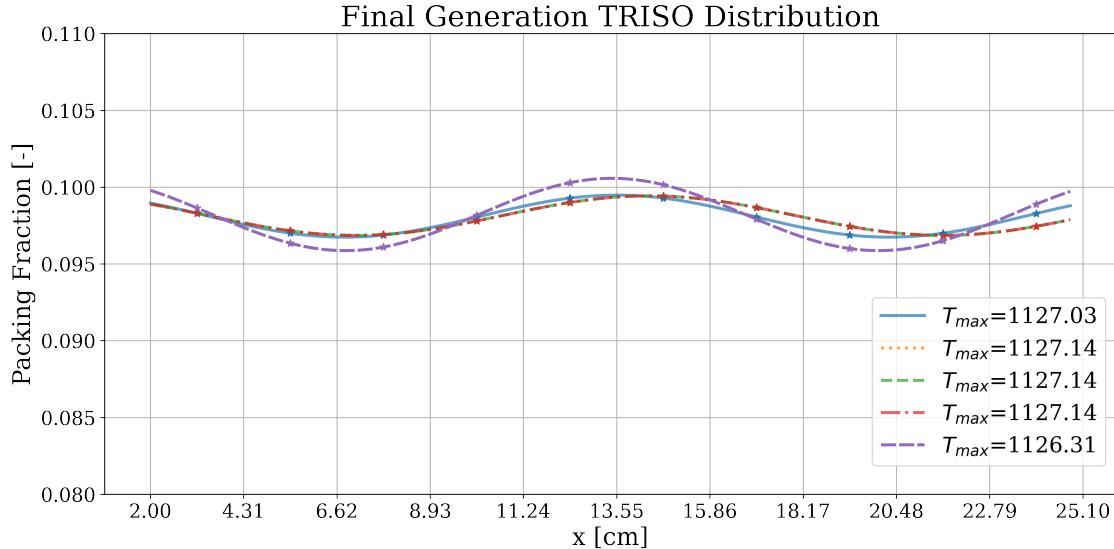
Single Objective: Simulation p-1b	
Objectives	Minimize T_{max}
Input parameter variations	$\rho_{TRISO}(\vec{r}): 0 < a < 2$ $\rho_{TRISO}(\vec{r}): 0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 < c < 2\pi$
Constraints	$k_{eff} \geq 1.0$ $PF_{total} = 0.0979$
Genetic algorithm parameters	Population size: 64 Generations: 10

Unlike simulation p-1a, I did not vary PF_{total} and held it constant at 0.0979. Section 6.1 described that for the optimization simulations which do not vary PF_{total} , I use $PF_{total} = 0.0979$, which corresponds to the FHR benchmark's PF_{total} . This PF_{total} is higher than the PF_{total} variation utilized in simulation p-1a of 0.02 to 0.04. I chose to use a lower range for PF_{total} variations, I observed that the TRISO spatial variation enabled comparable k_{eff} values with lower PF_{total} values.

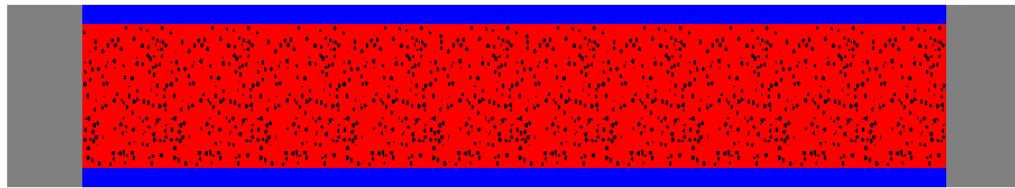
Figure 6.3a shows the plank's T_{max} evolution. Figure 6.3b shows the five TRISO packing fraction distributions ($\rho_{TRISO}(\vec{r})$) with the most-minimized T_{max} from the final generation. Figure 6.3c illustrates the AHTR plank model with the most-minimized T_{max} . Figure 6.3a shows that the minimum and average plank's T_{max} converged to approximately 1125 K. In Figure 6.3b, a mostly flat TRISO distribution minimizes T_{max} in the plank; the TRISO distribution has two small dips at the one-third and two-third points in the plank (6.62cm and 20.48cm). Because it is more productive to compare all of the single objective results to one another, Section 6.2.4 discusses and explains simulation p-1b's most-minimized T_{max} mostly flat TRISO distribution with a packing fraction standard deviation of 0.0015.



(a) Minimum, average, and maximum T_{max} evolution of the population in each generation.



(b) TRISO distribution for the five reactor models with the lowest AHTR plank T_{max} at the final generation.



(c) AHTR plank model with the most-minimized T_{max} (corresponds to the purple densely dashed distribution in Figure 6.3b).

Figure 6.3: Simulation p-1b – ROLLO single-objective optimization to minimize the maximum plank temperature (T_{max}). Input parameter varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$). $PF_{total} = 0.0979$.

Simulation p-1e: Variation of Coolant Channel Shape

Table 6.6 summarizes the optimization problem parameters for simulation p-1e.

Table 6.6: Simulation p-1e optimization problem parameters.

Single Objective: Simulation p-1e	
Objectives	Minimize T_{max}
Input parameter variations	Coolant channel shape: $0.05 < r_{top} < 0.35$ Coolant channel shape: $0.05 < r_{bot} < 0.35$
Constraints	$k_{eff} \geq 1.35$ $PF_{total} = 0.0979$
Genetic algorithm parameters	Population size: 64 Generations: 4

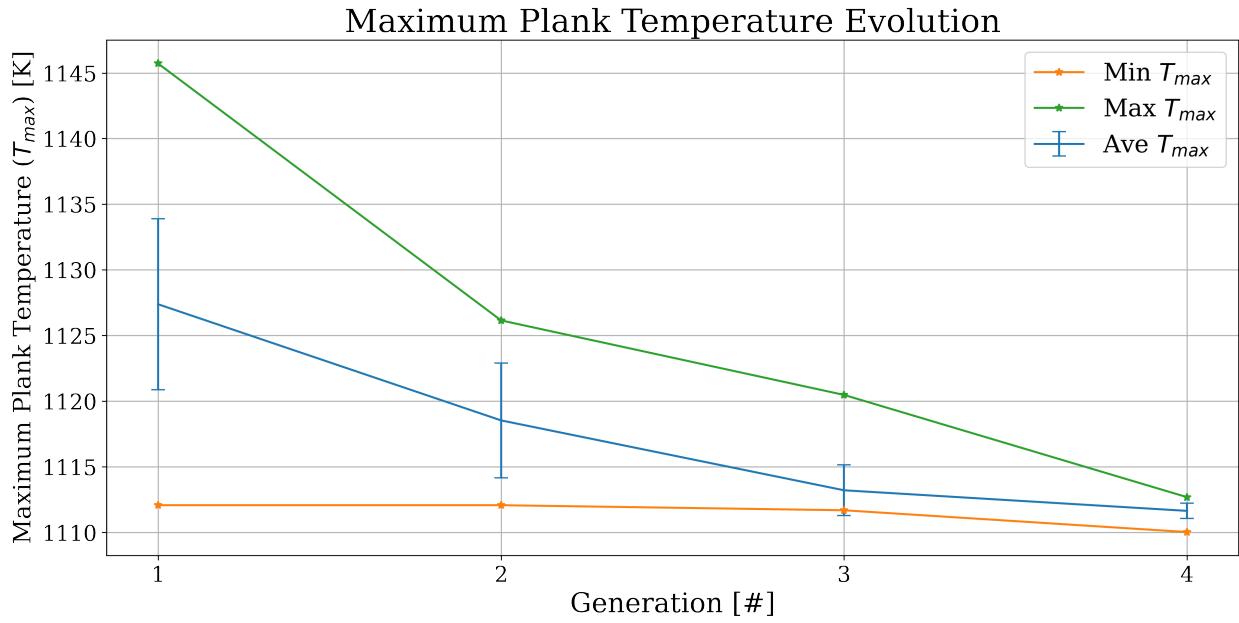
Figure 6.4a shows the plank's T_{max} evolution. Figure 6.4b shows the plots of r_{top} and r_{bot} against T_{max} . Figure 6.4c illustrates the AHTR plank model with the most-minimized T_{max} . Figure 6.4b demonstrates a negative linear correlation between the plank's T_{max} , and r_{top} and r_{bot} . Section 6.2.4 discusses and explains the relationship between T_{max} and coolant channel shape.

6.2.3 Objective: Minimize Fuel-Normalized Power Peaking Factor (PPF_{fuel})

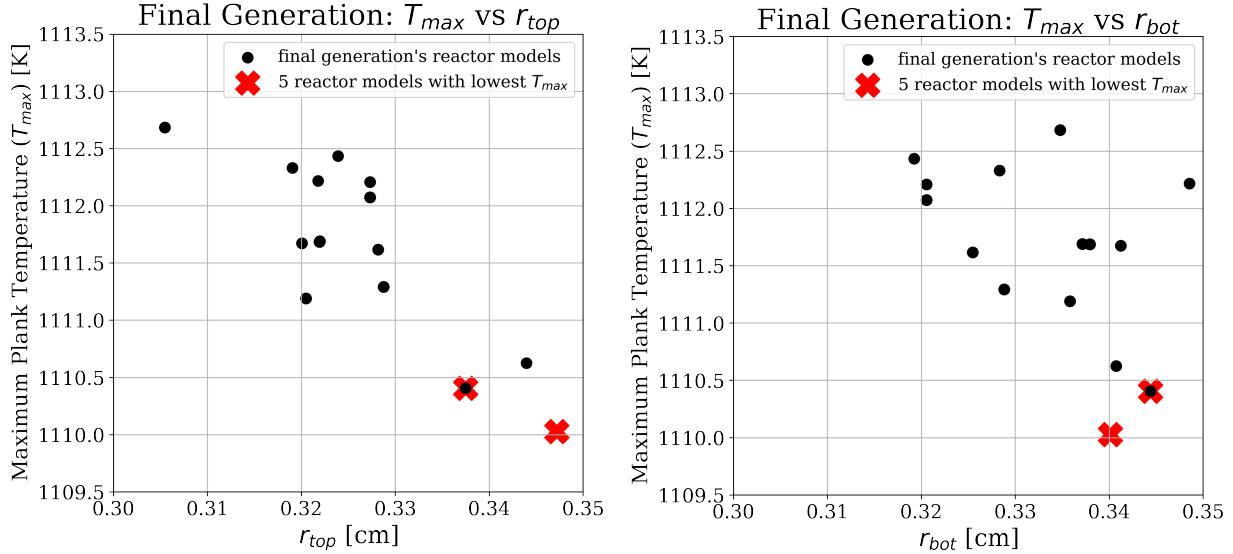
This section describes the single-objective p-1c and p-1f optimization simulation results. Both simulations minimize fuel-normalized power peaking factor (PPF_{fuel}). The minimize PPF_{fuel} objective is important because a reactor that has a lower fuel-normalized power peaking factor will have more even and efficient fuel utilization. Simulation p-1c varies the TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), while simulation p-1f varies the coolant channel shape (r_{top} and r_{bot}).

Simulation p-1c: Variation of $\rho_{TRISO}(\vec{r})$

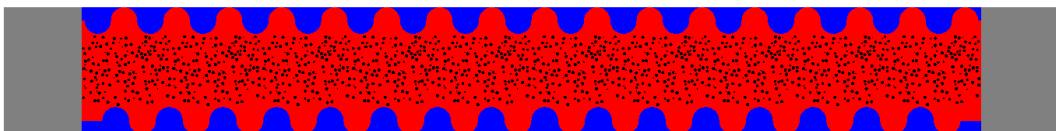
Table 6.7 summarizes simulation p-1c's optimization problem parameters.



(a) Minimum, average, and maximum T_{max} evolution of the population in each generation.



(b) Plots of r_{top} and r_{bot} against T_{max} . Red crosses indicate the five reactor models with the lowest T_{max} . Some of the five reactor models with the lowest T_{max} are the same, thus their crosses overlap.



(c) AHTR plank model with the most-minimized T_{max} . $r_{top} = 0.337$ and $r_{bot} = 0.344$.

Figure 6.4: Simulation p-1e – ROLLO single-objective optimization to minimize the maximum plank temperature (T_{max}). Input parameters varied: r_{top}, r_{bot} . $PF_{total} = 0.0979$ and constant TRISO distribution.

Table 6.7: Simulation p-1c optimization problem parameters.

Single Objective: Simulation p-1c	
Objectives	Minimize PPF_{fuel}
Input parameter variations	$\rho_{TRISO}(\vec{r})$: $0 < a < 2$ $\rho_{TRISO}(\vec{r})$: $0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r})$: $0 < c < 2\pi$
Constraints	$k_{eff} \geq 1.0$ $PF_{total} = 0.0979$
Genetic algorithm parameters	Population size: 60 Generations: 10

Figure 6.5a shows the plank's PPF_{fuel} evolution converges over ten generations. Figure 6.5b shows the five overlapped TRISO packing fraction distributions ($\rho_{TRISO}(\vec{r})$) in the final generation with the most-minimized PPF_{fuel} . Figure 6.5c illustrates the AHTR plank model with the most-minimized PPF_{fuel} . In Figure 6.5b, the plank model with the most-minimized PPF_{fuel} has a TRISO distribution that peaks near the edges of the fuel region and has a minimum point at the plank's center. Section 6.2.4 discusses the driving factors for the minimize PPF_{fuel} objective and explains simulation p-1c's most-minimized PPF_{fuel} TRISO distribution.

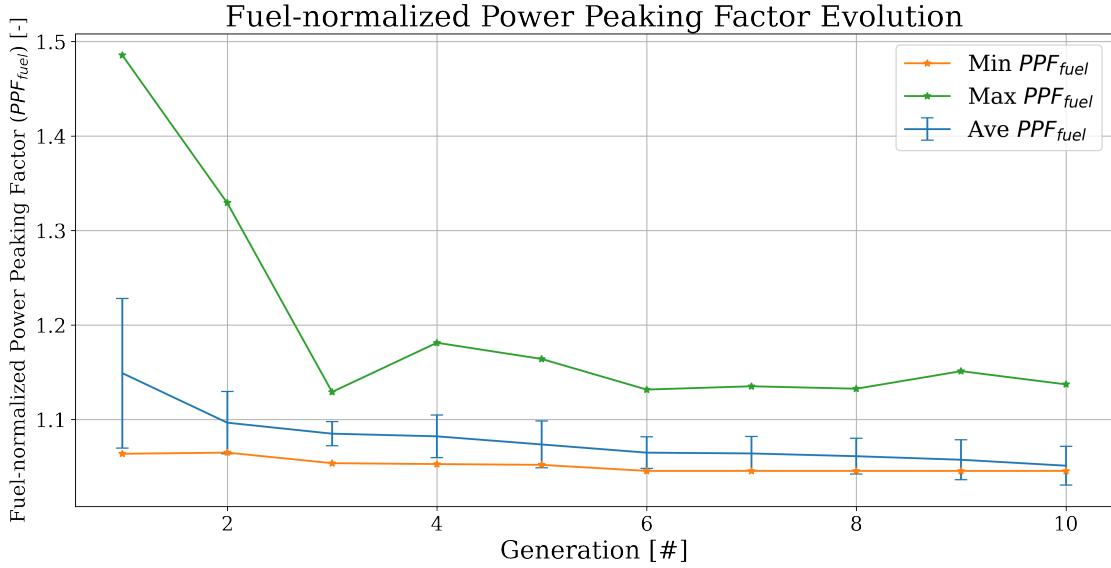
Simulation p-1f: Variation of Coolant Channel Shape

Table 6.8 summarizes simulation p-1f's optimization problem parameters.

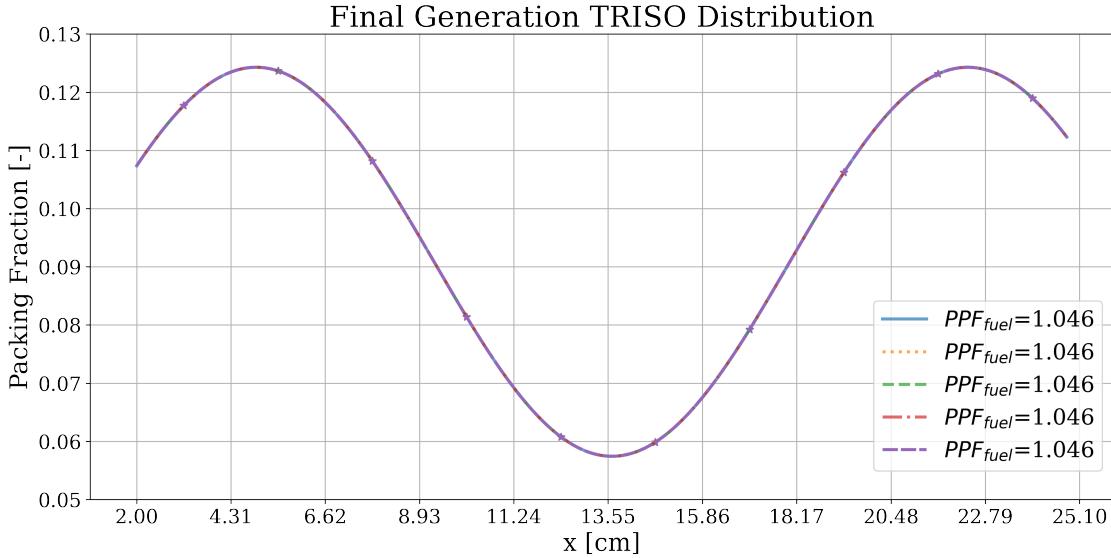
Table 6.8: Simulation p-1f optimization problem parameters.

Single Objective: Simulation p-1f	
Objectives	Minimize PPF_{fuel}
Input parameter variations	Coolant channel shape: $0.05 < r_{top} < 0.35$ Coolant channel shape: $0.05 < r_{bot} < 0.35$
Constraints	$k_{eff} \geq 1.35$ $PF_{total} = 0.0979$
Genetic algorithm parameters	Population size: 64 Generations: 5

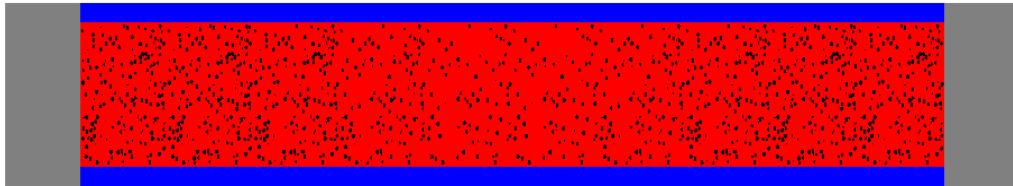
Figure 6.6a shows the plank's PPF_{fuel} evolution, and Figure 6.6b shows the plots r_{top} and r_{bot} against PPF_{fuel} . The random scattering of reactor model points in Figure 6.6b demonstrates that there is no correlation between PPF_{fuel} , and r_{top} and r_{bot} .



(a) Minimum, average, and maximum PPF_{fuel} evolution of the population in each generation.

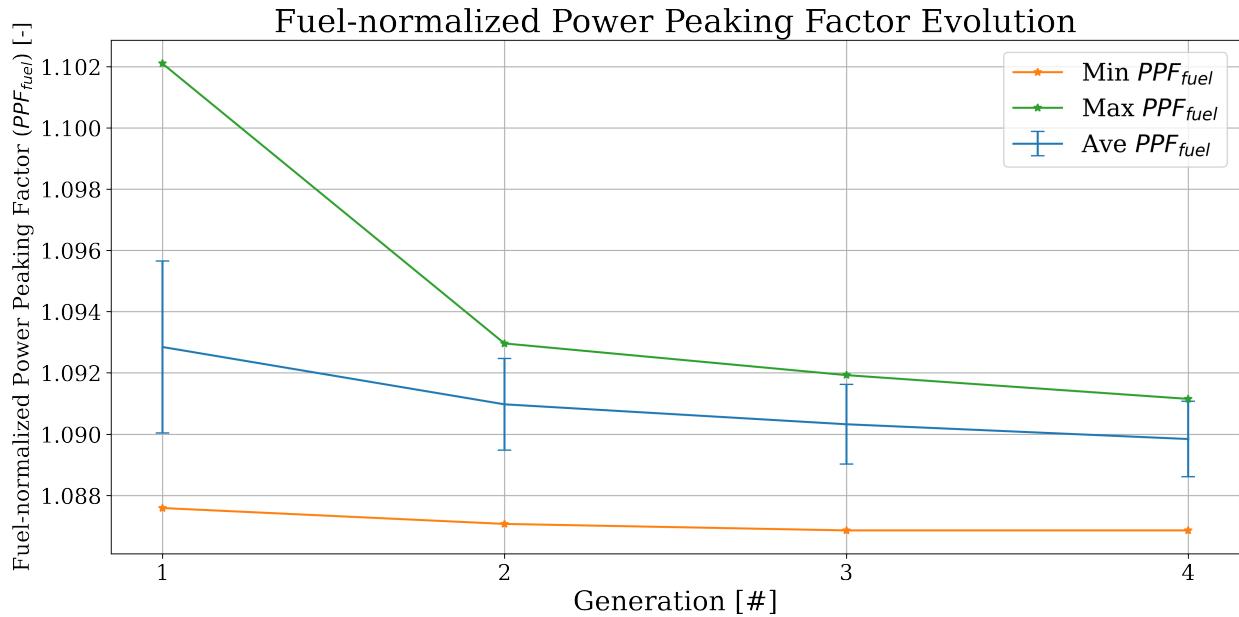


(b) TRISO distribution for the five reactor models with the lowest PPF_{fuel} in the AHTR plank at the final generation. The five reactor models with the lowest PPF_{fuel} are the same, thus their lines overlap.

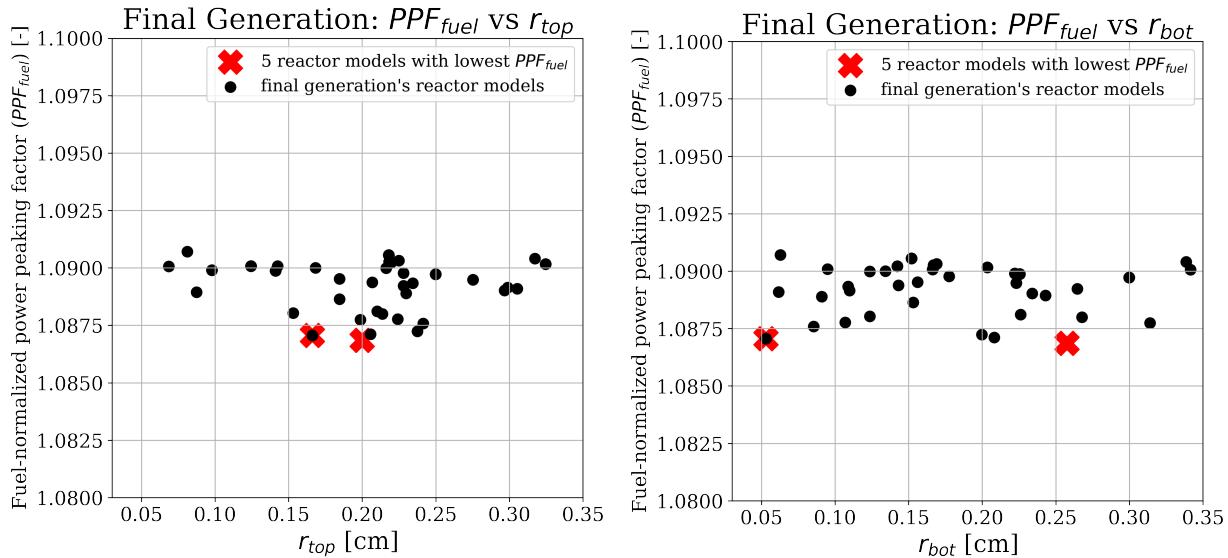


(c) AHTR plank model with the most-minimized PPF_{fuel} (corresponds to the purple densely dashed distribution in Figure 6.5b).

Figure 6.5: Simulation p-1c – ROLLO single-objective optimization to minimize AHTR plank's fuel-normalized power peaking factor (PPF_{fuel}). Input parameters varied: TRISO distribution ($\rho_{TRISO}(\vec{r})$). $PF_{total} = 0.0979$.



(a) Minimum, average, and maximum PPF_{fuel} evolution of the population in each generation.



(b) Plot of r_{top} and r_{bot} against PPF_{fuel} . Red crosses indicate the five reactor models with the lowest PPF_{fuel} .

Figure 6.6: Simulation p-1f – ROLLO single-objective optimization to minimize fuel-normalized power peaking factor (PPF_{fuel}) in the slab. Input parameters varied: coolant channel shape (r_{top}, r_{bot}). $PF_{total} = 0.0979$.

6.2.4 Single-Objective Optimization Discussion

In this section, I conduct an in-depth examination to understand and compare the driving factors for each individual objective.

Discussion: Minimize PF_{total} Objective

Simulation p-1a In Section 6.2.1's simulation p-1a, I conducted a single-objective optimization simulation to minimize the total fuel packing fraction (PF_{total}) by varying the PF_{total} and the TRISO distribution. In simulation p-1a, ROLLO found that an AHTR plank model with a $PF_{total} = 0.023$ and oscillating TRISO distribution most-minimized PF_{total} while meeting the $k_{eff} \geq 1.35$ constraint (Figure 6.1).

I ran a simulation for a constant TRISO distribution of $PF_{total} = 0.023$ and compared its fission reaction rate with the oscillating TRISO distribution to understand why the oscillating TRISO distribution enabled a lower PF_{total} . Figure 6.7 shows the TRISO distributions for the two compared simulations: Figure 6.1's most-minimized PF_{total} and the constant $PF_{total} = 0.023$.

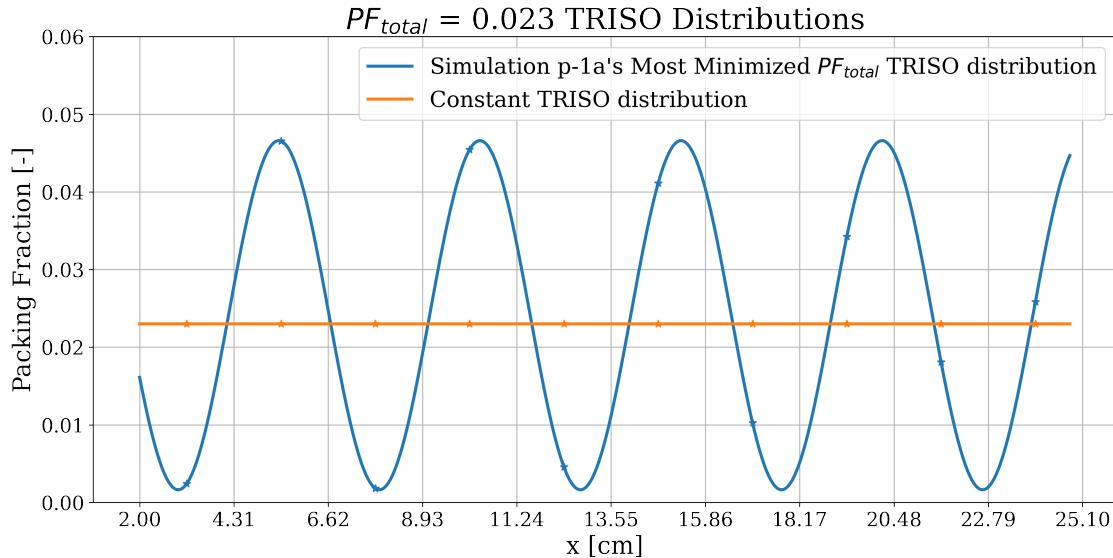


Figure 6.7: Simulation p-1a's most-minimized PF_{total} TRISO distribution (oscillating TRISO distribution) from Figure 6.1 and the constant $PF_{total} = 0.023$ TRISO distribution.

Table 6.9 compares the total fission reaction rate (OpenMC's `fission` tally) between the most-minimized PF_{total} TRISO distribution and a constant $PF_{total} = 0.023$ TRISO distribution (both shown in Figure 6.7). The most-minimized PF_{total} TRISO distribution has 0.18% higher

total fission reaction rate than the constant $PF_{total} = 0.023$ TRISO distribution, explaining why the oscillating TRISO distribution enabled a lower packing fraction for the same k_{eff} compared to the constant TRISO distribution.

Table 6.9: Total fission reaction rate comparison between the most-minimized PF_{total} TRISO distribution and a constant $PF_{total} = 0.023$ TRISO distribution. Both distributions shown in Figure 6.7. % Fission difference is the deviation between most-minimized and flat TRISO distributions' fission reaction rates.

Energy Group	Most-minimized PF_{total} Fission [reactions/src]	Flat PF_{total} Fission [reactions/src]	% Fission Diff
1	0.001260	0.001234	+2.07
2	0.006653	0.006658	-0.07
3	0.006566	0.006580	-0.21
4	0.556897	0.555889	+0.18
Total	0.542907	0.541924	+0.18

Simulation p-1d In Section 6.2.1's simulation p-1d, I conducted a single-objective optimization simulation to minimize the total fuel packing fraction (PF_{total}) by varying the PF_{total} and the coolant channel shape. In simulation p-1a, ROLLO found that there is no correlation between the PF_{total} and the coolant channel shape (demonstrated in Figure 6.2b).

Summary The minimize PF_{total} objective is driven by maximizing the total fission reaction rates. The minimize PF_{total} objective has correlations with the PF_{total} and TRISO packing fraction distribution input parameters. The objective has no correlation with the coolant channel shape input parameter.

Discussion: Minimize T_{max} Objective

Simulation p-1b In Section 6.2.2's simulation p-1b, I conducted a single-objective optimization simulation to minimize the maximum plank temperature (T_{max}) by varying the TRISO distribution. In simulation p-1b, ROLLO found that an AHTR plank model with a mostly flat TRISO distribution most-minimized T_{max} (Figure 6.3b).

I found that a fully flat TRISO distribution results in a $1.83K$ higher T_{max} . Figure 6.8 compares the Moltres-generated centerline temperature distribution for the plank with mostly flat

(Figure 6.3b) and flat TRISO distributions for the same $PF_{total} = 0.0979$.

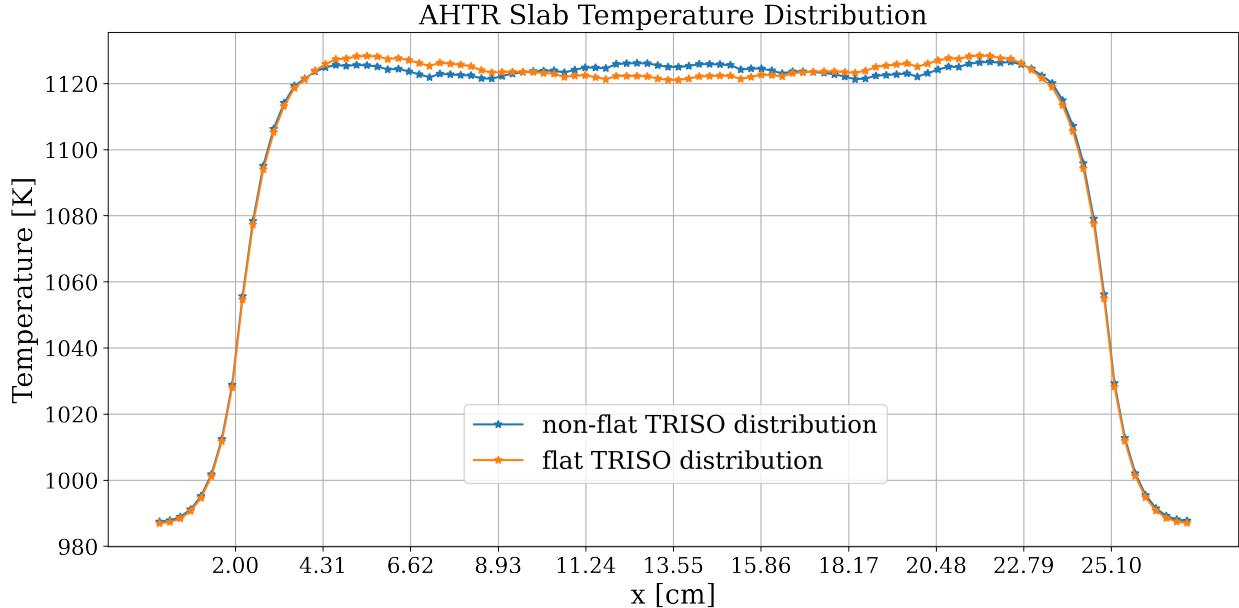


Figure 6.8: Comparison of Moltres-generated AHTR plank temperature distribution for non-flat and flat TRISO distribution. Both models have $PF_{total} = 0.0979$.

The AHTR plank with a flat TRISO distribution has higher plank temperatures on the left and right sides near the moderator. To combat this temperature peak, ROLLO found a TRISO distribution that has a slight dip near the moderator regions, resulting in a lower T_{max} . However, the peak temperature discrepancy between the mostly flat and fully flat TRISO distributions is very small. Thus, the minimize T_{max} objective mainly flattens the TRISO distribution.

Simulation p-1e In Section 6.2.2's simulation p-1e, I conducted a single-objective optimization simulation to minimize the maximum plank temperature (T_{max}) by varying the coolant channel shape. In simulation p-1e, ROLLO found that there is a negative linear correlation between the plank's T_{max} , and r_{bot} and r_{top} , shown in Figure 6.4b. Comparison of simulation p-1b and p-1e's results in Figures 6.3a and 6.4a show that the average T_{max} due to TRISO variation decreased by 60K over 10 generations, while average T_{max} due to the coolant channel shape variation only decreased by 15K over 10 generations. This demonstrates that during the genetic algorithm optimization process, variations in the coolant channel shape only minimize T_{max} by 15K, while variations in the TRISO distribution minimizes T_{max} by 60K. The TRISO distribution variations

have a higher impact on the T_{max} value compared to coolant channel variation, highlighting to reactor designers that for minimizing the T_{max} objective, they should pay more attention to the fuel density distribution in the AHTR design compared to coolant channel shape.

Summary The minimize T_{max} objective flattens the TRISO distribution and maximizes the coolant channel shape's r_{bot} and r_{top} to achieve the objective. The minimize T_{max} objective has correlations with all the input parameters: PF_{total} , TRISO packing fraction distribution, and coolant channel shape. The results from simulation p-1b and p-1e demonstrate that the minimize T_{max} objective is more influenced by the TRISO distribution than the coolant channel shape.

Discussion: Minimize PPF_{fuel} Objective

I conduct an equation analysis to determine the AHTR plank properties that influence the PPF_{fuel} objective. I then demonstrate the derived relationship in simulation p-1c results.

Equation Analysis Equation 6.1 shows the relationship between fission reaction rate, flux, and material properties:

$$RR_f = \Phi \times \sigma_f \times N \quad (6.1)$$

where

$$RR_f = \text{fission reaction rate } [\text{reactions} \cdot \text{cm}^{-3} \cdot \text{s}^{-1}]$$

$$\Phi = \text{neutron flux } [\text{neutrons} \cdot \text{cm}^{-2} \cdot \text{s}^{-1}]$$

$$\sigma_f = \text{microscopic cross section } [\text{cm}^2]$$

$$N = \text{atomic number density } [\text{atoms} \cdot \text{cm}^{-3}]$$

Since microscopic cross section is constant for the same fuel material, I rearrange Equation 6.1 into Equation 6.2:

$$\Phi \propto \frac{RR_f}{N} \quad (6.2)$$

In Section 5.3.3, I defined the fuel-normalized power peaking factor (PPF_{fuel}) as:

$$PPF_{fuel} = \frac{\max(\frac{fqr_j}{PF_j})}{\text{ave}(\frac{fqr_j}{PF_j})} \quad (6.3)$$

where

PPF_{fuel} = fuel-normalized power peaking factor

j = discretized fuel area j

fqr_j = fission-q-recoverable at position j (OpenMC tally)

PF_j = fuel packing fraction at position j

The fission reaction rate (RR_f) is proportional to fission energy production rate (fqr). The atomic number density (N) is proportional to the fuel packing fraction (PF). Thus, I can rearrange Equation 6.2 into Equation 6.4:

$$\Phi_j \propto \frac{fqr_j}{PF_j} \quad (6.4)$$

Finally, I can further rearrange Equation 6.4 into Equation 6.5:

$$\begin{aligned} \frac{\max(\Phi_j)}{\text{ave}(\Phi_j)} &\propto \frac{\max(\frac{fqr_j}{PF_j})}{\text{ave}(\frac{fqr_j}{PF_j})} \\ \frac{\max(\Phi_j)}{\text{ave}(\Phi_j)} &\propto PPF_{fuel} \end{aligned} \quad (6.5)$$

Therefore, from Equation 6.5, a flatter flux (smaller $\frac{\max(\Phi_j)}{\text{ave}(\Phi_j)}$ value) will result in a smaller PPF_{fuel} . Specifically, flatter energy group 4 thermal flux results in a smaller PPF_{fuel} (see Table 6.10). Table 6.10 shows the percentage contributions of fission reactions from each energy group from the AHTR plank model with constant $PF_{total} = 0.0979$ TRISO distribution shown in Figure 6.9. Most fission reactions are occurring in energy group 4.

In the following section, I demonstrate Equation 6.5's relationship in simulation p-1c results.

Table 6.10: Percentage of fission reactions from each energy group for AHTR plank model.

Energy Group	Energy Bounds [MeV]	% of Total FissionReactions
1	$9.1188 \times 10^{-3} < E < 2.0000 \times 10^1$	0.85
2	$2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$	4.85
3	$1.8554 \times 10^{-6} < E < 2.9023 \times 10^{-5}$	4.13
4	$1.0000 \times 10^{-12} < E < 1.8554 \times 10^{-6}$	90.16

Simulation p-1c In Section 6.2.3’s simulation p-1c, I conducted a single-objective optimization simulation to minimize the fuel-normalized power peaking factor (PPF_{fuel}) by varying TRISO distribution. In simulation p-1c, ROLLO found that for $PF_{total} = 0.0979$, an AHTR plank model with the TRISO distribution that peaks near the edges of the fuel region of the plank and a minimum point in plank’s center with a variation of ~ 0.07 , most-minimized PPF_{fuel} (Figure 6.5).

I ran a simulation for constant $PF_{total} = 0.0979$ and compared its flux to simulation p-1c’s most-minimized PPF_{fuel} TRISO distribution to understand why the latter enabled a lower PPF_{fuel} . Figure 6.9 shows the TRISO distributions and the PPF_{fuel} values for the two compared simulations: simulation p-1c’s (Figure 6.5) most-minimized PPF_{fuel} TRISO distribution and the constant $PF_{total} = 0.0979$ TRISO distribution.

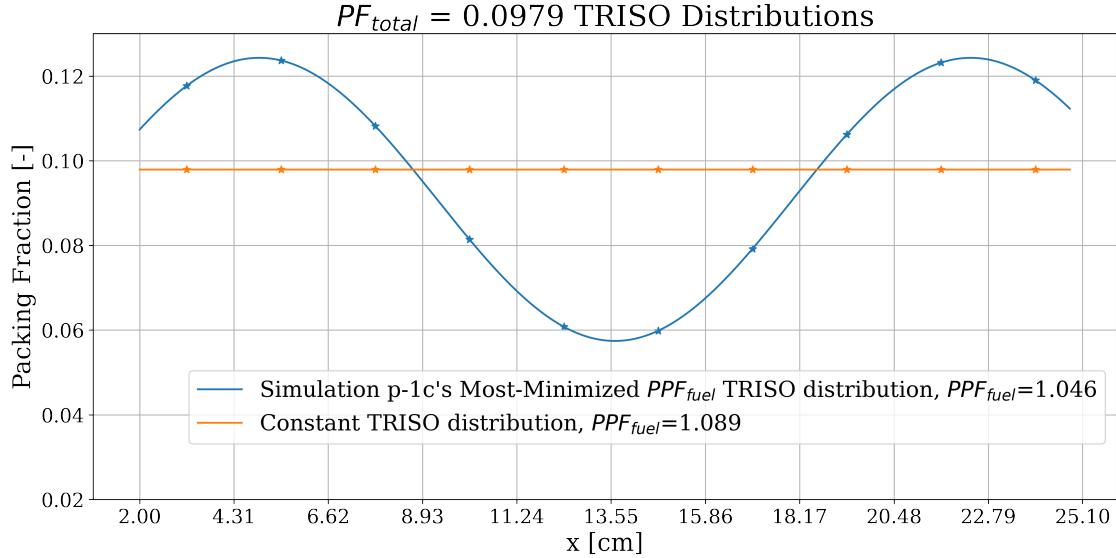


Figure 6.9: Simulation p-1c’s most-minimized PPF_{fuel} TRISO distribution from Figure 6.5 and the constant $PF_{total} = 0.0979$ TRISO distribution.

Figure 6.10 compares the flux distributions between the most-minimized PPF_{fuel} TRISO distribution and the constant $PF_{total} = 0.0979$ TRISO distribution (both shown in Figure 6.9). In

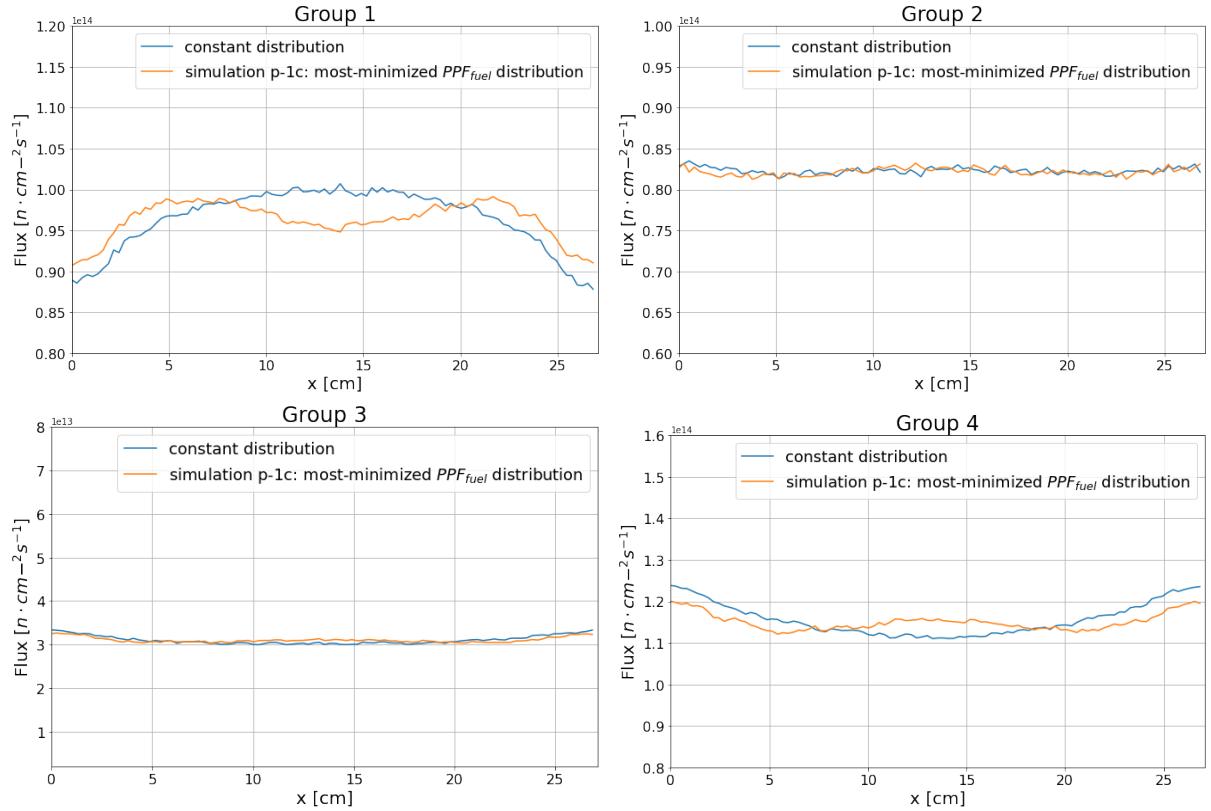


Figure 6.10: Flux comparison between Figure 6.5b's TRISO distribution that most-minimized PPF_{fuel} and a constant $PF_{total} = 0.0979$ TRISO distribution. AHTR plank's centerline neutron flux distribution in 4 groups at 948K. The centerline is the white line in Figure 5.12. Energy Group 1: $E > 9.1188 \times 10^{-3}$ MeV; Energy Group 2: $2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$ MeV; Energy Group 3: $1.8556 \times 10^{-5} < E < 2.9023 \times 10^{-5}$ MeV; Energy Group 4: $1.0 \times 10^{-12} < E < 1.8554 \times 10^{-6}$ MeV.

Figure 6.10, the constant TRISO distribution's Group 4 flux dips in the center of the plank due to spatial self-shielding effects. In the constant TRISO distribution's Group 1 flux, there is a peak in fast neutrons born in the plank's center, the fast neutrons are moderated in the graphite matrix and graphite structure (Figure 5.2). The self-shielding neutrons are more likely absorbed at the fuel regions at the plank's sides, near the pure graphite structure moderating regions. The outer sides of the plank absorb these neutrons and geometrically shield the plank's center from neutron flux, leading to a relatively lower group 4 thermal flux in the plank's center for the constant TRISO distribution.

Table 6.11 quantifies the flux comparison from Figure 6.10 between simulation p-1c's most-minimized PPF_{fuel} TRISO distribution and the constant $PF_{total} = 0.0979$ TRISO distribution (both shown in Figure 6.9).

Table 6.11: Flux value comparison between Figure 6.5b's TRISO distribution that most-minimized PPF_{fuel} and a constant $PF_{total} = 0.0979$ TRISO distribution. Energy Group 1: $E > 9.1188 \times 10^{-3}$ MeV; Energy Group 2: $2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$ MeV; Energy Group 3: $1.8556 \times 10^{-5} < E < 2.9023 \times 10^{-5}$ MeV; Energy Group 4: $1.0 \times 10^{-12} < E < 1.8554 \times 10^{-6}$ MeV.

Energy Group	$\max(\phi)/\min(\phi)$	$\max(\phi)/\min(\phi)$	% Diff
	Most-minimized PPF_{fuel}	Constant TRISO	
	TRISO Distribution	Distribution	
1	1.093	1.147	-4.68
2	1.024	1.027	-0.22
3	1.077	1.114	-3.32
4	1.071	1.115	-3.96

In energy group 4, the most-minimized PPF_{fuel} flux distribution is 3.96% flatter than the constant $PF_{total} = 0.0979$ flux distribution, resulting in a lower PPF_{fuel} (as shown by Equation 6.5). Therefore, for the same PF_{total} , a TRISO distribution that flattens thermal flux distribution results in a lower fuel-normalized power peaking factor (PPF_{fuel}), which is beneficial for fuel utilization.

Simulation p-1f In Section 6.2.3's simulation p-1f, I conducted a single-objective optimization simulation to minimize the fuel-normalized power peaking factor (PPF_{fuel}) by varying the coolant channel shape. Comparison of simulation p-1c and p-1f's results in Figures 6.5a and 6.6a show that the coolant channel shape variation has minimal impact on PPF_{fuel} as compared to TRISO

distribution variation: the average PPF_{fuel} due to TRISO variation decreased by 0.1 over 10 generations, while average PPF_{fuel} due to coolant channel shape variation decreased 0.003 over 4 generations. The minimal decrease in PPF_{fuel} over 10 generations in Figure 6.6b reinforces that there is no correlation between total radius and PPF_{fuel} .

Summary The minimize PPF_{fuel} objective is driven by flattening thermal (Group 4) flux distribution. The minimize PPF_{fuel} objective has correlations with the following input parameters: PF_{total} and TRISO packing fraction distribution. The objective has no correlation with the coolant channel shape input parameter.

6.2.5 Single-Objective Optimization Major Takeaways

Section 6.2 conducted an in-depth examination to understand and compare the driving factors for each individual objective. The minimize T_{max} objective flattens TRISO distribution and maximizes coolant channel shape's r_{top} and r_{bot} to achieve the objective. The minimize PF_{total} objective is driven by maximizing the plank's total fission reaction rate and influences the TRISO distribution to achieve the objective. The minimize PPF_{fuel} objective is driven by flattening the plank's thermal flux distribution and influences PF_{total} and TRISO distribution to achieve the objective. Both the minimize PF_{total} and minimize PPF_{fuel} objectives have no correlation with the coolant channel shape. Since the variations in coolant channel shape only impact the minimize plank's maximum temperature (T_{max}) objective, I do not conduct double-objective optimization for coolant channel shape variations.

6.3 AHTR Plank: Double-Objective Optimization Results

This section reports the AHTR plank's ROLLO double-objective optimization results. The previous section's single-objective optimization results inform the double-objective optimization simulations in this section. Section 6.2 concluded that coolant channel shape variations only correlate with the minimize T_{max} objective, thus, I do not conduct double-objective optimization for coolant channel shape variations. I run double-objective optimization simulations that vary total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), for combinations

of the three objectives (minimize PF_{total} , T_{max} , and PPF_{fuel}). Earlier in this chapter, Table 6.2 summarized the double-objective simulations in this section: p-2a, p-2b, and p-2c.

As previously described in Section 2.3, multi-objective optimization returns multiple optimal solutions that meet each objective to varying degrees; this set of solutions is the Pareto front [66]. For each solution in the Pareto front, none of the objective functions can be improved without degrading another objective. An ideal optimization method for a multi-objective problem, like reactor design optimization, should find widely spread solutions in the obtained Pareto front [66]. Thus, I report the optimal reactor models on the Pareto front for the multi-objective optimization problems in this section and Section 6.4. Several optimal solutions will be highlighted in the Pareto front figures.

To ensure that the multi-objective optimization problems are converged, I report the hypervolume values for each generation. As previously described in Section 4.4.2, the hypervolume indicator quantifies the Pareto front's goodness (bigger = better). The hypervolume is calculated by finding the volume between the reference point and the objective values of the Pareto front's reactor models. The reference point must be selected to ensure that every reactor model falls within the hypervolume; thus, I use a different reference point for each optimization problem. I use the hypervolume values to determine if the simulation converges across generations and do not compare them across different optimization simulations. The hypervolume is converging if the difference between generations' hypervolume values is getting smaller.

For a single generation with 64 reactor models, OpenMC and Moltres software must be run for each reactor model, resulting in high computational cost. For each optimization simulation, I must balance convergence and computational cost. ROLLO's purpose is to help the human reactor designer narrow down reactor design the search space, therefore, the reactor designer uses the computational power available to them, to narrow down the search space as much as possible. Section 4.5.1 described that the reactor designer should plot the Pareto front and observe the hypervolume values to evaluate if they are confident about the final solution set. Thus, I determine if convergence criteria is met by evaluating if the difference between generations' hypervolume values are getting smaller. If a multi-objective optimization problem's hypervolume converges earlier than the five generations I intended to run (determined in Section 5.5.2), I stop the simulation at that

generation. However, if the multi-objective optimization problem's hypervolume does not converge by generation 5, I run the problem for a few more generations until satisfactory convergence is achieved.

6.3.1 p-2a: Minimize PF_{total} and T_{max}

This section reports results from the double-objective optimization simulation p-2a; the objectives minimized are total fuel packing fraction (PF_{total}) and maximum plank temperature (T_{max}). Combining the minimize PF_{total} and T_{max} objectives enables us to observe if their driving factors result in any synergistic effects on the PF_{total} and $\rho_{TRISO}(\vec{r})$ input parameter variation. Table 6.12 summarizes simulation p-2a's optimization problem parameters.

Table 6.12: Simulation p-2a optimization problem parameters.

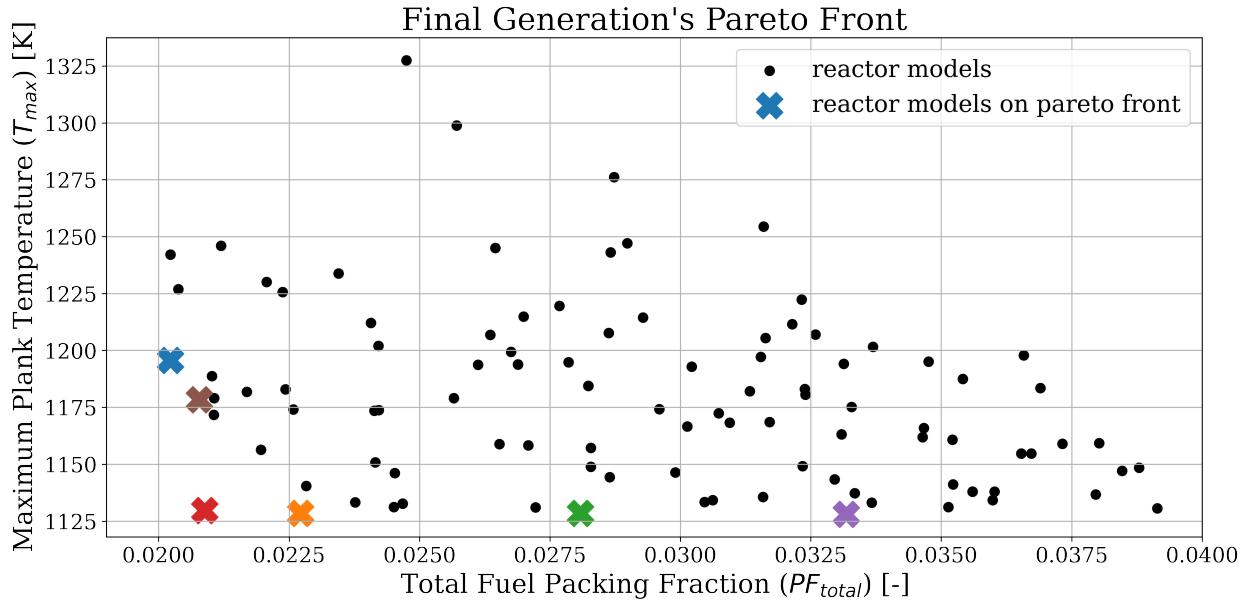
Two Objectives: Simulation p-2a	
Objectives	Minimize PF_{total} Minimize T_{max}
Input parameter variations	$0.02 \leq PF_{total} \leq 0.04$ $\rho_{TRISO}(\vec{r}): 0 < a < 2$ $\rho_{TRISO}(\vec{r}): 0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 < c < 2\pi$
Constraints	$k_{eff} \geq 1.35$
Genetic algorithm parameters	Population size: 128 Generations: 2

Table 6.13 shows each generation's hypervolume values, confirming that simulation p-2a is sufficiently converged by generation 2.

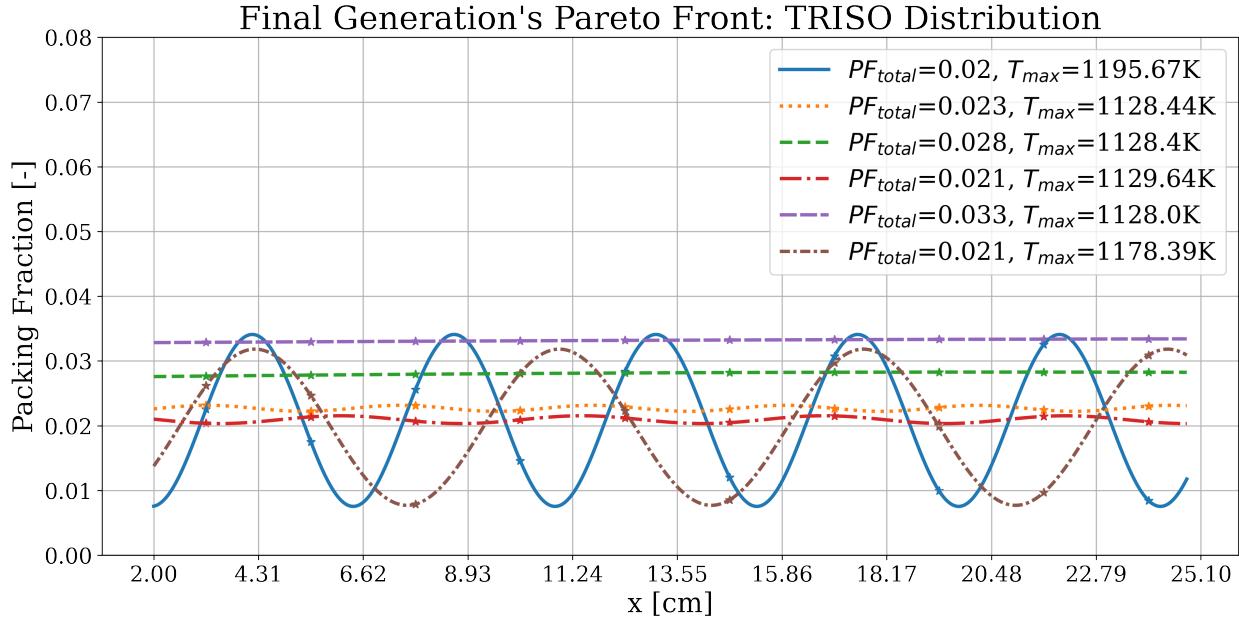
Table 6.13: Simulation p-2a hypervolume values at each generation.

Two Objectives: Simulation p-2a	
Reference point: (0.1, 1350)	
Generation	Hypervolume [-]
1	17.659
2	17.659

Figure 6.11a shows a plot of the final generation's PF_{total} against T_{max} ; crosses indicate the reactor models that fall on the Pareto front. Figure 6.11b shows the six TRISO distributions in the final generation that fall on the Pareto front. Figures 6.11c and 6.11d are two of the AHTR plank geometries that fall on the Pareto front.

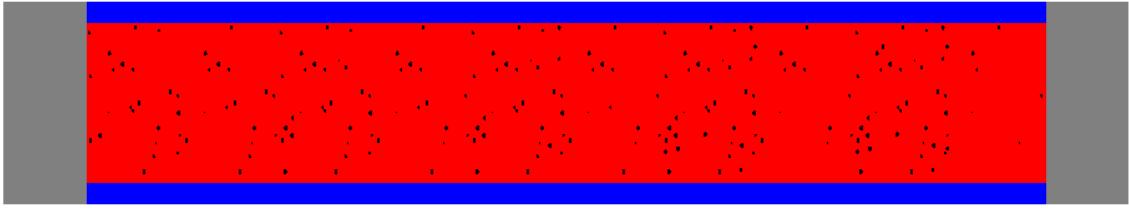


(a) Plot of final generation's reactor models' PF_{total} against T_{max} . Crosses indicate the reactor models on the Pareto front and the colors correspond to TRISO distributions in Figure 6.11b .

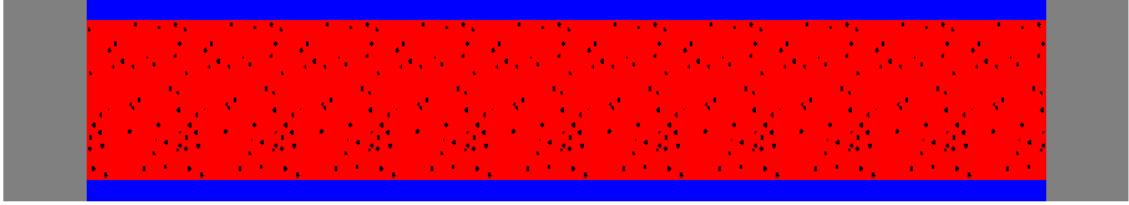


(b) TRISO packing fraction distribution for the six reactor models on the Pareto front.

Figure 6.11: Simulation p-2a – ROLLO double-objective optimization to minimize the total fuel packing fraction (PF_{total}) and maximum plank temperature (T_{max}). Input parameters varied: total fuel packing fraction (PF_{total}), TRISO distribution ($\rho_{TRISO}(\vec{r})$).



(c) AHTR plank model with the most-minimized PF_{total} (corresponds to the blue solid distribution in Figure 6.11b). $PF_{total} = 0.02$, $T_{max} = 1195.67K$.



(d) AHTR plank model with the most-minimized T_{max} (corresponds to the purple densely dashed distribution in Figure 6.11b). $PF_{total} = 0.033$, $T_{max} = 1128.0K$.

Figure 6.11: (contd.) Simulation p-2a – ROLLO double-objective optimization to minimize the total packing fraction (PF_{total}) and maximum plank temperature (T_{max}). Input parameters varied: total fuel packing fraction (PF_{total}), TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

In Figure 6.11a, ROLLO finds six widely spread reactor model solutions on simulation p-2a’s Pareto front. These TRISO distributions on the Pareto front that minimize both PF_{total} and T_{max} vary between the two extreme cases: most-minimized PF_{total} and most-minimized T_{max} . In Figure 6.11b, the plank model with the most-minimized PF_{total} and the highest T_{max} (the blue solid distribution) has an oscillating TRISO distribution (the corresponding geometry is illustrated in Figure 6.11c). In Figure 6.11b, the plank model with the most-minimized T_{max} and highest PF_{total} (the purple densely dashed distribution) has an almost constant TRISO distribution of $PF_{total} = 0.032$ across the plank (also illustrated in Figure 6.11d). Section 6.3.4 discusses and explains simulation p-2a’s results in context with the other double-objective optimization results.

6.3.2 p-2b: Minimize PF_{total} and PPF_{fuel}

This section reports results from the double-objective optimization simulation p-2b; the objectives minimized are total fuel packing fraction (PF_{total}) and fuel-normalized power peaking factor (PPF_{fuel}). Combining the minimize PF_{total} and PPF_{fuel} objectives enables us to observe if their driving factors result in any synergistic effects on the PF_{total} and $\rho_{TRISO}(\vec{r})$ input parameter

variation. Table 6.14 summarizes simulation p-2b's optimization problem parameters.

Table 6.14: Simulation p-2b optimization problem parameters.

Two Objectives: Simulation p-2b	
Objectives	Minimize PF_{total} Minimize PPF_{fuel}
Input parameter variations	$0.02 \leq PF_{total} \leq 0.04$ $\rho_{TRISO}(\vec{r}): 0 < a < 2$ $\rho_{TRISO}(\vec{r}): 0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 < c < 2\pi$
Constraints	$k_{eff} \geq 1.35$
Genetic algorithm parameters	Population size: 128 Generations: 3

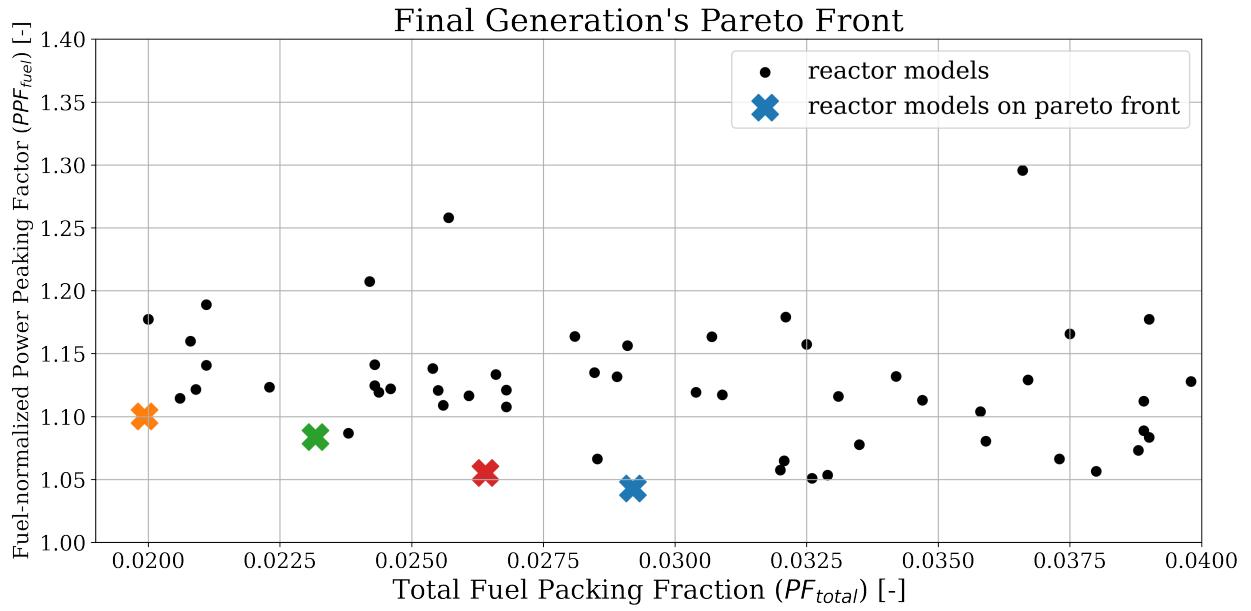
Table 6.15 shows each generation's hypervolume value, confirming that simulation p-2b is sufficiently converged by generation 3.

Table 6.15: Simulation p-2b hypervolume values at each generation.

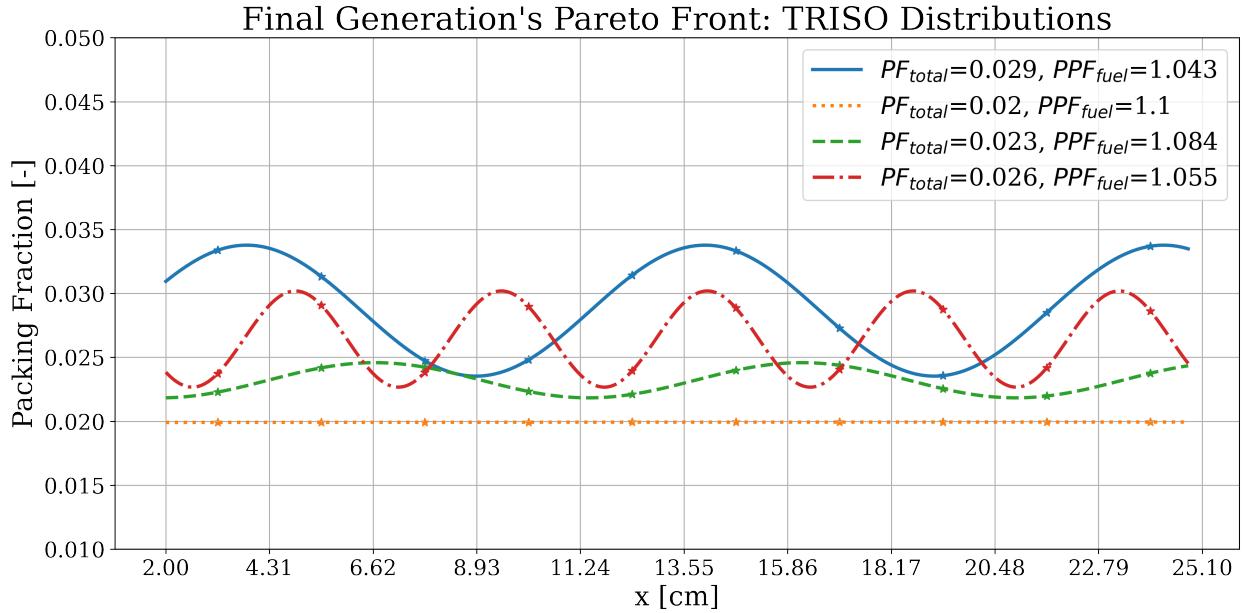
Two Objectives: Simulation p-2b	
Reference point: (0.1, 1.5)	
Generation	Hypervolume [-]
1	0.03607
2	0.03619
3	0.03625

Figure 6.12a shows a plot of the final generation's PF_{total} against PPF_{fuel} ; crosses indicate the reactor models that fall on the Pareto front. Figure 6.12b shows the four TRISO packing fraction distributions in the final generation that fall on the Pareto front. Figures 6.12c and 6.12d are two of the AHTR plank geometries that fall on the Pareto front.

All reactor models on Figure 6.12's Pareto front's PF_{total} values are between 0.02 and 0.029 and their TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$) are relatively flat, spatially varying between 0.02 and 0.04. In Figure 6.12b, the plank model with the most-minimized PF_{total} and highest PPF_{fuel} (the orange distribution) has a constant TRISO distribution of $PF_{total} = 0.02$ across the plank (the corresponding geometry is illustrated in Figure 6.12c). In Figure 6.12b, the plank model with the most-minimized PPF_{fuel} and highest PF_{total} (the blue distribution) has a TRISO distribution that peaks in the center of the plank and on both sides (the corresponding geometry is illustrated in Figure 6.12d). Section 6.3.4 discusses and explains simulation p-2b's

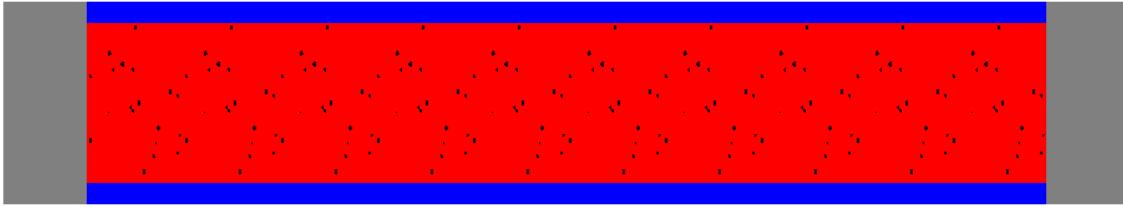


(a) Plot of final generation's reactor models' PF_{total} against PPF_{fuel} . Crosses indicate the reactor models on the Pareto front. Cross colors correspond to TRISO distributions in the plot below.

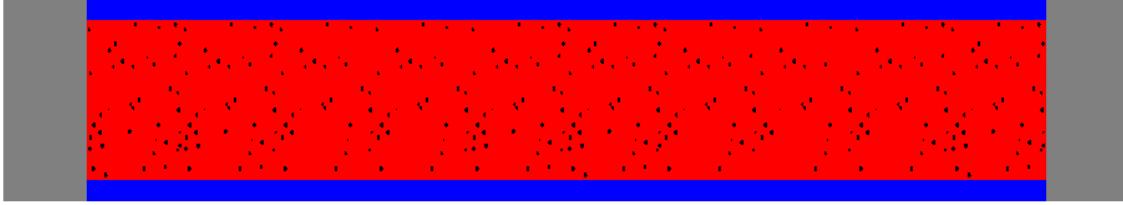


(b) TRISO distribution for the four reactor models on the Pareto front.

Figure 6.12: Simulation p-2b – ROLLO double-objective optimization to minimize total fuel packing fraction (PF_{total}) and normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



(c) AHTR plank model with the most-minimized PF_{total} (corresponds to the orange dotted distribution in Figure 6.12b).



(d) AHTR plank model with the most-minimized PPF_{fuel} (corresponds to the blue solid distribution in Figure 6.12b).

Figure 6.12: (contd.) Simulation p-2b – ROLLO double-objective optimization to minimize total packing fraction (PF_{total}) and fuel-normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

results in context with the other double-objective optimization results.

6.3.3 p-2c: Minimize T_{max} and PPF_{fuel}

This section reports results from the double-objective optimization simulation p-2c; the objectives minimized are maximum plank temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}). Combining the minimize T_{max} and PPF_{fuel} objectives enables us to observe if their driving factors result in any synergistic effects on the $\rho_{TRISO}(\vec{r})$ input parameter variation. Table 6.16 summarizes simulation p-2c’s optimization problem parameters. Unlike simulations p-2a and p-2b, I did not vary PF_{total} and held it constant at 0.0979. Section 6.1 described that for the optimization simulations which do not vary PF_{total} , I use $PF_{total} = 0.0979$, which corresponds to the FHR benchmark’s PF_{total} . This PF_{total} is higher than the PF_{total} variation utilized in simulation p-2a and p-2b of 0.02 to 0.04. I chose to use a lower range for PF_{total} variations, I observed that the TRISO spatial variation enabled comparable k_{eff} values with lower PF_{total} values.

Table 6.17 shows each generation’s hypervolume value, confirming that simulation p-2c is sufficiently converged by generation 3.

Table 6.16: Simulation p-2c optimization problem parameters.

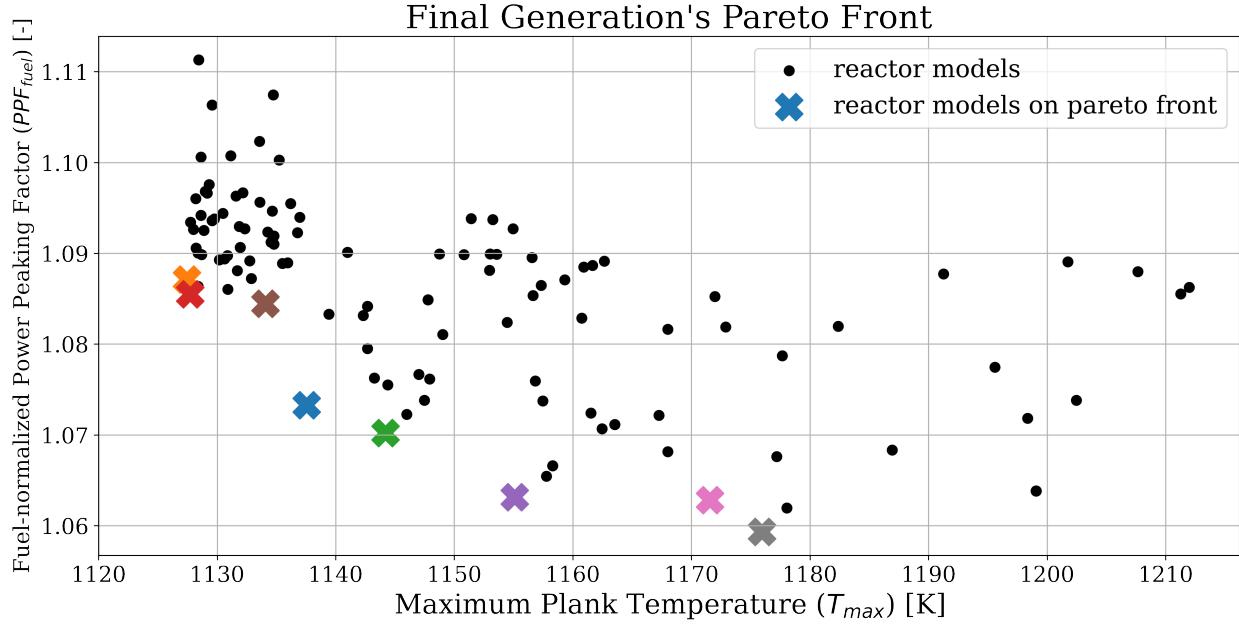
Two Objectives: Simulation p-2c	
Objectives	Minimize T_{max} Minimize PPF_{fuel}
Input parameter variations	$\rho_{TRISO}(\vec{r})$: $0 < a < 2$ $\rho_{TRISO}(\vec{r})$: $0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r})$: $0 < c < 2\pi$
Constraints	$k_{eff} \geq 1.0$ $PF_{total} = 0.0979$
Genetic algorithm parameters	Population size: 120 Generations: 3

Table 6.17: Simulation p-2c hypervolume values at each generation.

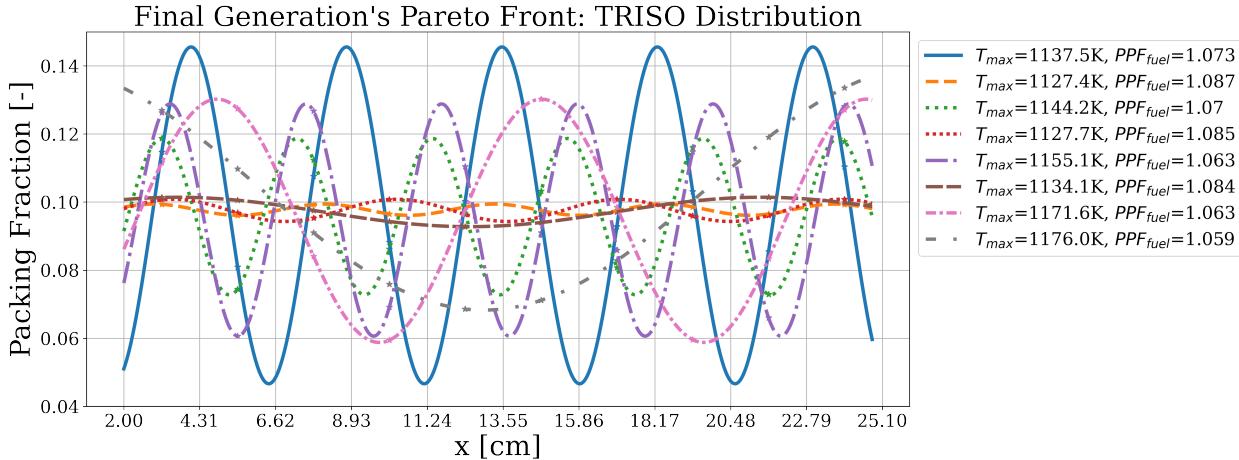
Two Objectives: Simulation p-2c	
Reference point: (1300, 1.6)	
Generation	Hypervolume [-]
1	92.041
2	92.087
3	92.751

Figure 6.13a shows a plot of the final generation’s reactor models’ T_{max} against PPF_{fuel} ; crosses indicate the reactor models that fall on the Pareto front. Figure 6.13b shows the eight TRISO distributions in the final generation that fall on the Pareto front. Figures 6.13c and 6.13d are two of the AHTR plank geometries that fall on the Pareto front.

In Figure 6.13a, ROLLO finds eight widely spread reactor model solutions on simulation p-2c’s Pareto front. These TRISO distributions on the Pareto front that minimize both T_{max} and PPF_{fuel} vary between the two extreme cases: most-minimized T_{max} and most-minimized PPF_{fuel} . In Figure 6.13b, the plank model with the most-minimized T_{max} and highest PPF_{fuel} (the orange dashed distribution) has an almost constant TRISO distribution of $PF_{total} = 0.0979$ across the plank, with a slight oscillating pattern (the corresponding geometry is illustrated in Figure 6.13c). In Figure 6.13b, the plank model with the most-minimized PPF_{fuel} and highest T_{max} (the grey densely dotted distribution) has a TRISO distribution that peaks near the plank’s sides and has a minimum point at the plank’s center (the corresponding geometry is illustrated in Figure 6.13d). Section 6.3.4 discusses and explains simulation p-2c’s results in context with the other double-objective optimization results.

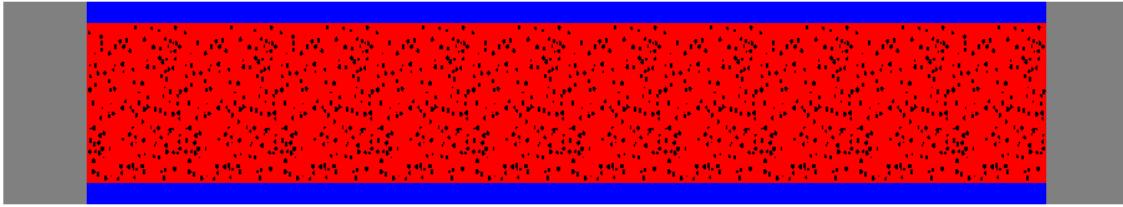


(a) Plot of final generation's reactor models' T_{max} against PPF_{fuel} . Crosses indicate the reactor models on the Pareto front. Cross colors correspond to TRISO distributions in the plot below.

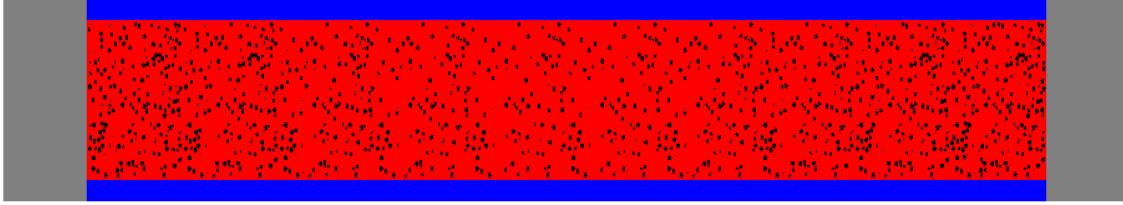


(b) TRISO packing fraction distribution for the eight reactor models on the Pareto front.

Figure 6.13: Simulation p-2c – ROLLO double-objective optimization to minimize the maximum plank temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$). $PF_{total} = 0.0979$.



(c) AHTR plank model with the most-minimized T_{max} (corresponds to the orange dashed distribution in Figure 6.13b).



(d) AHTR plank model with the most-minimized PPF_{fuel} (corresponds to the grey densely dotted distribution in Figure 6.13b).

Figure 6.13: (contd.) Simulation p-2c – ROLLO double-objective optimization to minimize the maximum plank temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$). $PF_{total} = 0.0979$.

6.3.4 Double-Objective Optimization Discussion

ROLLO successfully found a wide spread of reactor models in each of the double-objective optimization simulation's final generation Pareto fronts. In this section, I explain how the driving factors and phenomena observed in the previous single-objective discussion (Section 6.2.4) combine to result in the optimal reactor models found by the double-objective optimization simulations.

Simulation p-2a

In Section 6.3.1's simulation p-2a, I conducted a double-objective optimization simulation to minimize the total fuel packing fraction (PF_{total}) and maximum plank temperature (T_{max}) by varying the total fuel packing fraction (PF_{total}) and the TRISO distribution. ROLLO found six widely spread reactor models on simulation p-2a's Pareto front (Figure 6.11).

Figure 6.14 compares the TRISO distribution's of simulation p-2a's most-minimized PF_{total} reactor model and simulation p-1a's most-minimized PF_{total} reactor model.

Figure 6.14 shows that simulation p-2a and p-1a AHTR plank models' that most-minimized

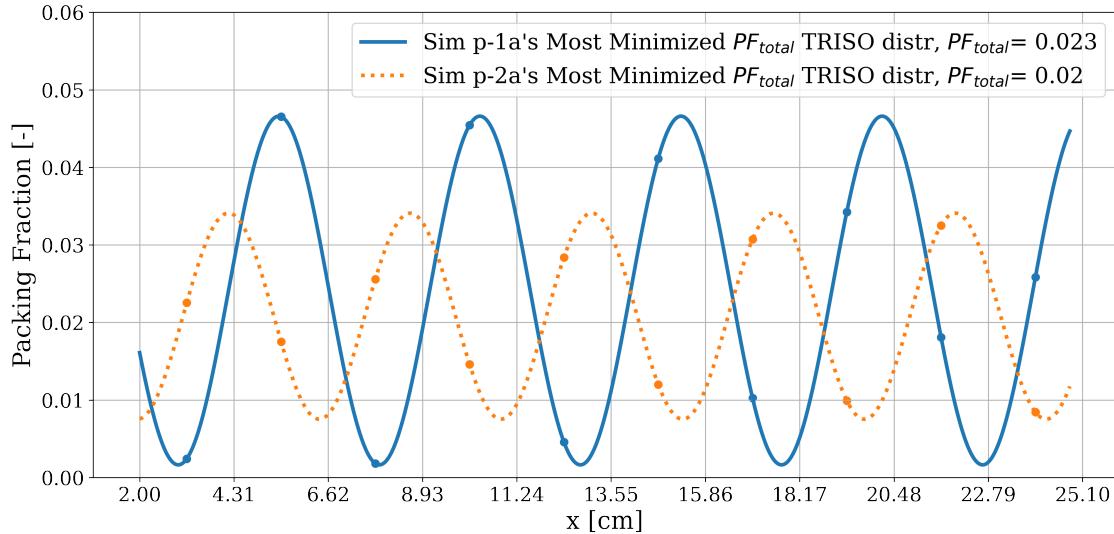


Figure 6.14: Simulation p-2a’s most-minimized PF_{total} TRISO distribution (blue solid distribution) and simulation p-1a’s most-minimized PF_{total} TRISO distribution (orange dotted distribution).

PF_{total} objective have similar oscillating TRISO distributions. Their differences lie in the TRISO distribution ranges, simulation p-2a’s distribution varies between 0.005 and 0.035, while simulation p-1a’s distribution varies between 0 and 0.4, due to influences from the minimize T_{max} objective, which influences the flattening of the TRISO distribution.

Figure 6.15 compares the TRISO distribution’s of simulation p-2a’s most-minimized T_{max} reactor model and simulation p-1b’s most-minimized T_{max} reactor model. Figure 6.15 shows that simulation p-2a and p-1b AHTR plank models’ that most-minimized T_{max} objective have similar flat TRISO distribution shapes, as the minimize T_{max} objective influences the flattening of the TRISO distribution. However, they have different PF_{total} values because simulation p-1b held a constant $PF_{total} = 0.0979$, while simulation p-2a varied PF_{total} between 0.02 and 0.04.

The TRISO distributions on simulation p-2a’s Pareto front in Figure 6.11 minimize both PF_{total} and T_{max} , they vary between the two extreme cases: most-minimized PF_{total} and most-minimized T_{max} . The minimize T_{max} objective influences the TRISO distribution’s flatness, while the minimize PF_{total} objective influences the oscillating pattern. Section 6.2.4 described both influences in detail.

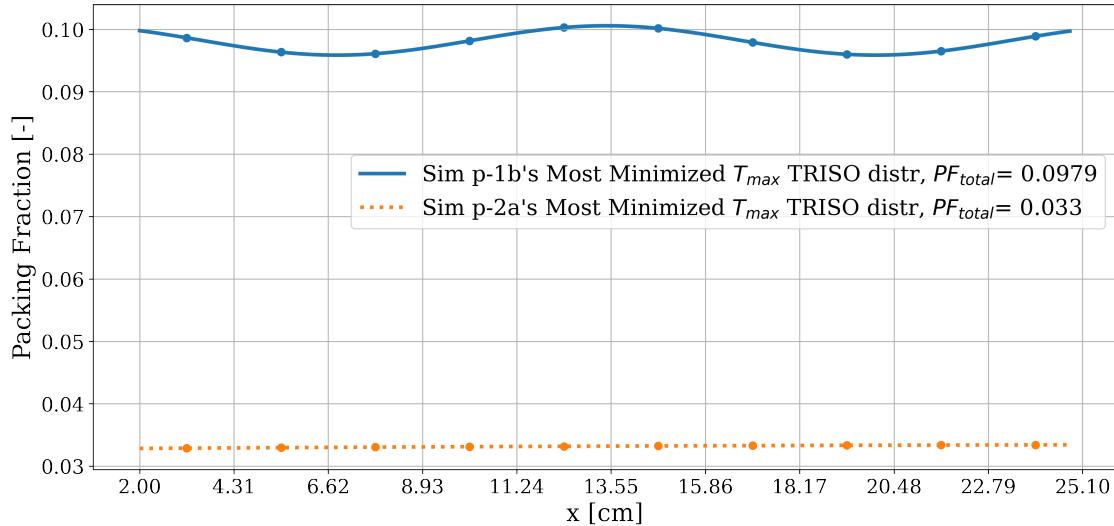


Figure 6.15: Simulation p-2a’s most-minimized T_{max} TRISO distribution (blue solid distribution) and simulation p-1b’s most-minimized T_{max} TRISO distribution (orange dotted distribution).

Simulation p-2b

In Section 6.3.2’s simulation p-2b, I conducted a double-objective optimization simulation to minimize the total fuel packing fraction (PF_{total}) and fuel-normalized power peaking factor (PPF_{fuel}) by varying total fuel packing fraction (PF_{total}) and TRISO distribution. ROLLO found four widely spread reactor models on simulation p-2b’s Pareto front (Figure 6.12).

Figure 6.16 compares the TRISO distribution’s of simulation p-2b’s most-minimized PF_{total} reactor model and simulation p-1a’s most-minimized PF_{total} reactor model.

Figure 6.16 shows that simulation p-2b’s plank model with the most-minimized PF_{total} has a constant TRISO distribution which differs from simulation p-1a’s most-minimized PF_{total} oscillating TRISO distribution. Simulation p-1a’s most-minimized PF_{total} oscillating TRISO distribution has a high $PPF_{fuel} = 1.363$ compared to the constant TRISO distribution’s $PPF_{fuel} = 1.1$. The influences from the minimize PPF_{fuel} objective explains why for simulation p-2b, ROLLO found that the plank model that most-minimized PF_{total} has a constant TRISO distribution instead of an oscillating TRISO distribution.

Figure 6.17 compares the TRISO distribution’s of simulation p-2b’s most-minimized PPF_{fuel} reactor model and simulation p-1c’s most-minimized PPF_{fuel} reactor model.

Figure 6.17 shows that simulation p-2b’s plank model with the most-minimized PPF_{fuel} (or-

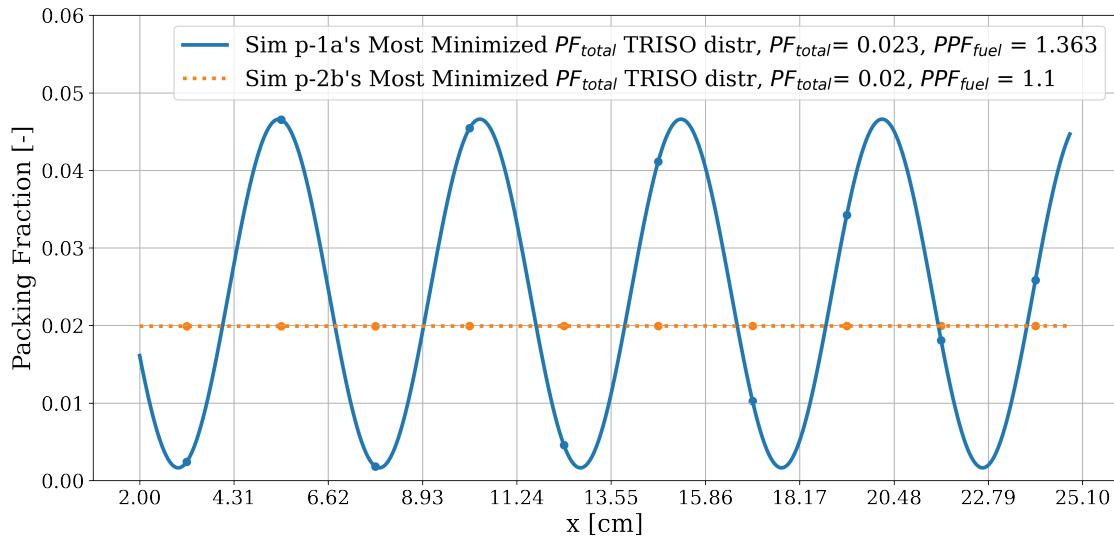


Figure 6.16: Simulation p-2b's most-minimized PF_{total} TRISO distribution (blue solid distribution) and simulation p-1a's most-minimized PF_{total} TRISO distribution (orange dotted distribution).

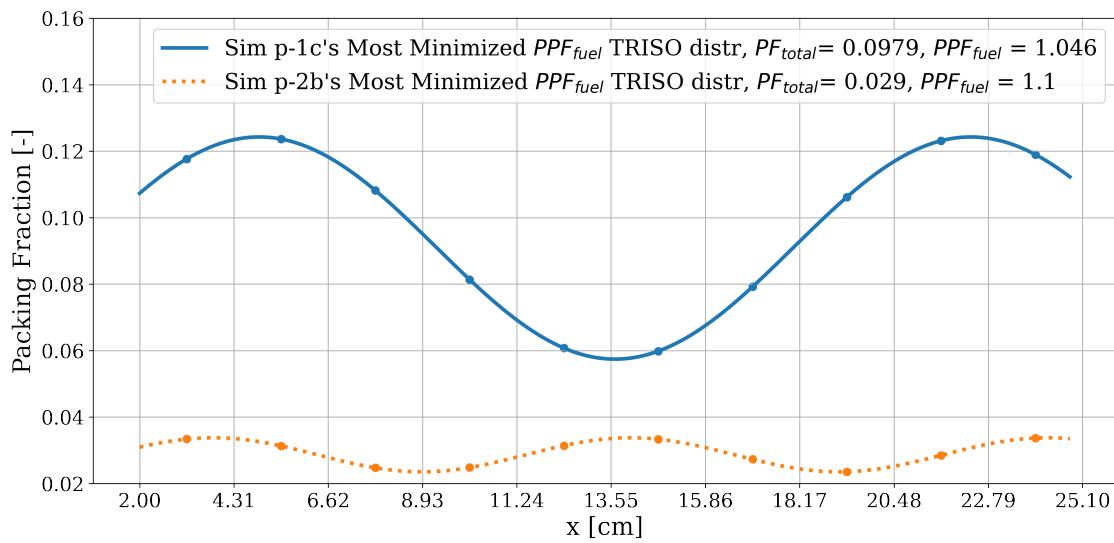


Figure 6.17: Simulation p-2b's most-minimized PPF_{fuel} TRISO distribution (blue solid distribution) and simulation p-1c's most-minimized PPF_{fuel} TRISO distribution (orange dotted distribution).

angled dotted distribution) has a TRISO distribution that peaks in the plank's center and both sides with a ~ 0.02 TRISO spatial variation, which differs from simulation p-1c's most-minimized PPF_{fuel} TRISO distribution (blue solid distribution) which has a large ~ 0.07 TRISO spatial variation and a minimum point in the center of the plank.

I ran a simulation that used simulation p-2b's $PF_{total} = 0.029$ and simulation p-1c's most-minimized PPF_{fuel} TRISO distribution. I compared its flux to simulation p-2b's most-minimized PPF_{fuel} TRISO distribution. Figure 6.18 shows the TRISO distributions with their PPF_{fuel} values for the two compared simulations: simulation p-2b's (Figure 6.12) most-minimized PPF_{fuel} TRISO distribution and simulation p-1c's most-minimized PPF_{fuel} TRISO distribution with $PF_{total} = 0.029$.

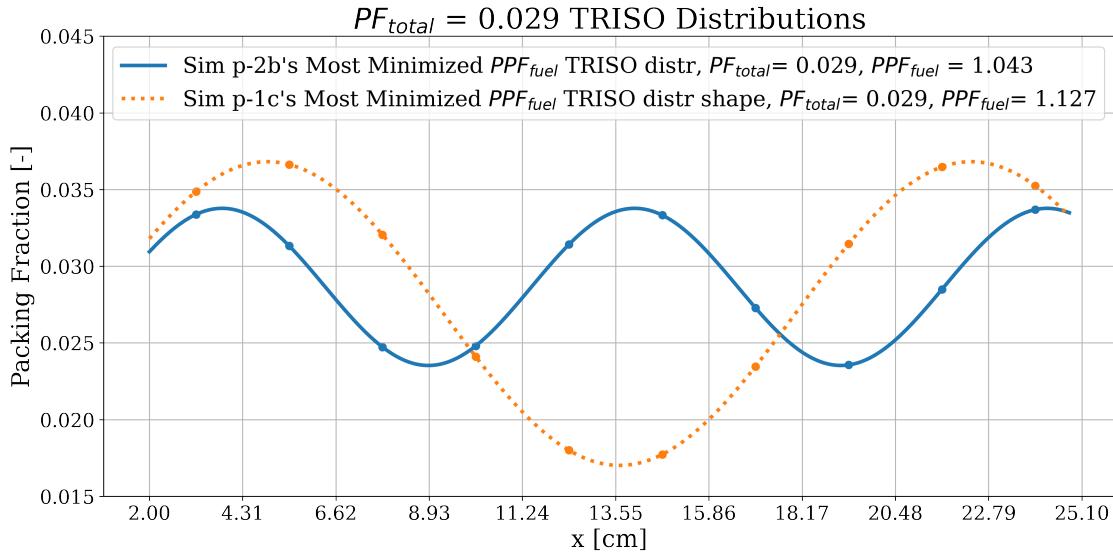


Figure 6.18: Simulation p-2b's most-minimized PPF_{fuel} TRISO distribution from Figure 6.12 and the constant $PF_{total} = 0.0292$ TRISO distribution.

Figure 6.18 shows that for the same $PF_{total} = 0.0292$, simulation p-2b's most-minimized PPF_{fuel} TRISO distribution has a lower PPF_{fuel} than the simulation p-1c's most-minimized PPF_{fuel} TRISO distribution shape. This explains why ROLLO found that the plank model that most-minimized PPF_{fuel} has a TRISO distribution with smaller spatial variations.

Unlike simulations p-2a and p-2c, simulation p-2b's extreme most-minimized PF_{total} and most-minimized PPF_{fuel} do not follow similar results as their single-objective counterparts. The results from simulation p-2b and investigations in Section 6.2.4 suggest that the minimize PF_{total} objec-

tive's driving factor maximize total fission reaction rate and minimize the PPF_{fuel} objective's driving factor flattening thermal flux distribution influence each other resulting in unexpected TRISO distributions at different PF_{total} values. Reactor designers participating in further optimization of the AHTR design will benefit from awareness of these objectives' relationship.

Simulation p-2c

In Section 6.3.3's simulation p-2c, I conducted a double-objective optimization simulation to minimize the maximum plank temperature (T_{max}) and the fuel-normalized power peaking factor (PPF_{fuel}) by varying TRISO distribution. ROLLO found eight widely spread reactor models on simulation p-2c's Pareto front.

Figure 6.19 compares the TRISO distribution's of simulation p-2c's most-minimized T_{max} reactor model and simulation p-1b's most-minimized T_{max} reactor model. Figure 6.19 shows that simulation p-2c and p-1b AHTR plank models' that most-minimized T_{max} objective have somewhat similar flat slightly TRISO distributions.

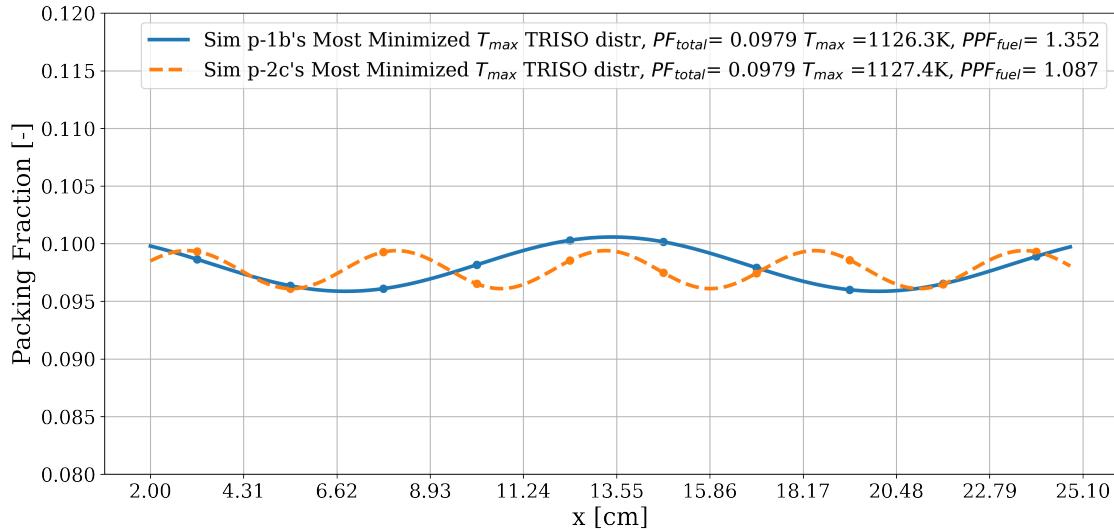


Figure 6.19: Simulation p-2c's most-minimized T_{max} TRISO distribution (blue solid distribution) and simulation p-1b's most-minimized T_{max} TRISO distribution (orange dotted distribution).

Figure 6.20 compares the TRISO distribution's of simulation p-2c's most-minimized PPF_{fuel} reactor model and simulation p-1c's most-minimized PPF_{fuel} reactor model. Figure 6.20 shows that simulation p-2c and p-1c AHTR plank models' that most-minimized PPF_{fuel} objective have

somewhat similar TRISO distributions with peaks near the sides of the plank and a minimum point at the plank's center.

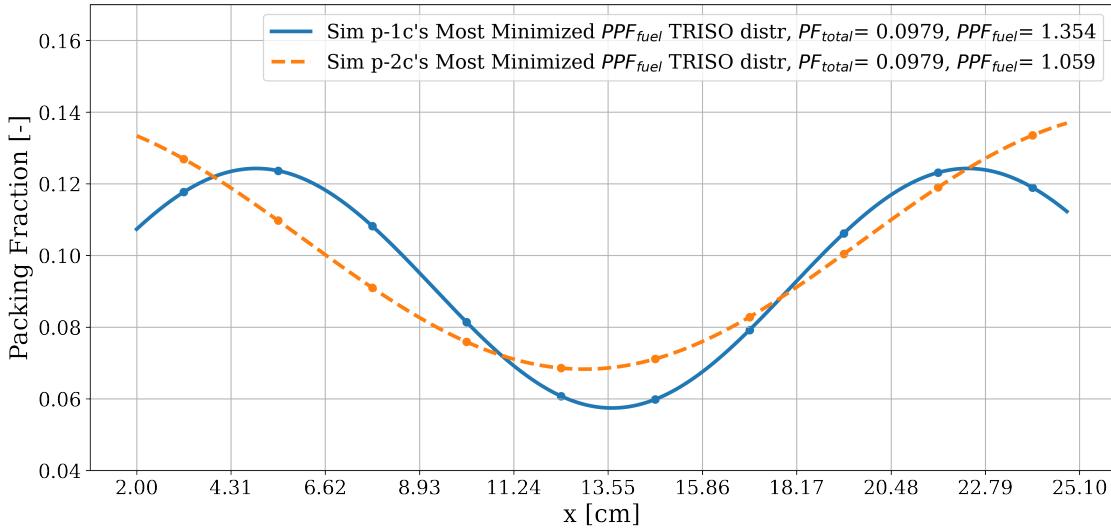


Figure 6.20: Simulation p-2c's most-minimized PPF_{fuel} TRISO distribution (blue solid distribution) and simulation p-1c's most-minimized PPF_{fuel} TRISO distribution (orange dotted distribution).

The TRISO distributions on simulation p-2c's Pareto front in Figure 6.13 minimize both T_{max} and PPF_{fuel} , they vary between the two extreme cases: most-minimized T_{max} and most-minimized PPF_{fuel} . The minimize T_{max} objective influences the TRISO distribution's flatness, while the minimize PPF_{fuel} objective influences the oscillating pattern. Section 6.2.4 described both influences in detail.

6.3.5 Double-Objective Optimization Major Takeaways

The driving factors and influences from each objective come together in each double-objective optimization simulation to give a set of optimal reactor models on a Pareto front. In the double-objective optimization simulations, the minimize T_{max} objective continued to influence the flattening of the TRISO distribution. The results from simulation p-2b suggests that the minimize PF_{total} objective's driving factor maximize total fission reaction rate and minimize PPF_{fuel} objective's driving factor flattening thermal flux distribution influence each other resulting in unexpected TRISO distributions at different PF_{total} values.

6.4 AHTR Plank: Triple-Objective Optimization Results

This section reports the AHTR plank's ROLLO triple-objective optimization results. I run two triple-objective optimization simulations that optimize all the objectives (minimize PF_{total} , T_{max} , and PPF_{fuel}). The first simulation varies total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$). The second simulation varies PF_{total} , $\rho_{TRISO}(\vec{r})$, and coolant channel shape. Table 6.2 summarized the triple-objective simulations in this section: p-3a, and p-3b. The following two subsections describing the optimization results are distinguished by the input parameters varied, since both simulations optimize for all three objectives.

6.4.1 p-3a: Variation of PF_{total} and $\rho_{TRISO}(\vec{r})$

This section reports results from the triple-objective optimization simulation p-3a, with all objectives minimized: total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}). The input parameters varied are total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$). Table 6.18 summarizes simulation p-3a's optimization problem parameters.

Table 6.18: Simulation p-3a optimization problem parameters.

Three Objectives: Simulation p-3a	
Objectives	Minimize PF_{total} Minimize T_{max} Minimize PPF_{fuel}
Input parameter variations	$0.02 \leq PF_{total} \leq 0.04$ $\rho_{TRISO}(\vec{r}): 0 < a < 2$ $\rho_{TRISO}(\vec{r}): 0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 < c < 2\pi$
Constraints	$k_{eff} \geq 1.35$
Genetic algorithm parameters	Population size: 128 Generations: 5

Table 6.19 shows each generation's hypervolume value, confirming that simulation p-3a is sufficiently converged by generation 5.

Table 6.19: Simulation p-3a hypervolume values at each generation.

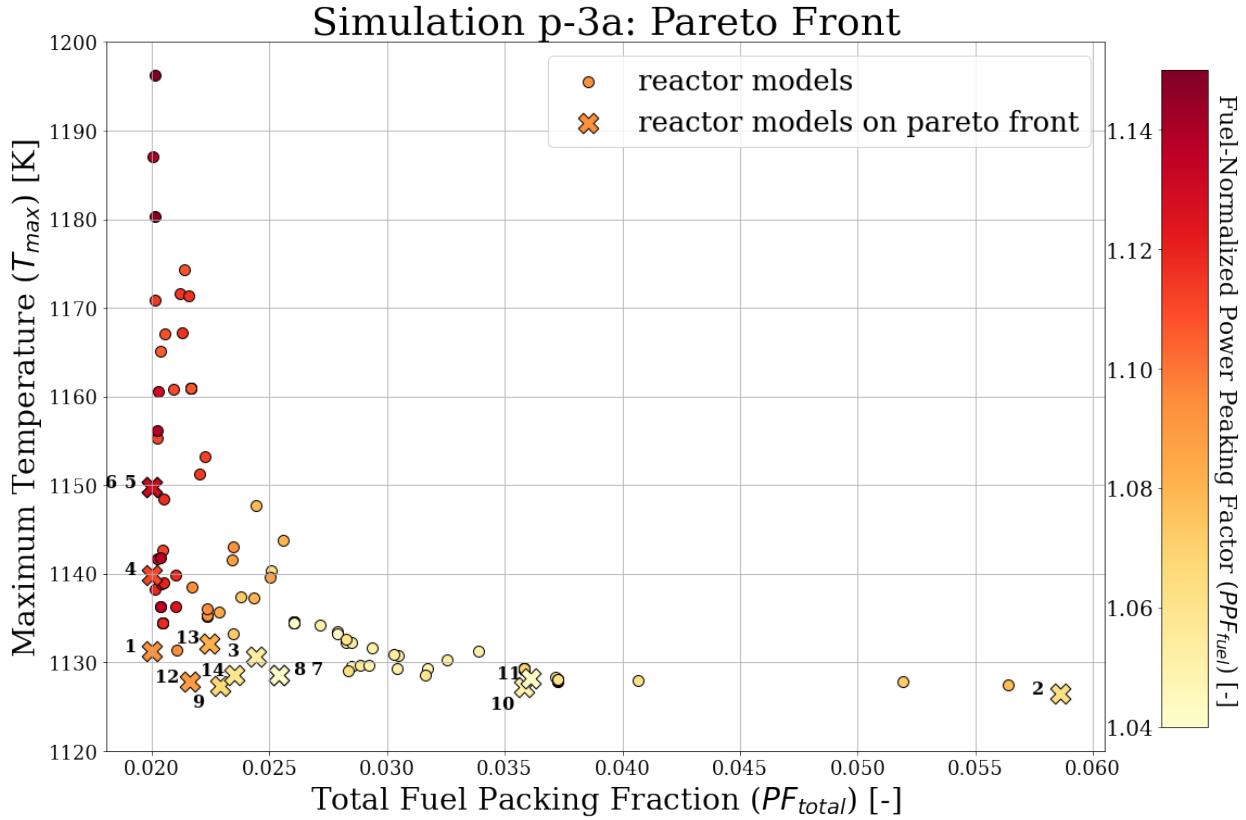
Three Objectives: Simulation p-3a	
Reference point: (0.06, 1260 ,1.5)	
Generation	Hypervolume [-]
1	2.392
2	2.392
3	2.393
4	2.442
5	2.446

Figure 6.21a shows a plot of the final generation’s reactor models’ PF_{total} against T_{max} against PPF_{fuel} ; crosses indicate the reactor models that fall on the Pareto front. Figure 6.21b shows the 14 TRISO packing fraction distributions in the final generation, labeled numerically, that fall on the Pareto front. Figure 6.21a demonstrates that ROLLO found 14 widely spread out reactor models in the final generation’s Pareto front. Figure 6.22a shows the three distributions from Figure 6.21b that most-minimized each objective, respectively. Figures 6.22b, 6.22c, and 6.22d show the AHTR plank’s TRISO distribution and coolant channel shape for most-minimized cases.

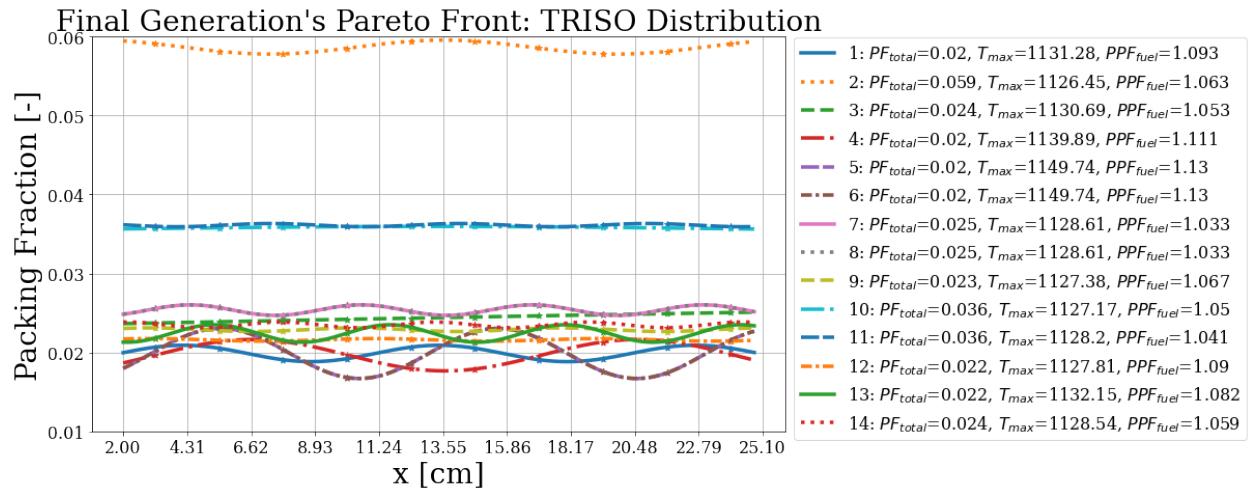
In Figure 6.22a, reactor model 5 with the most-minimized PF_{total} (the purple dashed distribution) has an oscillating TRISO distribution with ~ 0.01 TRISO spatial variation and $PF_{total} = 0.02$ (the corresponding geometry is illustrated in Figure 6.22b). In Figure 6.22a, reactor model 2 with the most-minimized T_{max} (the orange dotted distribution) has an almost constant TRISO distribution of $PF_{total} = 0.059$ (the corresponding geometry is illustrated in Figure 6.22c). In Figure 6.22a, reactor model 7 with the most-minimized PPF_{fuel} (the pink solid distribution) has a slightly oscillating TRISO distribution and $PF_{total} = 0.025$ (the corresponding geometry is illustrated in Figure 6.22d). Section 6.4.3 discusses and explains simulation p-3a’s results.

6.4.2 p-3b: Variation of PF_{total} , $\rho_{TRISO}(\vec{r})$, and Coolant Channel Shape

This section reports results from the triple-objective optimization simulation p-3b, the largest and final optimization problem run for the AHTR plank model. Simulation p-3b minimized all the objectives: total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}), and varied all the input parameters: total fuel packing fraction (PF_{total}), TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), and coolant channel shape.

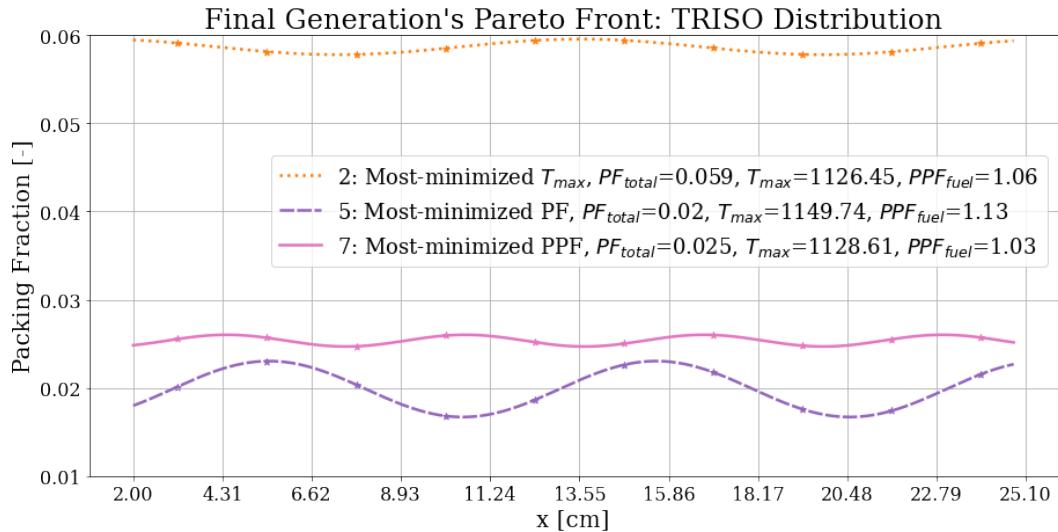


(a) Plot of final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} as a color dimension. Crosses indicate the reactor models on the Pareto front. Cross numbering correspond to TRISO distributions in Figure 6.21b.

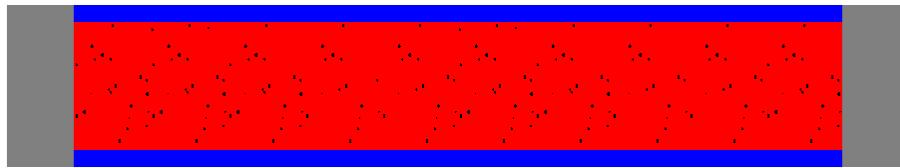


(b) TRISO packing fraction distribution for the 14 reactor models on the Pareto front. Numbered reactor models correspond to numbered crosses in Figure 6.21a.

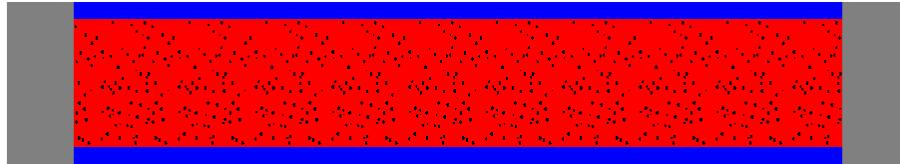
Figure 6.21: Simulation p-3a – ROLLO triple-objective optimization to minimize the total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



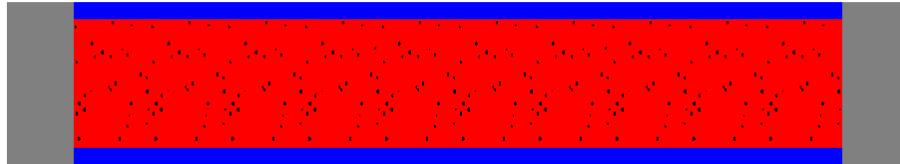
(a) TRISO distributions for the reactor models on the Pareto front that most-minimized each objective.



(b) AHTR plank model with the most-minimized PF_{total} (corresponds to reactor model 5, the purple dashed distribution in Figure 6.22a.)



(c) AHTR plank model with the most-minimized T_{max} (corresponds to reactor model 2, the orange dotted distribution in Figure 6.22a.)



(d) AHTR plank model with the most-minimized PPF_{fuel} (corresponds to reactor model 7, the pink solid distribution in Figure 6.22a.)

Figure 6.22: AHTR plank models and TRISO distribution for the 3 reactor models on simulation p-3a's Pareto front that most-minimized each objective. Simulation p-3a – ROLLO triple-objective optimization to minimize the total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}) and normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$)

Previous optimization simulations varied combinations of the objectives and input parameters, but not all together. Simulation p-3b ran for 1425 node-hours on the Theta supercomputer with 8 generations and 128 reactor models per generation, which is more than three times the compute time compared to the other optimization simulations reported in this Chapter (see Table 6.22).

Table 6.20 summarizes simulation p-3b's optimization problem parameters.

Table 6.20: Simulation p-3b optimization problem parameters.

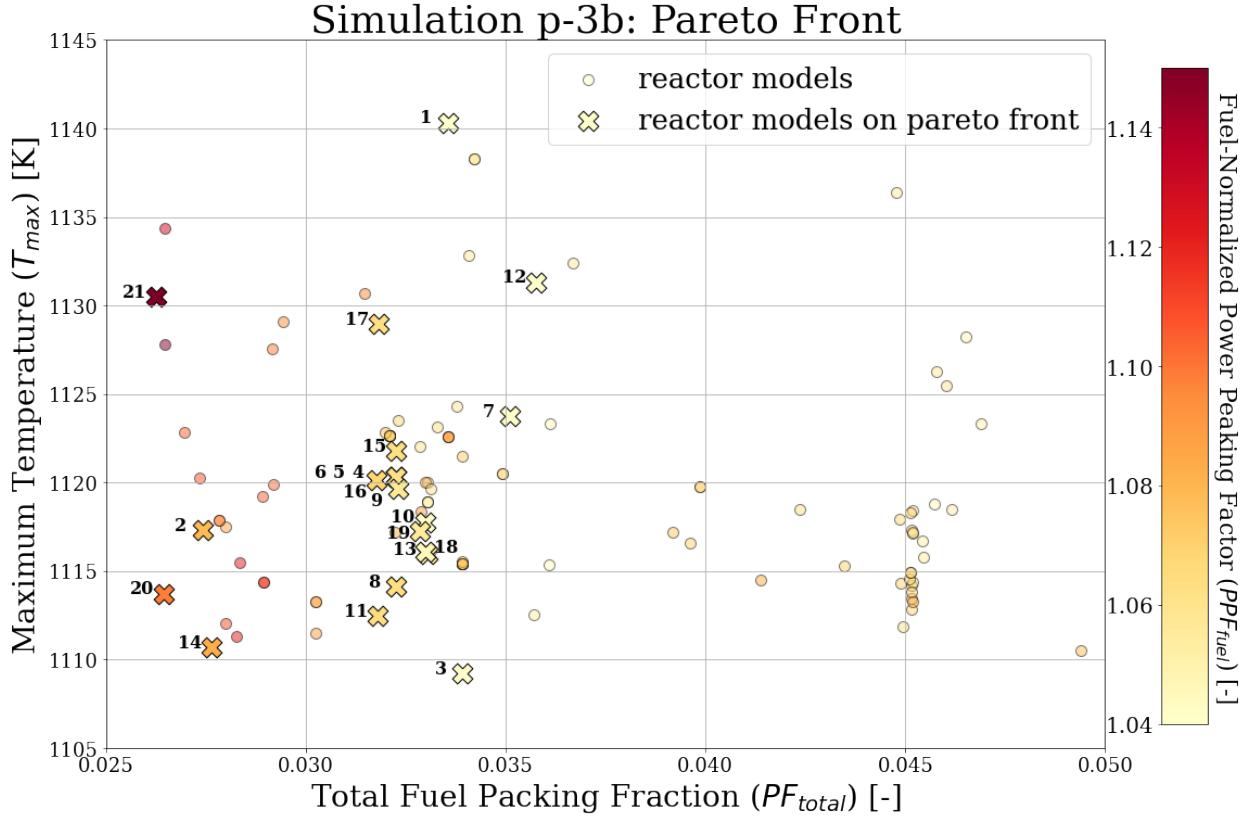
Three Objectives: Simulation p-3b	
Objectives	Minimize PF_{total} Minimize T_{max} Minimize PPF_{fuel}
Input parameter variations	$0.02 \leq PF_{total} \leq 0.04$ $\rho_{TRISO}(\vec{r})$: $0 < a < 2$ $\rho_{TRISO}(\vec{r})$: $0 < b < \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r})$: $0 < c < 2\pi$ Coolant channel shape: $0.05 < r_{top} < 0.35$ Coolant channel shape: $0.05 < r_{bot} < 0.35$
Constraints	$k_{eff} \geq 1.35$
Genetic algorithm parameters	Population size: 128 Generations: 8

Table 6.21 shows each generation's hypervolume value, confirming that simulation p-3b is sufficiently converged by generation 8.

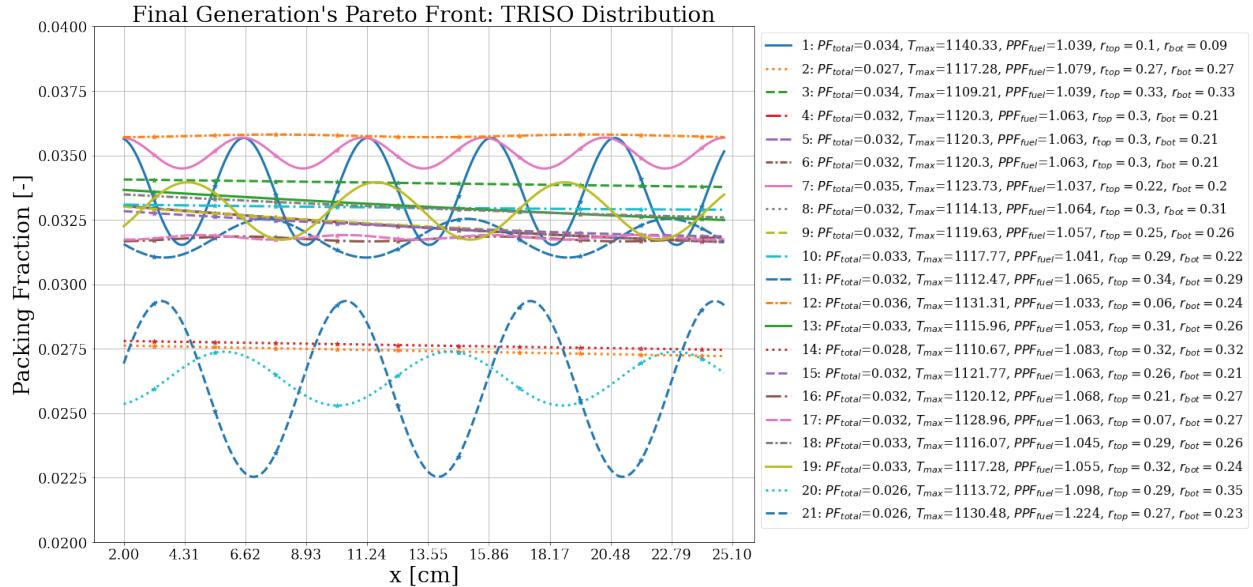
Table 6.21: Simulation p-3b hypervolume values at each generation.

Three Objectives: Simulation p-3b	
Reference point: (0.06, 1260 ,1.5)	
Generation	Hypervolume [-]
1	1.985
2	2.119
3	2.158
4	2.251
5	2.262
6	2.280
7	2.310
8	2.313

Figure 6.23a shows a plot of the final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} ; crosses indicate the reactor models that fall on the Pareto front. Figure 6.23b shows the 21 reactor models that fall on the Pareto front.



(a) Plot of final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} as a color dimension. Crosses indicate the reactor models on the Pareto front. Cross numbering correspond to TRISO distributions in Figure 6.23b.



(b) TRISO distribution for the 21 reactor models on the Pareto front.

Figure 6.23: Simulation p-3b – ROLLO triple-objective optimization to minimize the total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_{top}, r_{bot}).

Figure 6.23a demonstrates that ROLLO found 21 widely spread solutions in the final generation's Pareto front. Figure 6.24a shows the three distributions from Figure 6.23b that most-minimized each objective, respectively. Figures 6.24b, 6.24c, and 6.24d show the AHTR plank's TRISO distribution and coolant channel shape for most-minimized cases.

In Figure 6.24a, the AHTR plank model with the most-minimized PF_{total} (the blue dashed distribution) has an oscillating TRISO distribution with approximately $\sim 0.01\text{cm}$ of TRISO spatial variation, $PF_{total} = 0.026$, $r_{top} = 0.273\text{cm}$, and $r_{bot} = 0.232\text{cm}$ (the corresponding geometry is illustrated in Figure 6.24b). In Figure 6.24a, the AHTR plank model with the most-minimized T_{max} (the green dashed distribution) has a constant TRISO distribution of $PF_{total} = 0.034$, $r_{top} = 0.327\text{cm}$, and $r_{bot} = 0.33\text{cm}$ (the corresponding geometry is illustrated in Figure 6.24c). In Figure 6.24a, the AHTR plank model with the most-minimized PPF_{fuel} (the orange dashdot distribution) has a constant TRISO distribution of $PF_{total} = 0.036$, $r_{top} = 0.062\text{cm}$, and $r_{bot} = 0.245\text{cm}$ (the corresponding geometry is illustrated in Figure 6.24d). Section 6.4.3 discusses and explains simulation p-3b's results.

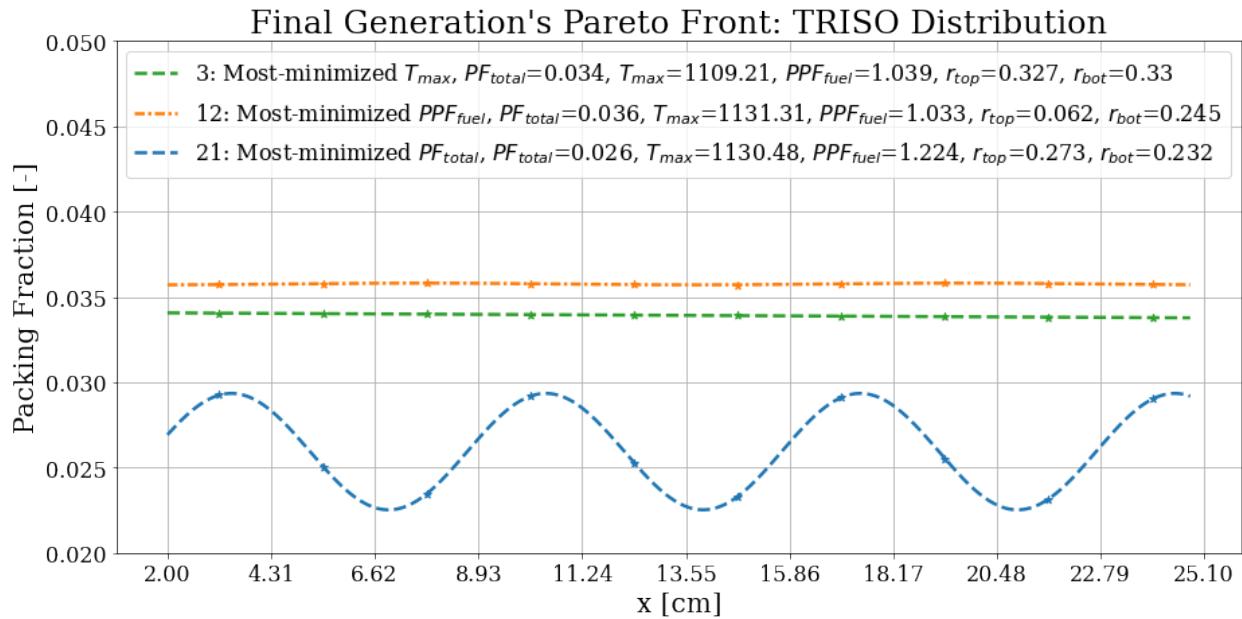
6.4.3 Triple-Objective Optimization Discussion

ROLLO successfully found a wide spread of reactor models in each of the triple-objective optimization simulation's final generation Pareto fronts. In this section, I explain how the driving factors and phenomena observed in the previous single-objective discussion (Section 6.2.4) combine to result in the optimal reactor models found by the triple-objective optimization simulations.

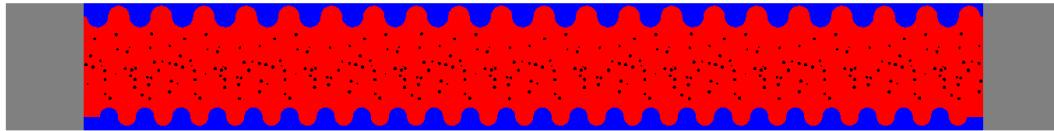
Simulation p-3a

In Section 6.4.1's simulation p-3a, I conducted a triple-objective optimization simulation to minimize total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) by varying PF_{total} and TRISO distribution. ROLLO found 14 widely spread reactor models on simulation p-3a's Pareto front (Figure 6.21).

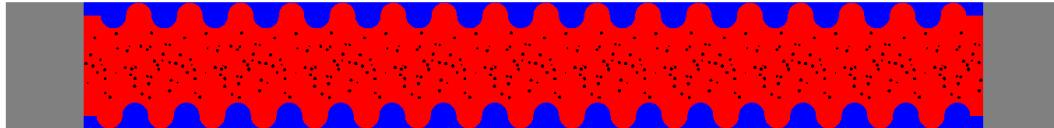
In simulation p-3a, ROLLO found that the plank model with the most-minimized T_{max} has a constant TRISO distribution with $PF_{total} = 0.059$ (reactor model 2 with the orange dotted distribution in Figure 6.22a). This distribution follows a similar flat shape as simulation p-1b's



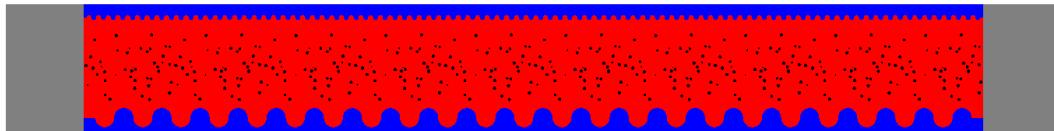
(a) TRISO distributions for the reactor models on the Pareto front that most minimized each objective.



(b) AHTR plank model with the most-minimized PF_{total} (corresponds to reactor model 21, the blue dashed distribution in Figure 6.24a.)



(c) AHTR plank model with the most-minimized T_{max} (corresponds to reactor model 3, the green straight dashed distribution in Figure 6.24a.)



(d) AHTR plank model with the most-minimized PPF_{fuel} (corresponds to reactor model 12, the orange dashdot distribution in Figure 6.24a.).

Figure 6.24: AHTR models and TRISO distribution for the 3 reactor models on simulation p-3b – ROLLO triple-objective optimization to minimize the total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the plank. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_{top}, r_{bot}).

most-minimized T_{max} TRISO distribution (Figure 6.8). In simulation p-3a, ROLLO found that the plank model with the most-minimized PF_{total} has an oscillating TRISO distribution with $PF_{total} = 0.02$ (reactor model 5 with purple dashed distribution in Figure 6.22a). This distribution is similar to simulation p-1a's most-minimized PF_{total} TRISO distribution (Figure 6.1b) with an oscillating TRISO distribution. In simulation p-3a, ROLLO found that the plank model with the most-minimized PPF_{fuel} has a slightly oscillating TRISO distribution with $PF_{total} = 0.025$ (reactor model with pink solid distribution in Figure 6.22a).

Most of the TRISO distributions on the Pareto front (Figure 6.21b) have a mostly flat distribution with approximately ~ 0.01 of TRISO spatial variation. The flatness is influenced by the minimize T_{max} objective. The variations in TRISO distributions are influenced by both the minimize PF_{total} and minimize PPF_{fuel} objectives. However, as mentioned previously, the PF_{total} and PPF_{fuel} relationship results in unexpected TRISO distributions at different PF_{total} values. The minimize PF_{total} objective tries to maximize fission reaction rate to enable a higher k_{eff} for a lower PF_{total} , and the PPF_{fuel} objective tries to flatten thermal flux.

Simulation p-3b

In Section 6.4.2's simulation p-3b, I conducted a triple-objective optimization simulation to minimize the total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) by varying PF_{total} , TRISO distribution, and coolant channel shape. ROLLO found 21 widely spread reactor models on simulation p-3b's Pareto front (Figure 6.23).

In simulation p-3b, ROLLO found that the plank model with the most-minimized T_{max} has a constant TRISO distribution, $PF_{total} = 0.034$, $r_{top} = 0.327\text{cm}$, and $r_{bot} = 0.33\text{cm}$ (reactor model 3 with the green dashed distribution in Figure 6.24a). This distribution follows a similar flat shape as simulation p-1b's most-minimized T_{max} TRISO distribution (Figure 6.8) and has large r_{top} and r_{bot} values due to the negative correlation between T_{max} and, r_{top} and r_{bot} determined in simulation p-1e (Sections 6.2.2 and 6.2.4).

In simulation p-3b, ROLLO found that the plank model with the most-minimized PF_{total} has an oscillating TRISO distribution with ~ 0.01 variation, $PF_{total} = 0.026$, $r_{top} = 0.273\text{cm}$, and $r_{bot} =$

0.232cm (reactor model 21 with the blue dashed distribution in Figure 6.24a). This distribution is similar to simulation p-1a's most-minimized PF_{total} TRISO distribution (Figure 6.1b) with an oscillating TRISO distribution.

In simulation p-3b, ROLLO found that the plank model with the most-minimized PPF_{fuel} has a constant TRISO distribution, $PF_{total} = 0.036$, $r_{top} = 0.062\text{cm}$, and $r_{bot} = 0.245\text{cm}$ (reactor model 12 with the orange dashdot distribution in Figure 6.24a). The constant TRISO distribution is unexpected for the most-minimized PPF_{fuel} result. However, as mentioned before the PF_{total} and PPF_{fuel} relationship results in unexpected TRISO distributions at different PF_{total} values. Simulation p-1d and p-1f concluded that there was no correlation between PF_{total} and PPF_{fuel} with the coolant channel shape, thus, the large r_{bot} and r_{top} values are only influenced by the minimize T_{max} objective.

Similar to simulation p-3a, most of the TRISO distributions on simulation p-3b's Pareto front (Figure 6.23b) have a mostly flat distribution with approximately $\sim 0.01\text{cm}$ of variation. Most of the Pareto front's r_{tot} values are large due to the minimize T_{max} objective. The TRISO distribution's flatness is influenced by the minimize T_{max} objective. The variations in TRISO distributions are influenced by both the minimize total packing fraction and minimize PPF_{fuel} objectives. However, as mentioned previously, the PF_{total} and PPF_{fuel} relationship results in unexpected TRISO distributions at different PF_{total} values. The minimize PF_{total} objective tries to maximize fission reaction rate to enable a higher k_{eff} for lower PF_{total} , and the PPF_{fuel} objective tries to flatten thermal flux.

6.4.4 Triple-Objective Optimization Major Takeaways

The triple-objective optimization simulations successfully found a wide spread of reactor models on their Pareto fronts that meet each objective to varying degrees. Simulation p-3b's multi-objective optimization shows the result of minimizing all three objectives (minimize PF_{total} , T_{max} , and PPF_{fuel}) while varying all the input parameters (PF_{total} , TRISO distribution, and coolant channel shape). Figure 6.23 shows the 21 widely spread out reactor models on simulation p-3b's Pareto front.

6.5 AHTR Plank: Computational Cost Summary

Each ROLLO optimization simulation in this chapter runs the AHTR model *generations* \times *individuals* number of times with different geometries, resulting in high computationally cost. I report a summary of the computational cost to give readers a sense of how much computational power is required to run similar ROLLO reactor optimization simulations.

The optimization simulations for this work were run on the BlueWaters supercomputer [87] and the Theta supercomputer at the Argonne Leadership Computing Facility under the Director's Discretionary Allocation Program [107]. Each BlueWaters XE compute node has 32 Interlagos processor cores with a nominal clock speed of 3.2GHz [87], and each Theta compute node is a single Xeon Phi chip with 64 cores with a nominal clock speed of 1.5GHz [107]. Each optimization simulation takes a different amount of node-hours to run due to differences in simulation software, tallies, and intermediate steps required. Table 6.22 reports the computational cost for each optimization simulation. One may refer to Table 6.2 for each simulation's parameters.

Table 6.22: Computational cost of ROLLO simulations for optimizing AHTR plank. BW: BlueWaters Supercomputer, Theta: Theta Supercomputer.

Num of Objs	Sim	Machine	Compute Cost Per Gen [node-hours]	Generations [#]	Total Compute Cost [node-hours]
1	p-1a	BW	23.9	10	238.7
	p-1b	BW	96.0	10	959.7
	p-1c	BW	65.9	10	658.6
	p-1d	Theta	49.6	3	148.7
	p-1e	Theta	71.4	4	285.6
	p-1f	Theta	46.4	5	231.8
2	p-2a	Theta	132.2	2	264.5
	p-2b	Theta	66.3	3	198.9
	p-2c	Theta	167.1	3	501.2
3	p-3a	Theta	92.9	5	464.5
	p-3b	Theta	178.2	8	1425.8

6.6 Summary

This chapter described the Advanced High-Temperature Reactor (AHTR) plank's Reactor evOLutionary aLgorithm Optimizer (ROLLO) optimization results. I varied the following AHTR plank input parameters: Tristructural Isotropic (TRISO) packing fraction distribution ($\rho_{TRISO}(\vec{r})$), total fuel packing fraction (PF_{total}), and coolant channel shape; in an effort to minimize the following objectives: total fuel packing fraction (PF_{total}), maximum plank temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}).

In Section 6.2's six single-objective optimization simulations (p-1a, p-1b, p-1c, p-1d, p-1e, p-1f) and Section 6.2.4 discussions, I characterized each objective's driving factors and relationship with each input parameter. I determined that the minimize T_{max} objective is driven by minimizing plank's maximum temperature and, flattens TRISO distribution and maximizes coolant channel shape's r_{tot} to achieve the objective. The minimize PF_{total} objective is driven by maximizing the plank's total fission reaction rate and influences the TRISO distribution to achieve the objective. The minimize PPF_{fuel} objective is driven by flattening the plank's thermal flux distribution and influences PF_{total} and TRISO distribution to achieve the objective. Both the minimize PF_{total} and minimize PPF_{fuel} objectives have no correlation with the coolant channel shape.

In Sections 6.3 and 6.4's five multi-objective optimization simulations (p-2a, p-2b, p-2c, p-3a, and p-3b) and Sections 6.3.4 and 6.4.3 discussion, I further analyzed how the objectives' combined effects resulted in the optimal reactor models found by each multi-objective optimization simulation. All the multi-objective optimization simulations successfully found a wide spread of reactor models on a Pareto front that meets each objective to varying degrees. In the multi-objective optimization simulations, the minimize T_{max} objective continued to influence the flattening of the TRISO distribution and maximizing of the coolant channel shape (r_{bot} and r_{top}) to achieve the objective. The results from simulation p-2b suggested that the minimize PF_{total} objective's driving factor maximize total fission reaction rate and minimize PPF_{fuel} objective's driving factor flattening thermal flux distribution influence each other resulting in unexpected TRISO distributions at different PF_{total} values. Simulation p-3b's multi-objective optimization shows the result of minimizing all three objectives (minimize PF_{total} , T_{max} , and PPF_{fuel}) while varying all the input parameters (PF_{total} , TRISO distribution, and coolant channel shape). Figure 6.23 shows the 21

widely spread reactor models on simulation p-3b's Pareto front that meet the three objectives to varying degrees.

This chapter demonstrated ROLLO's success in conducting multi-objective optimization, a global search of the large reactor design space, to find optimal reactor models on the Pareto front that satisfy all the objectives. Reactor designers can utilize ROLLO for multi-objective optimization problems with any number of objectives and arbitrary input parameters, to narrow down the search space to find reactor models that meet their desired requirements. The results from ROLLO's multi-objective optimizations help reactor designers gain a clearer understanding of the reactor models' parameters that meet their defined objectives to varying degrees. From there, reactor designers can determine the importance of each objective for their purposes, then conduct sensitivity analysis and use higher fidelity models to further study a smaller section of the optimal design space that they are interested in.

This chapter successfully determined and explained the driving factors behind each reactor optimization objective and their relationship with one another. Characterizations of each objective for a simple AHTR plank model provides insights for the next chapter in which I conduct multi-objective optimization for the more complex AHTR one-third assembly model.

Chapter 7

AHTR One-Third Assembly Optimization Results

This chapter reports the Advanced High-Temperature Reactor (AHTR) one-third assembly's Reactor evOLutionary aLgorithm Optimizer (ROLLO) optimization results. As with the AHTR plank models presented in Chapter 6, I vary the following input parameters for the AHTR one-third assembly:

- Tristructural Isotropic (TRISO) packing fraction distribution ($\rho_{TRISO}(\vec{r})$)
- Total fuel packing fraction (PF_{total})
- Coolant channel shape (r_1, r_2, r_3, r_4 , and r_5).

Section 5.3.1 detailed how I vary these input parameters in my models. I optimize the AHTR one-third assembly for the following objectives:

- Minimize total fuel packing fraction (PF_{total})
- Minimize maximum one-third assembly temperature (T_{max})
- Minimize fuel-normalized power peaking factor (PPF_{fuel})

Table 7.1 outlines these objectives and their motivation. Chapter 5 detailed the methodology for AHTR one-third assembly modeling and ROLLO optimization.

Table 7.1: Reactor evOLutionary aLgorithm Optimizer (ROLLO) Advanced High-Temperature Reactor (AHTR) optimization problem objectives with their quantification descriptions and motivation.

Objective	Quantification	Motivation
Minimize fuel amount	Minimize total fuel packing fraction	Cost savings, Non-proliferation
Maximize heat transfer	Minimize maximum temperature	Minimize thermal stress in the fuel
Minimize power peaking	Minimize power peaking factor normalized by fuel distribution	Efficient fuel utilization, longer core life, safety

The subsequent sections outline the AHTR one-third assembly optimization simulations (Section 7.1), describe the single-objective (Section 7.2), double-objective (Section 7.3), and triple-objective (Section 7.4) ROLLO optimization simulation results, and report each simulation’s computational cost (Section 7.5). Appendix A lists all the data and analysis related to this chapter to enable the reproduction of all the simulations.

7.1 ROLLO AHTR One-Third Assembly Optimization Simulations Overview

In this chapter, I first conduct single-objective, single input parameter ROLLO optimizations to understand the individual impacts of each objective on each input parameter. I then run multi-objective optimizations for combinations of the input parameters and objectives, whose setup is informed by the single-objective optimization results. Table 7.2 summarizes the details of each ROLLO optimization simulation conducted in this chapter.

Simulations are run on the Theta supercomputer at the Argonne Leadership Computing Facility under the Director’s Discretionary Allocation Program [107]. Section 7.5 details each optimization simulation’s computational cost. Readers must consider the computational cost if they desire to reproduce this analysis.

Table 7.2: Reactor evOLutionary aLgorithm Optimizer (ROLLO) simulations for optimizing Advanced High-Temperature Reactor (AHTR) one-third assembly. Relevant variables include: PF_{total} : total fuel packing fraction, T_{max} : maximum one-third assembly temperature, PPF_{fuel} : normalized power peaking factor, $\rho_{TRISO}(\vec{r})$: TRISO packing fraction distribution

Objs [#]	Sim	Objectives	Constraints	Varying Parameters	Simulation Software
1	a-1a	• $\min(PF_{total})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC
	a-1b	• $\min(T_{max})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$	OpenMC, Moltres
	a-1c	• $\min(PPF_{fuel})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$	OpenMC
	a-1d	• $\min(PF_{total})$	• $k_{eff} \geq 1.0$	• Coolant channel shape • PF_{total}	OpenMC
	a-1e	• $\min(T_{max})$	• $k_{eff} \geq 1.38$	• Coolant channel shape	OpenMC, Moltres
	a-1f	• $\min(PPF_{fuel})$	• $k_{eff} \geq 1.0$	• Coolant channel shape	OpenMC
2	a-2a	• $\min(PF_{total})$ • $\min(T_{max})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC, Moltres
	a-2b	• $\min(PF_{total})$ • $\min(PPF_{fuel})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC
	a-2c	• $\min(T_{max})$ • $\min(PPF_{fuel})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$	OpenMC, Moltres
3	a-3a	• $\min(PF_{total})$ • $\min(PPF_{fuel})$ • $\min(T_{max})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$ • PF_{total}	OpenMC, Moltres
	a-3b	• $\min(PF_{total})$ • $\min(PPF_{fuel})$ • $\min(T_{max})$	• $k_{eff} \geq 1.38$	• $\rho_{TRISO}(\vec{r})$ • PF_{total} • Coolant channel shape	OpenMC, Moltres

7.2 AHTR One-Third Assembly: Single-Objective Optimization Results

This section reports the AHTR one-third assembly's ROLLO single-objective optimization results. Table 7.2 summarizes the parameters for the single-objective simulations that are presented in this section: a-1a, a-1b, a-1c, a-1d, a-1e, and a-1f. In the following subsections, I describe the single-objective simulation results grouped by the minimized objective (Sections 7.2.1, 7.2.2, and 7.2.3), and provide discussion about the single-objective simulations results (Section 7.2.4).

Section 4.5.1 described that for single-objective optimization, the reactor designer should plot the objective's minimum and average values at each generation and when the values are no longer changing, the simulation is converged. If a single-objective optimization problem's objective converges earlier than the five generations I intended to run (determined in Section 5.5.2), I stop the simulation at that generation to save computational resources as the solution will not change. However, if the problem's objective does not converge by generation 5, I run the problem for a few more generations until satisfactory convergence is reached.

7.2.1 Objective: Minimize Total Packing Fraction (PF_{total})

This section describes the single-objective a-1a and a-1d optimization simulation results. Both simulations minimize the total fuel packing fraction (PF_{total}) objective. The minimize PF_{total} objective is important because a reactor that uses less fuel with similar performance enables cost savings. Simulation a-1a varies the PF_{total} and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), and simulation a-1d varies the PF_{total} and coolant channel shape (r_1, r_2, r_3, r_4 , and r_5) to achieve this objective.

Simulation a-1a: Variation of PF_{total} and $\rho_{TRISO}(\vec{r})$

Table 7.3 summarizes the optimization problem parameters for simulation a-1a. The one-third assembly's TRISO distribution is varied based on sine distributions. These sine distributions vary in x across each plank and in y between planks, as described in Section 5.3.1. For simulation a-1a, I vary PF_{total} between 0.05 and 0.07. If the simulation used the FHR benchmark equivalent

Table 7.3: Simulation a-1a Optimization Problem Parameters

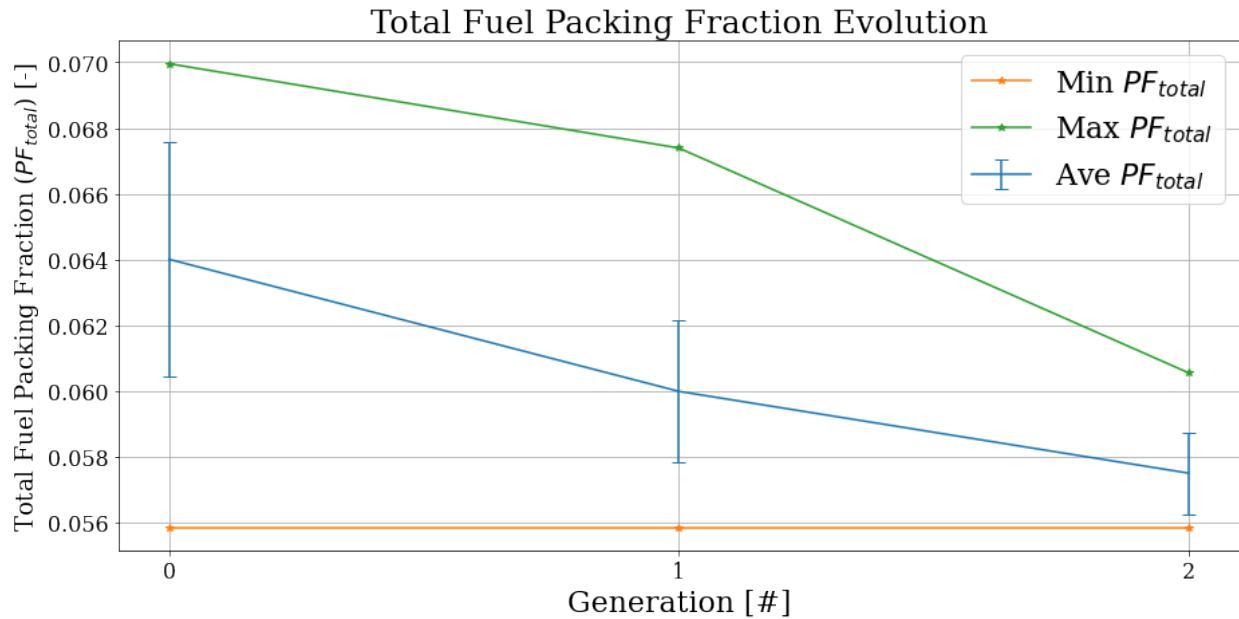
Single Objective: Simulation a-1a	
Objectives	Minimize PF_{total}
Input Parameter Variations	$0.05 \leq PF_{total} \leq 0.07$ $\rho_{TRISO}(\vec{r}): 0 \leq a \leq 2, 0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r}): 0 \leq b \leq \frac{\pi}{2}, 0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 \leq c \leq 2\pi, 0 \leq f \leq 2\pi$
Constraints	$k_{eff} \geq 1.38$
Genetic Algorithm Parameters	Population size: 128 Generations: 3

$PF_{total} = 0.153$, at certain sine distributions, some fuel cells would have $PF > 0.3$. OpenMC’s random sequential packing algorithm becomes prohibitively slow at $PF > 0.3$, resulting in long runtimes. For simulations that do not vary PF_{total} (a-1b, a-1c, a-1e, a-1f, and a-2c), I set $PF_{total} = 0.06$, and for simulations that vary PF_{total} (a-1a, a-1d, a-2a, a-2b, a-3a, and a-3b), I set the boundaries of PF_{total} between 0.05 and 0.07. I use $PF_{total} = 0.06$ because it is approximately the largest PF_{total} that enables $k_{eff} \geq 1.38$ and will avoid fuel cells with $PF > 0.3$ occurrences.

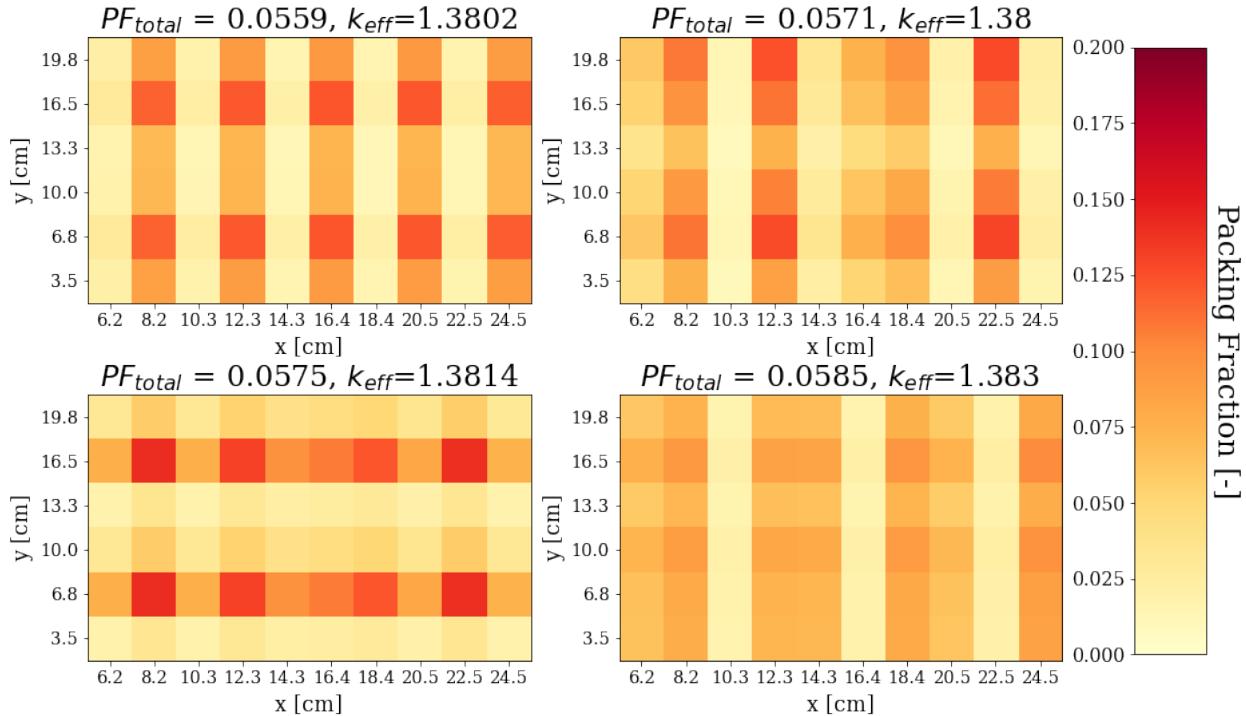
Figure 7.1a shows simulation a-1a’s PF_{total} evolution. Figure 7.1b shows four unique TRISO packing fraction distributions in the final generation with the most-minimized PF_{total} . Figure 7.1c illustrates the AHTR one-third assembly model with the most-minimized PF_{total} .

Figure 7.1a shows that the minimum and average PF_{total} converged to approximately 0.057 in the final generation. In Figure 7.1b, the four unique TRISO packing fraction distributions in the final generation that most-minimized PF_{total} have various oscillating TRISO distribution patterns.

The one-third assembly model with the most-minimized PF_{total} has a $PF_{total} = 0.0559$, an oscillating TRISO distribution along the x-axis and y-axis, and a packing fraction standard deviation of 0.04 across the one-third assembly. Along the x-axis, the distribution peaks on the even fuel cell columns (at 8.2cm, 12.3cm, 16.4cm, 20.5cm, and 24.5cm). The even columns have the largest y-axis variation of ~0.05 with peaks of $PF \approx 0.12$. Along the y-axis, the distribution peaks on the 2nd and 5th fuel cell rows (at 6.8cm and 16.5cm). The 2nd and 5th row have the largest x-axis variation of ~0.10 with peaks of $PF \approx 0.12$. Because it is more productive to compare all of the single objective results to one another, Section 7.2.4 discusses the driving factors for the minimize PF_{total} objective and explains simulation a-1a’s most-minimized PF_{total} oscillating TRISO distribution.

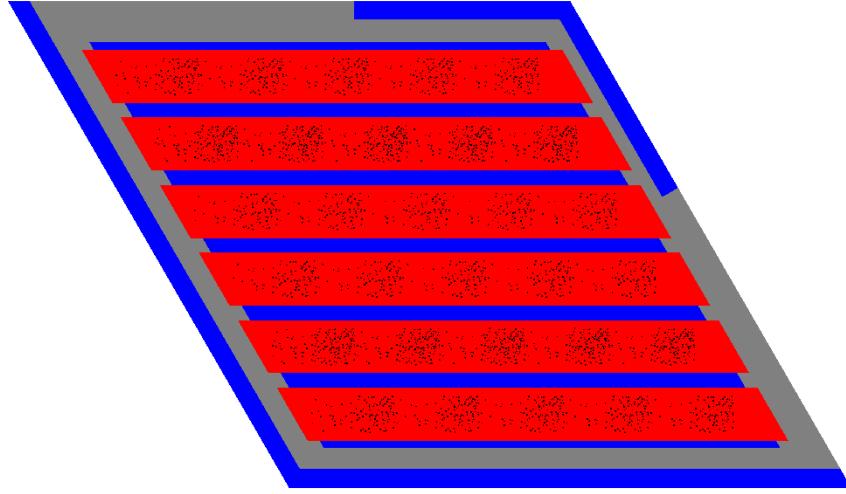


(a) Minimum, average, and maximum PF_{total} evolution.



(b) TRISO packing fraction distribution for four unique reactor models with the smallest PF_{total} in the final generation.

Figure 7.1: Simulation a-1a – ROLLO single-objective optimization to minimize total fuel packing fraction (PF_{total}) in AHTR one-third assembly. Input parameters varied: PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



(c) AHTR one-third assembly model with the most-minimized PF_{total} , corresponding to the top left TRISO distribution in Figure 7.1b. The reactor model has $PF_{total} = 0.0559$ and $k_{eff} = 1.3802$.

Figure 7.1: (contd.) Simulation a-1a – ROLLO single-objective optimization to minimize total fuel packing fraction (PF_{total}) in AHTR one-third assembly. Input parameters varied: total fuel packing fraction (PF_{total}), TRISO packing fraction distribution ($\rho_{TRISO}(\bar{r})$).

Simulation a-1d: Variation of PF_{total} and Coolant channel shape

Table 7.4 summarizes simulation a-1d's optimization problem parameters.

Table 7.4: Simulation a-1d Optimization Problem Parameters

Single Objective: Simulation a-1d	
Objectives	Minimize PF_{total}
Input Parameter variations	$0.01 < PF_{total} < 0.04$ coolant channel shape: $0.05 < r_1 < 0.35$ coolant channel shape: $0.05 < r_2 < 0.35$ coolant channel shape: $0.05 < r_3 < 0.35$ coolant channel shape: $0.05 < r_4 < 0.35$ coolant channel shape: $0.05 < r_5 < 0.35$
Constraints	$k_{eff} \geq 1.0$
Genetic Algorithm Parameters	Population size: 64 Generations: 2

To remind readers each radius value's orientation, Figure 7.2 shows an annotated AHTR one-third assembly's inter-plank coolant channel shapes for $r_1, r_2, r_3, r_4, r_5 = 0.3, 0.2, 0.1, 0.2, 0.3$.

Figure 7.3 shows the plots of the simulation a-1d final generation coolant channel shape's

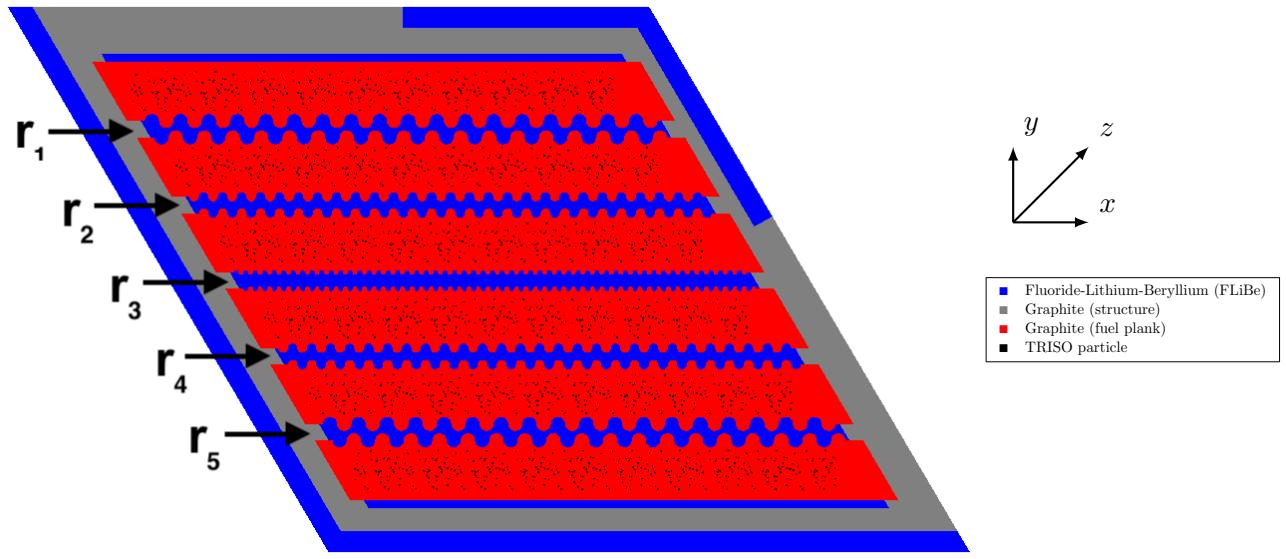


Figure 7.2: Advanced High-Temperature Reactor (AHTR) one-third assembly example with coolant channel shape variation, $r_1, r_2, r_3, r_4, r_5 = 0.3\text{cm}, 0.2\text{cm}, 0.1\text{cm}, 0.2\text{cm}, 0.3\text{cm}$.

r_1, r_2, r_3, r_4 , and r_5 values against PF_{total} . The random scattering of reactor model points in Figure 7.3 demonstrates that there is no correlation between PF_{total} and coolant channel shape's r_1, r_2, r_3, r_4 , and r_5 .

7.2.2 Objective: Minimize Maximum Temperature (T_{max})

This section describes the single-objective a-1b and a-1e optimization simulation results. Both simulations minimize the maximum one-third assembly temperature (T_{max}) objective. The minimize T_{max} objective is important because a reactor that has a lower peak temperature minimizes thermal stresses in the fuel. Simulation a-1b varies TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), and simulation a-1e varies the coolant channel shape (r_1, r_2, r_3, r_4 , and r_5).

Simulation a-1b: Variation of $\rho_{TRISO}(\vec{r})$

Table 7.5 summarizes simulation a-1b's optimization problem parameters. Figure 7.4a shows the one-third assembly's T_{max} evolution. Figure 7.4b shows four unique TRISO packing fraction distributions in the final generation with the most minimized T_{max} . Figure 7.4c illustrates the AHTR

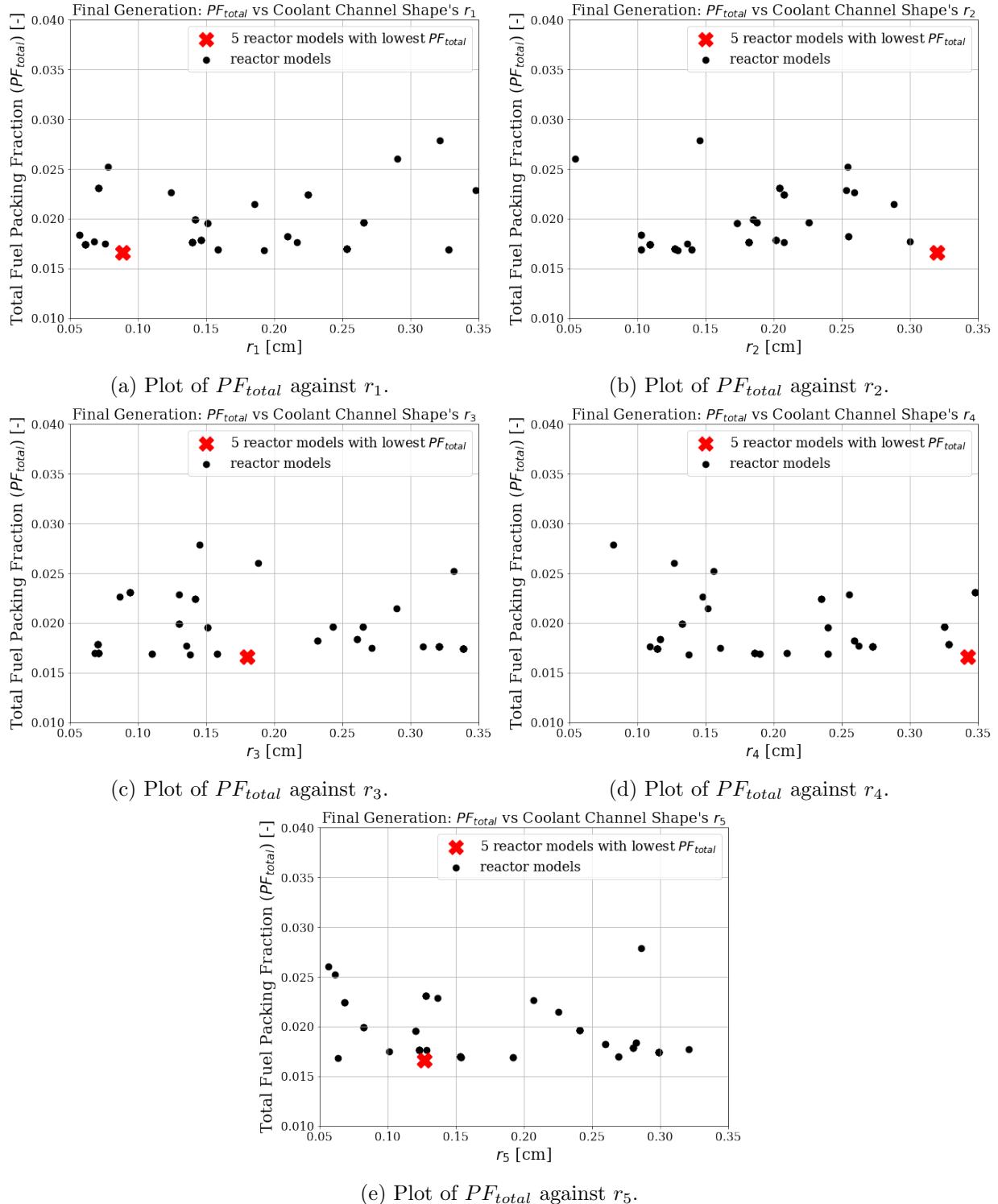


Figure 7.3: Simulation a-1d – ROLLO single-objective optimization to minimize total fuel packing fraction (PF_{total}). Plots of simulation a-1d final generation’s reactor models PF_{total} against coolant channel shape input parameters. Red crosses indicate the five reactor models with the lowest PF_{total} . There is only red cross shown each plot because the five reactor models with the lowest PF_{total} are the same and overlap. Input parameters varied: PF_{total} and coolant channel shape (r_1, r_2, r_3, r_4, r_5).

Table 7.5: Simulation a-1b Optimization Problem Parameters

Single Objective: Simulation a-1b	
Objectives	Minimize T_{max}
Input Parameter variations	$\rho_{TRISO}(\vec{r})$: $0 \leq a \leq 2$, $0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r})$: $0 \leq b \leq \frac{\pi}{2}$, $0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r})$: $0 \leq c \leq 2\pi$, $0 \leq f \leq 2\pi$
Constraints	$k_{eff} \geq 1.38$ $PF_{total} = 0.06$
Genetic Algorithm Parameters	Population size: 128 Generations: 3

one-third assembly model with the most-minimized T_{max} .

Some key features that can be observed in Figure 7.4a are that the minimum and average one-third assembly's T_{max} converged to approximately 1200 K. In Figure 7.4b, the one-third assembly model with the most-minimized T_{max} has a $T_{max} = 1186.5$ K and an almost constant TRISO packing fraction distribution with packing fraction standard deviation of 0.0009 across the one-third assembly. Because it is more productive to compare all of the single objective results to one another, Section 7.2.4 discusses and explains simulation p-1b's most-minimized T_{max} almost constant TRISO distribution.

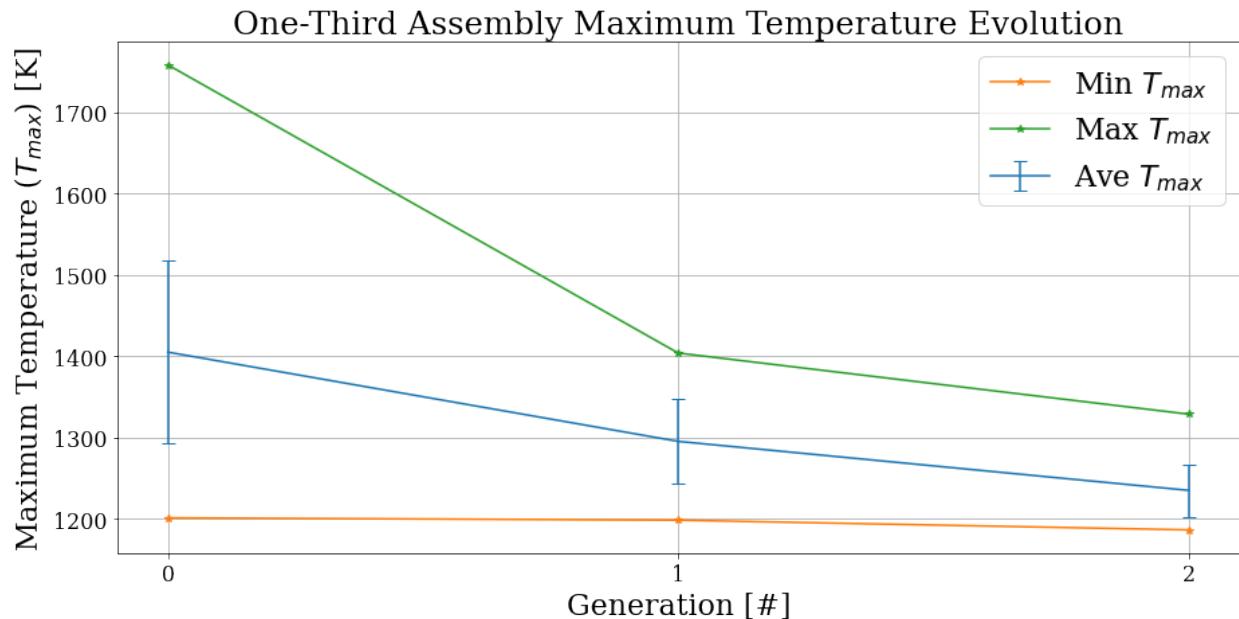
Simulation a-1e: Variation of Coolant channel shape

Table 7.6 summarizes simulation a-1e's optimization problem parameters.

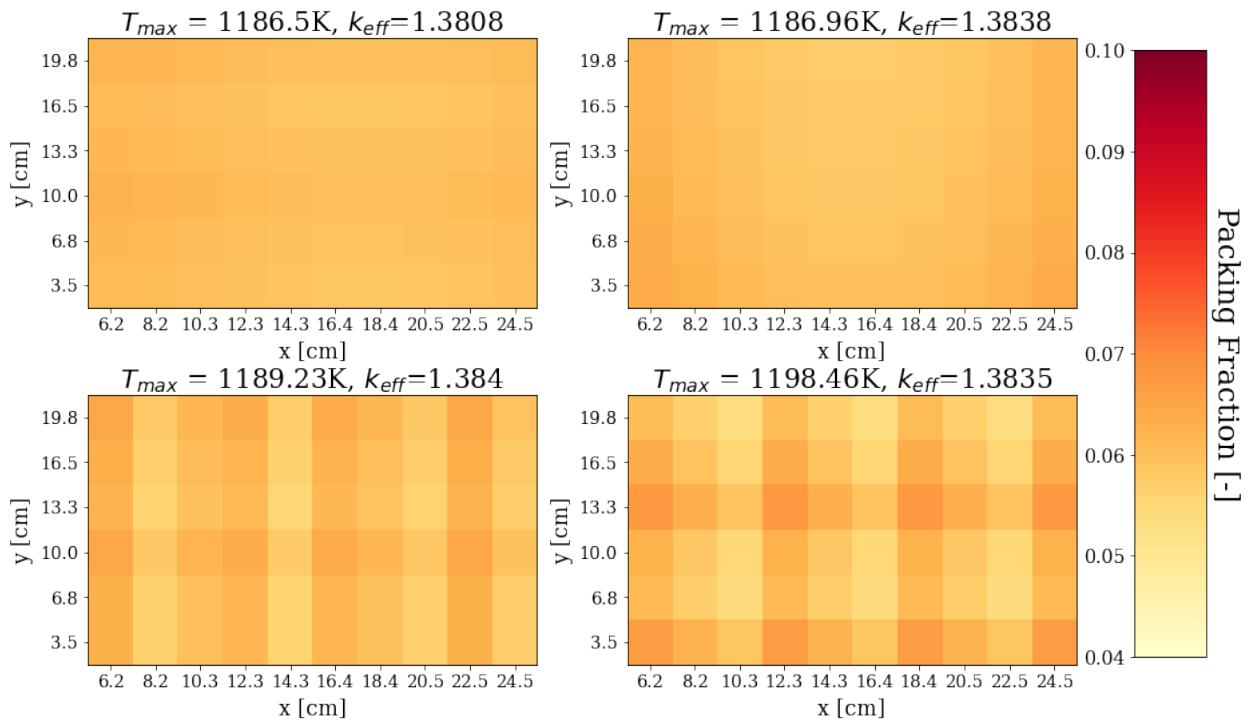
Table 7.6: Simulation a-1e Optimization Problem Parameters

Single Objective: Simulation a-1e	
Objectives	Minimize T_{max}
Input Parameter variations	coolant channel shape: $0.05 < r_1 < 0.35$ coolant channel shape: $0.05 < r_2 < 0.35$ coolant channel shape: $0.05 < r_3 < 0.35$ coolant channel shape: $0.05 < r_4 < 0.35$ coolant channel shape: $0.05 < r_5 < 0.35$
Constraints	$k_{eff} \geq 1.38$ $PF_{total} = 0.06$
Genetic Algorithm Parameters	Population size: 128 Generations: 2

To remind readers each r value's orientation, Figure 7.2 shows an annotated AHTR one-third assembly's inter-plank coolant channel shapes for $r_1, r_2, r_3, r_4, r_5 = 0.3, 0.2, 0.1, 0.2, 0.3$.

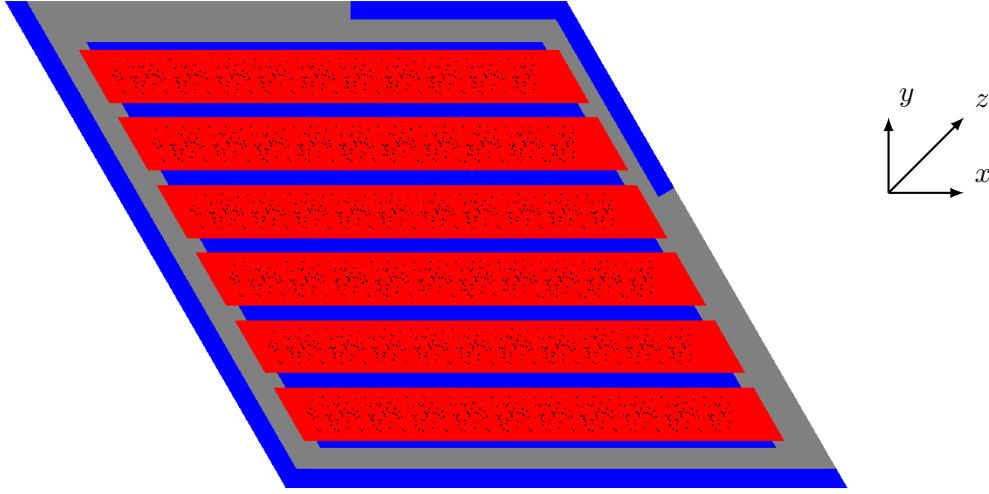


(a) Minimum, average, and maximum T_{max} evolution.



(b) TRISO packing fraction distribution for four unique reactor models with the smallest T_{max} in the final generation.

Figure 7.4: Simulation a-1b – ROLLO single-objective optimization to minimize maximum temperature (T_{max}) in the AHTR one-third assembly. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



(c) AHTR one-third assembly model with the most-minimized T_{max} , corresponding to the top left TRISO distribution in Figure 7.4b. The reactor model has $T_{max} = 1180.29\text{K}$ and $k_{eff} = 1.3046$.

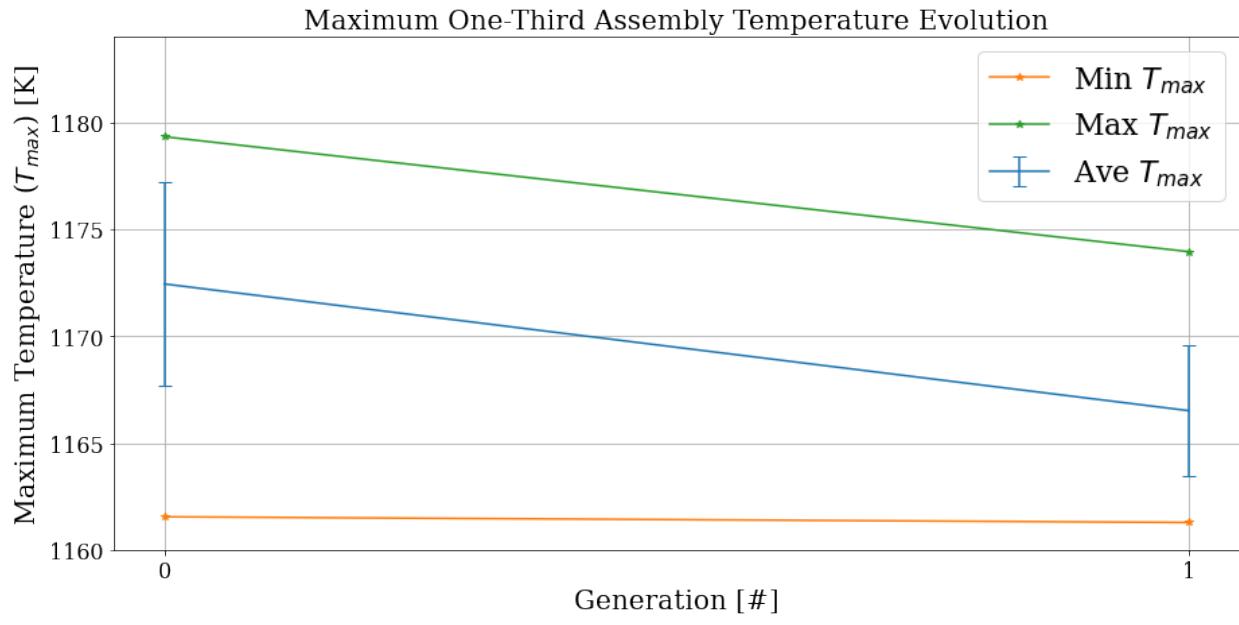
Figure 7.4: (contd.) Simulation a-1b – ROLLO single-objective optimization to minimize maximum temperature (T_{max}) in AHTR one-third assembly. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

Figure 7.5a shows the one-third assembly's T_{max} evolution. Figure 7.5b illustrates the AHTR one-third assembly model with the most-minimized T_{max} . Figures 7.5c, 7.5d, 7.5e, 7.5f, and 7.5g show the plots of coolant channel shape's r_1, r_2, r_3, r_4 , and r_5 values against T_{max} .

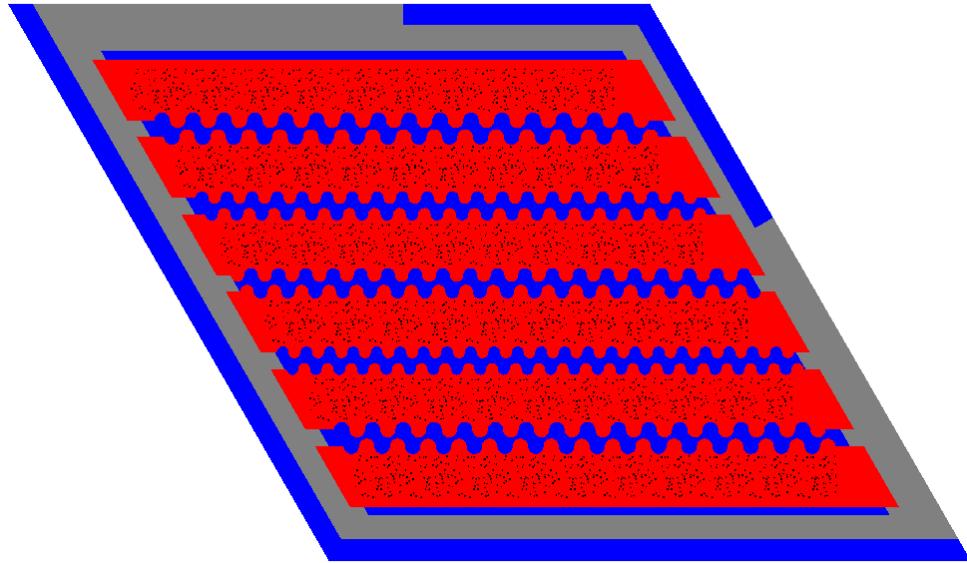
Figures 7.5c and 7.5g demonstrate negative linear correlations between the one-third assembly's T_{max} with r_1 and r_5 . The random scattering of reactor model points in Figures 7.5d, 7.5e and 7.5f demonstrate that there is no correlation between T_{max} with r_2, r_3 , and r_4 . Section 7.2.4 discusses and explains the relationship between T_{max} and coolant channel shape.

7.2.3 Objective: Minimize Fuel-Normalized Power Peaking Factor (PPF_{fuel})

This section describes the single-objective a-1c and a-1f optimization simulation results. Both simulations minimize fuel-normalized power peaking factor (PPF_{fuel}). The minimize PPF_{fuel} objective is important because a reactor that has a lower fuel-normalized power peaking factor will have more even and efficient fuel utilization. Simulation a-1c varies TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), and simulation a-1f varies the coolant channel shape (r_1, r_2, r_3, r_4, r_5).



(a) Minimum, average, and maximum evolution of AHTR one-third assembly's T_{max} .



(b) AHTR one-third assembly model with the most-minimized T_{max} . The reactor model has $T_{max} = 1161.28K$, $r_1 = 0.32cm$, $r_2 = 0.26cm$, $r_3 = 0.28cm$, $r_4 = 0.24cm$, and $r_5 = 0.32cm$.

Figure 7.5: Simulation a-1e – ROLLO single-objective optimization to minimize maximum one-third assembly temperature (T_{max}). Plots of final generation's reactor models T_{max} against coolant channel shape input parameters. Red crosses indicate the five reactor models with the lowest T_{max} . Input parameters varied: coolant channel shape (r_1, r_2, r_3, r_4, r_5).

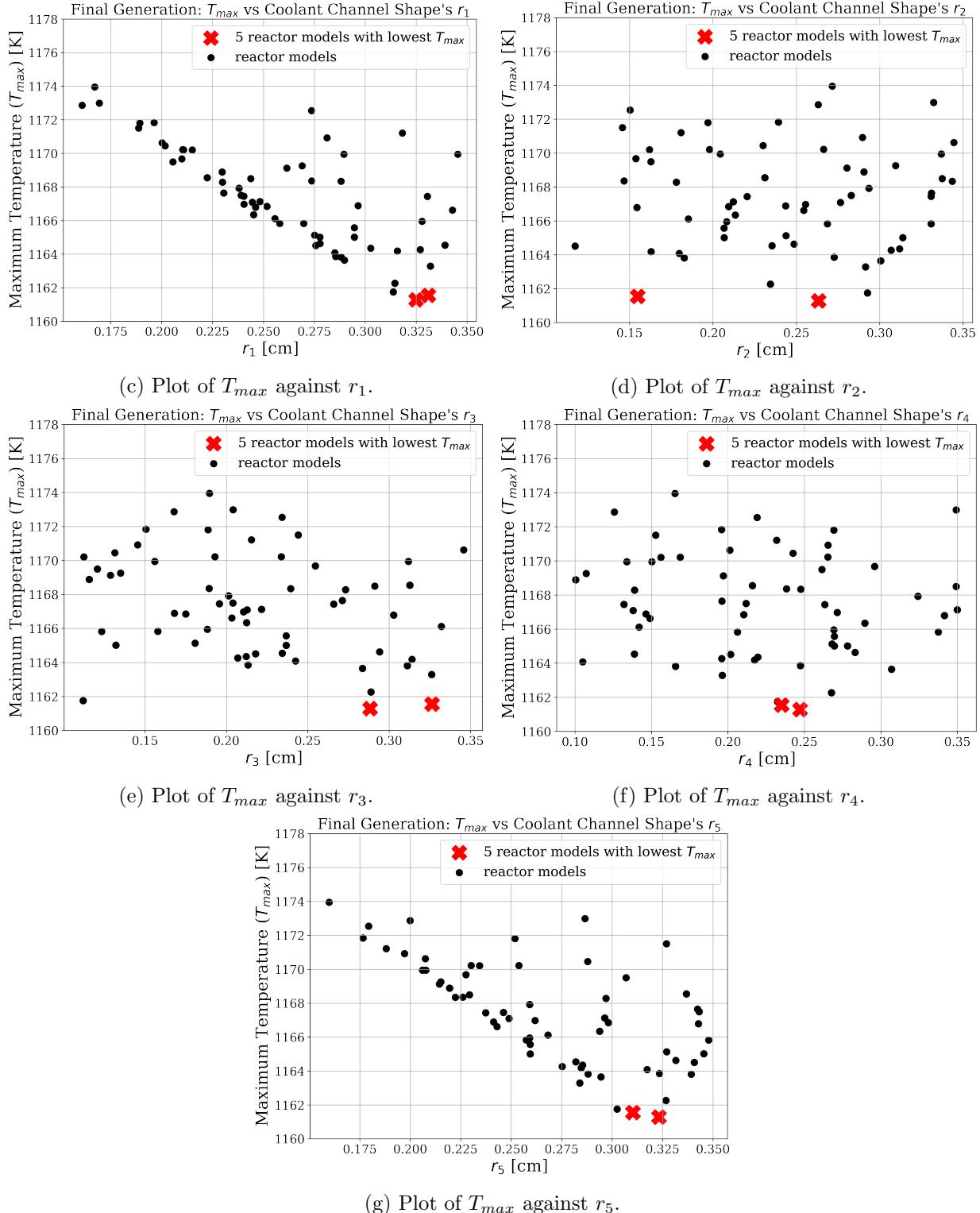


Figure 7.5: (contd.) Simulation a-1e – ROLLO single-objective optimization to minimize maximum one-third assembly temperature (T_{max}). Plots of final generation's reactor models T_{max} against coolant channel shape input parameters. Red crosses indicate the five reactor models with the lowest T_{max} . Some of the five reactor models with the lowest T_{max} are the same, thus their crosses overlap. Input parameters varied: coolant channel shape (r_1, r_2, r_3, r_4, r_5).

Simulation a-1c: Variation of $\rho_{TRISO}(\vec{r})$

Table 7.7 summarizes simulation a-1c's optimization problem parameters.

Table 7.7: Simulation a-1c Optimization Problem Parameters

Single Objective: Simulation a-1c	
Objectives	Minimize PPF_{fuel}
Input Parameter variations	$\rho_{TRISO}(\vec{r}): 0 \leq a \leq 2, 0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r}): 0 \leq b \leq \frac{\pi}{2}, 0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 \leq c \leq 2\pi, 0 \leq f \leq 2\pi$
Constraints	$k_{eff} \geq 1.38$ $PF_{total} = 0.06$
Genetic Algorithm Parameters	Population size: 128 Generations: 2

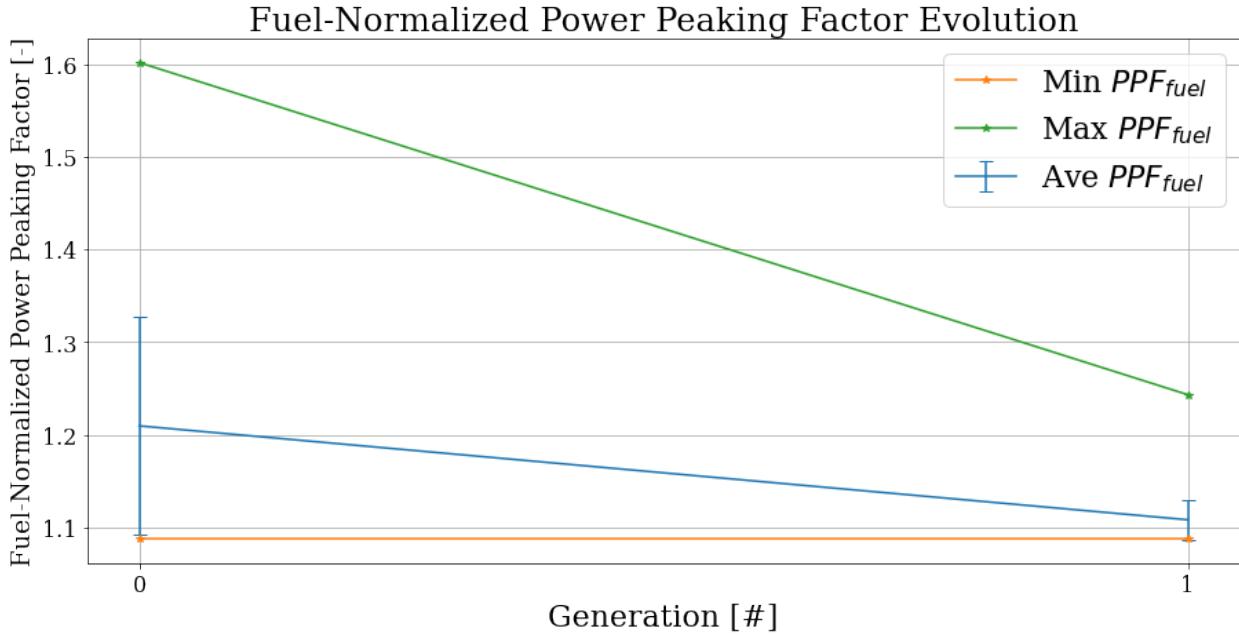
Figure 7.6a shows the one-third assembly's PPF_{fuel} evolution. Figure 7.6b shows the four unique TRISO packing fraction distributions in the final generation with the most minimized PPF_{fuel} . Figure 7.6c illustrates the AHTR one-third assembly model with the most-minimized PPF_{fuel} .

Features of note in Figure 7.6a are that the minimum and average one-third assembly's PPF_{fuel} converged to approximately 1.1. In Figure 7.6b, the most-minimized TRISO distribution has a $PPF_{fuel} = 1.0872$ and an oscillating TRISO distribution along the x-axis and a packing fraction standard deviation of 0.017 across the one-third assembly. Along the x-axis, the distribution peaks at the 2nd, 5th, and 9th fuel cell columns (at 8.2cm, 14.3cm, and 22.5cm) with $PF \approx 0.08$ and has minimum points at the 4th and 7th fuel cell columns (at 12.3cm and 18.4cm) with $PF \approx 0.035$. Section 7.2.4 discusses the driving factors for the minimize PPF_{fuel} objective and explains simulation a-1c's most-minimized PPF_{fuel} TRISO distribution.

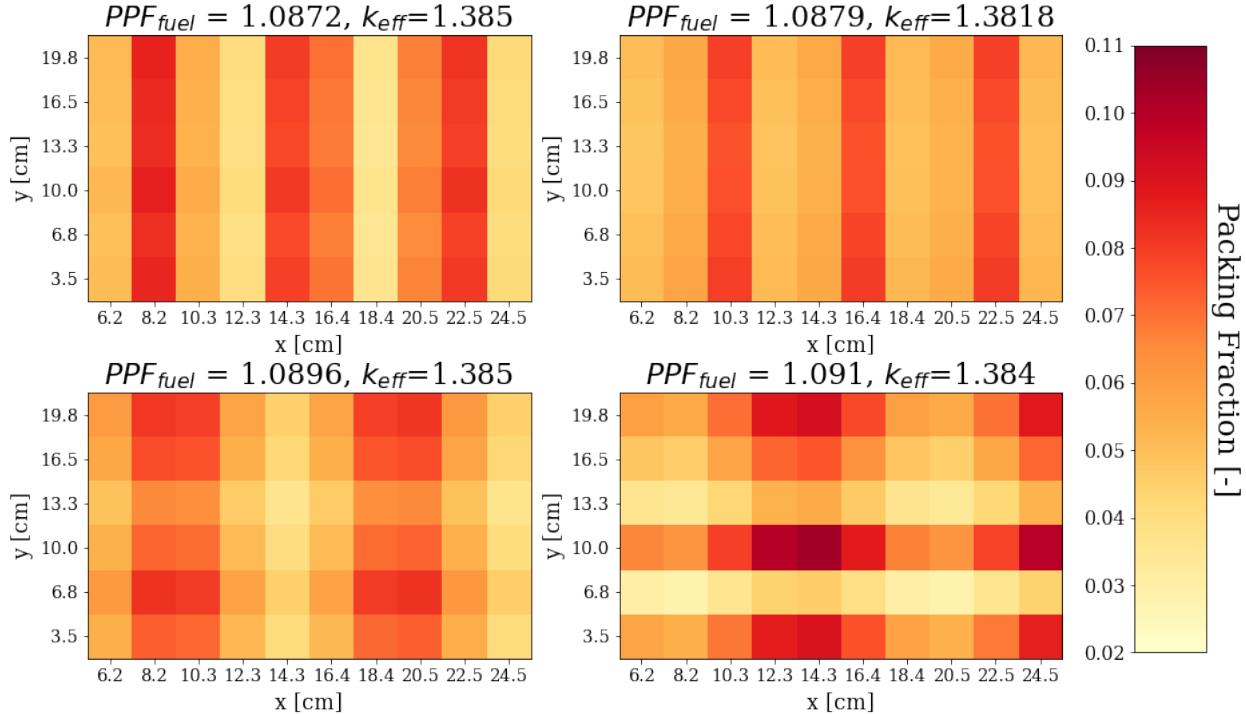
Simulation a-1f: Variation of Coolant channel shape

Table 7.8 summarizes simulation a-1f's optimization problem parameters. To remind readers each r value's orientation, Figure 7.2 shows an annotated AHTR one-third assembly's inter-plank coolant channel shapes for $r_1, r_2, r_3, r_4, r_5 = 0.3, 0.2, 0.1, 0.2, 0.3$.

Figure 7.7 shows the plots of coolant channel shape's r_1, r_2, r_3, r_4 , and r_5 values against PPF_{fuel} . The random scattering of reactor model points in Figure 7.7 demonstrates that there is

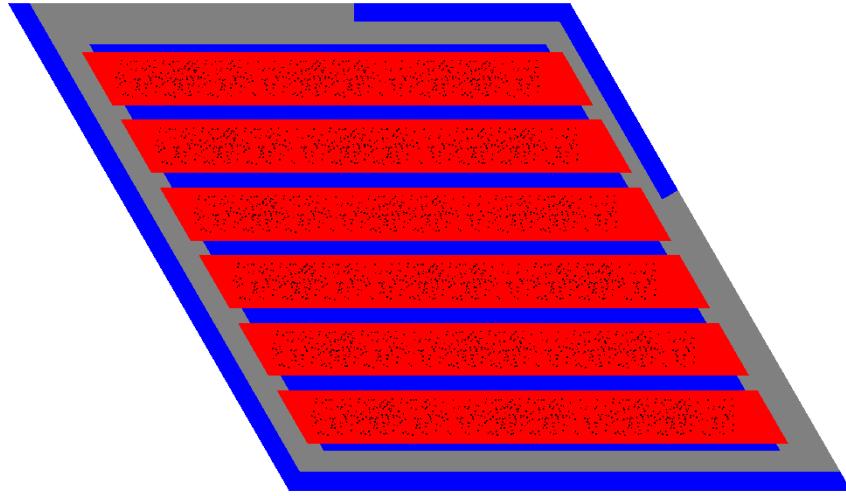


(a) Minimum, average, and maximum evolution of PPF_{fuel} in the AHTR one-third assembly.



(b) TRISO distribution for the four unique reactor models with the lowest PPF_{fuel} in the AHTR one-third assembly at the final generation.

Figure 7.6: Simulation a-1c – ROLLO single-objective optimization to minimize AHTR one-third assembly's fuel-normalized power peaking factor (PPF_{fuel}). Input parameters varied: TRISO distribution ($\rho_{TRISO}(\vec{r})$). $PF_{total} = 0.06$.



(c) AHTR one-third assembly model with the most-minimized PPF_{fuel} , corresponding to the top left TRISO distribution in Figure 7.6b. The reactor model has $PPF_{fuel} = 1.0872$ and $k_{eff} = 1.385$.

Figure 7.6: (contd.) Simulation a-1c – ROLLO single-objective optimization to minimize AHTR one-third assembly's fuel-normalized power peaking factor (PPF_{fuel}). Input parameters varied: TRISO distribution ($\rho_{TRISO}(\vec{r})$). $PF_{total} = 0.06$.

Table 7.8: Simulation a-1f Optimization Problem Parameters

Single Objective: Simulation a-1f	
Objectives	Minimize PPF_{fuel}
Input Parameter variations	coolant channel shape: $0.05 < r_1 < 0.35$ coolant channel shape: $0.05 < r_2 < 0.35$ coolant channel shape: $0.05 < r_3 < 0.35$ coolant channel shape: $0.05 < r_4 < 0.35$ coolant channel shape: $0.05 < r_5 < 0.35$
Constraints	$k_{eff} \geq 1.0$ $PF_{total} = 0.04$
Genetic Algorithm Parameters	Population size: 64 Generations: 2

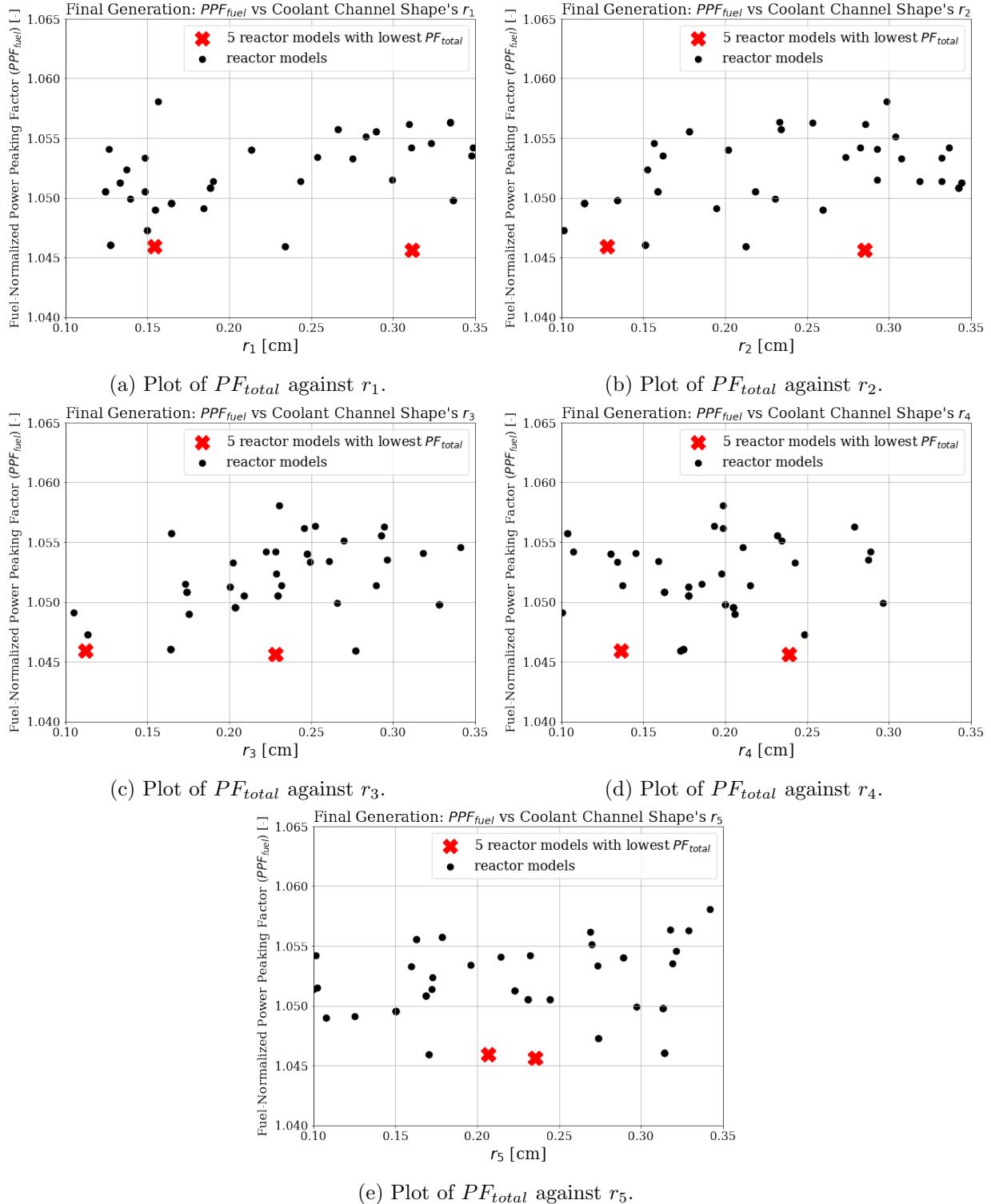


Figure 7.7: Simulation a-1f – ROLLO single-objective optimization to minimize AHTR one-third assembly's fuel-normalized power peaking factor (PPF_{fuel}). Plots of simulation a-1f final generation's reactor models PPF_{fuel} against coolant channel shape input parameters. Red crosses indicate the five reactor models with the lowest PPF_{fuel} . Some of the five reactor models with the lowest PPF_{fuel} are the same, thus their crosses overlap. Input parameters varied: total fuel packing fraction (PPF_{fuel}), and coolant channel shape (r_1, r_2, r_3, r_4, r_5).

no correlation between PPF_{fuel} and coolant channel shape's r_1, r_2, r_3, r_4 , and r_5 .

7.2.4 Single-Objective Optimization Discussion

Chapter 6 characterized the AHTR plank model's reactor optimization objectives' driving factors and their relationship with one another. This section utilizes the previous AHTR plank characterizations and conducts a comparative analysis to verify if the same driving factors apply to the AHTR one-third assembly model's optimization objectives.

Discussion: Minimize PF_{total} Objective

Simulation a-1a In Section 7.2.1's simulation a-1a, I conducted a single-objective optimization simulation to minimize the one-third assembly's total fuel packing fraction (PF_{total}) by varying PF_{total} and TRISO distribution. ROLLO found that an AHTR one-third assembly model with the most-minimized PF_{total} has a $PF_{total} = 0.0559$ and an oscillating TRISO distribution along the x-axis and y-axis, and a packing fraction standard deviation of 0.04 across the one-third assembly (Figure 7.1b).

Section 7.2.4 concluded that for the AHTR plank model, the minimize PF_{total} objective is driven by maximizing the total fission reaction rates. I ran a simulation for constant $PF_{total} = 0.0559$ TRISO distribution and compared its fission reaction rate with simulation a-1a's oscillating TRISO distribution that most-minimized PF_{total} . Figure 7.8 shows the TRISO distributions for the two compared reactor models: Figure 7.1b's most-minimized PF_{total} and the constant $PF_{total} = 0.0559$. The reactor model with the most-minimized PF_{total} has $k_{eff} = 1.3802$, and the reactor model with constant TRISO distribution has $k_{eff} = 1.3736$.

Table 7.9 compares the total fission reaction rate (OpenMC's `fission` tally) between the most-minimized PF_{total} TRISO distribution and a constant $PF_{total} = 0.0559$ TRISO distribution (both shown in Figure 7.8). The most-minimized PF_{total} TRISO distribution has 0.65% higher total fission reaction rate than the constant $PF_{total} = 0.0559$ TRISO distribution. For the same PF_{total} , the oscillating TRISO distribution enabled 660pcm higher k_{eff} than the constant TRISO distribution due to the higher total fission reaction rate. Therefore, the minimize PF_{total} objective is driven by maximizing the total fission reaction rates to find a reactor model with lower PF_{total} .

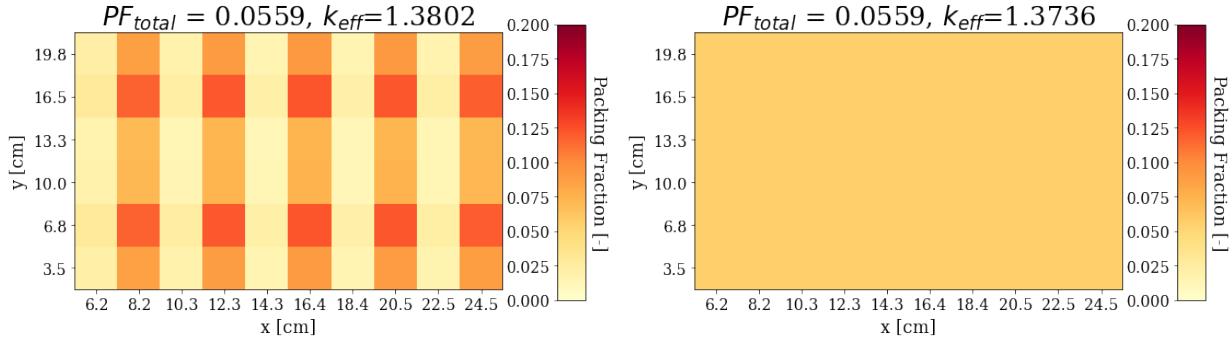


Figure 7.8: Simulation a-1a's most-minimized PF_{total} TRISO distribution (oscillating TRISO distribution) from Figure 7.1 (left) and the constant $PF_{total} = 0.0559$ TRISO distribution (right).

Table 7.9: Total fission reaction rate comparison between simulation a-1a's most-minimized PF_{total} TRISO distribution and a constant $PF_{total} = 0.0559$ TRISO distribution. Both distributions shown in Figure 7.8.

Energy Group	% of Total	Most-minimized PF_{total} Fission [reactions/src]	Flat PF_{total} Fission [reactions/src]	% Fission Difference
1	00.28	0.00165	0.00162	+2.01
2	01.56	0.00886	0.00884	+0.21
3	01.51	0.00854	0.00852	+0.23
4	96.63	0.54813	0.54465	+0.63
Total	-	0.52998	0.52656	+0.65

that meets k_{eff} constraints.

Simulation a-1d In Section 7.2.1's simulation a-1d, I conducted a single-objective optimization simulation to minimize total fuel packing fraction (PF_{total}) by varying PF_{total} and coolant channel shape. In simulation a-1d, ROLLO found no correlation between PF_{total} and coolant channel shape (demonstrated in Figure 7.3).

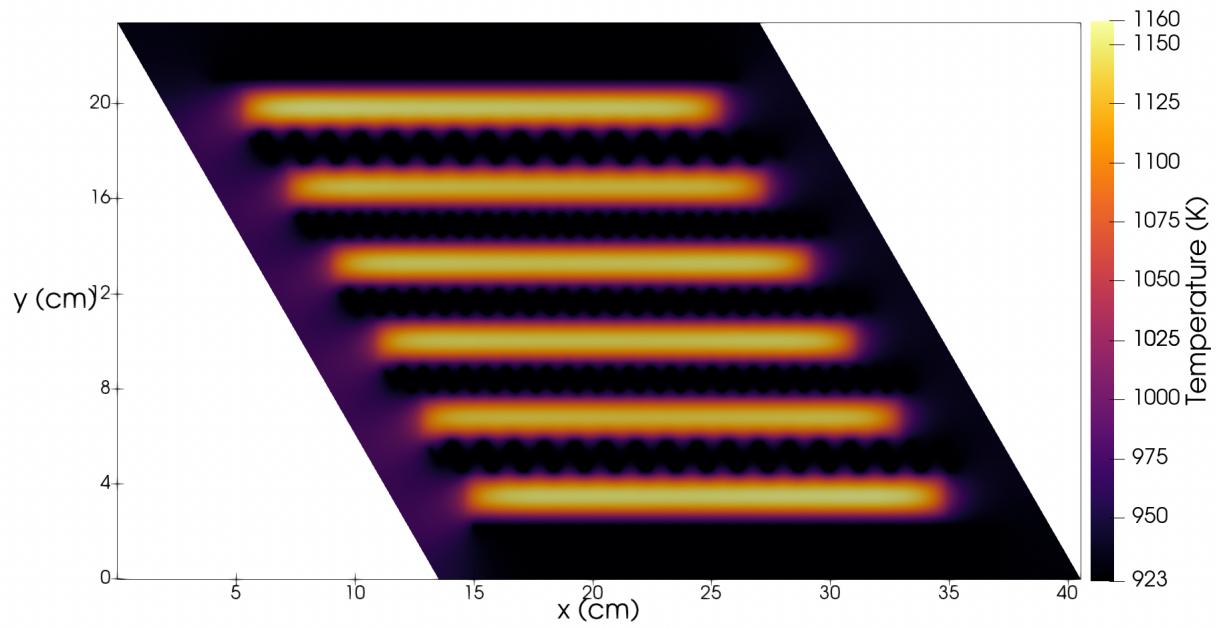
Summary I confirmed that the minimize PF_{total} objective for the AHTR one-third assembly model is also driven by maximizing the total fission reaction rates. The minimize PF_{total} objective influences oscillations in the TRISO distribution. The objective does not correlate with the coolant channel shape.

Discussion: Minimize T_{max} Objective

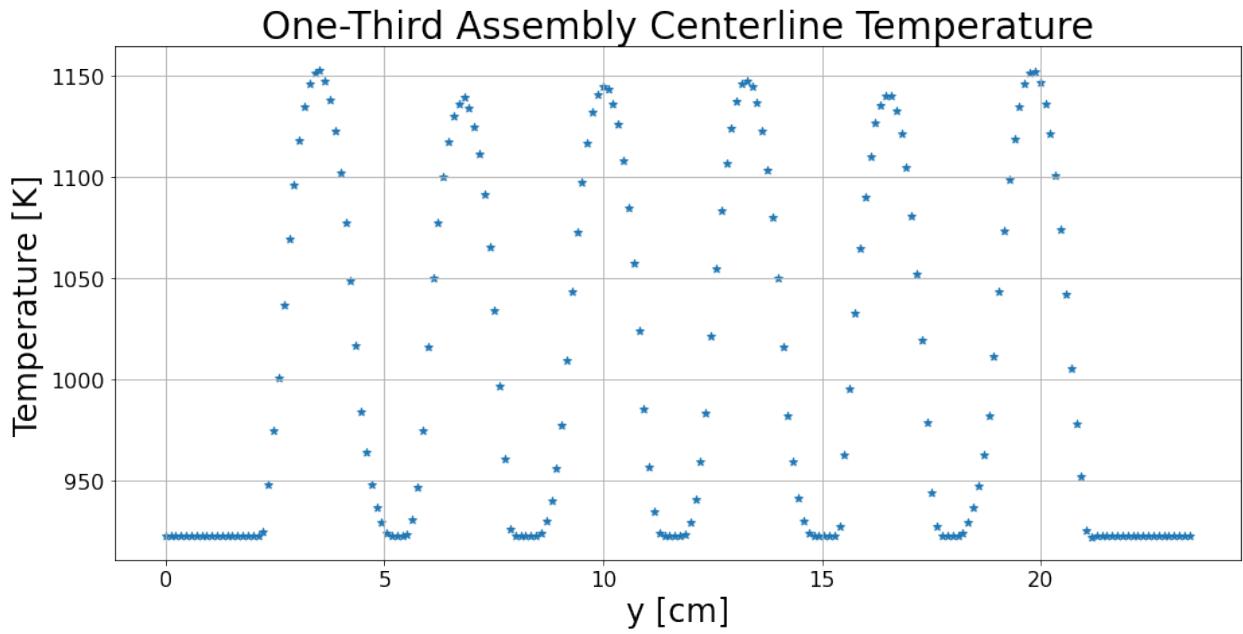
Simulation a-1b In Section 7.2.2's simulation a-1b, I conducted a single-objective optimization simulation to minimize the one-third assembly's maximum temperature (T_{max}) by varying TRISO distribution. In simulation a-1b, ROLLO found that for $PF_{total} = 0.06$, the reactor model with the most-minimized T_{max} has a $T_{max} = 1161.28K$ with an almost constant TRISO distribution (Figure 7.4b).

Simulation a-1e In Section 7.2.2's simulation a-1e, I conducted a single-objective optimization simulation to minimize the one-third assembly's maximum temperature (T_{max}) by varying coolant channel shape. In simulation a-1e, ROLLO found a negative linear correlation between the one-third assembly's T_{max} and r_1 and r_5 , but no correlation with r_2 , r_3 , and r_4 , shown in Figure 7.5.

Figures 7.9a and 7.9b show the 2D temperature distribution and centerline temperature for simulation a-1e's one-third assembly model with the most-minimized T_{max} (Figure 7.5b). Figure 7.9 demonstrates that for simulation a-1e's most-minimized T_{max} reactor model, the temperature peaks in the top and bottom graphite planks. r_1 and r_5 control the coolant channel shape of the FLiBe channels closest to the top and bottom graphite planks. This explains why ROLLO found a negative linear correlation between the one-third assembly's T_{max} and r_1 and r_5 . To



(a) 2D temperature distribution.



(b) Centerline temperature. AHTR assembly's centerline is the white line in Figure 5.15.

Figure 7.9: Simulation a-1e's most-minimized T_{max} one-third assembly reactor model's temperature distribution.

minimize the maximum one-third assembly temperature, ROLLO maximized r_1 and r_5 to enable enhanced cooling in the top and bottom graphite planks. Thus, depending on the temperature distribution in a one-third assembly, the FLiBe channels (corresponding to r_1, r_2, r_3, r_4, r_5) closest to the temperature peaks will be most important to minimizing T_{max} .

Summary I verified that, similar to the AHTR plank model, a flatter TRISO distribution minimizes the one-third assembly's T_{max} . For the one-third assembly, the FLiBe channels (corresponding to r_1, r_2, r_3, r_4, r_5) closest to the temperature peaks will be most important to minimizing maximum one-third assembly temperature and thus, will show a negative correlation with T_{max} . Figures 7.4a and 7.5a show that during the genetic algorithm optimization process, the average T_{max} due to TRISO variation decreased by 150K, while average T_{max} due to the coolant channel shape variation only decreased by 5K, demonstrating that TRISO distribution influences the minimize T_{max} objective more than the coolant channel shape.

Discussion: Minimize PPF_{fuel} Objective

Simulation a-1c In Section 7.2.3's simulation a-1c, I conducted a single-objective optimization simulation to minimize fuel-normalized power peaking factor (PPF_{fuel}) by varying TRISO distribution. In simulation a-1c, ROLLO found that for $PF_{total} = 0.06$, the reactor model with the most-minimized PPF_{fuel} has a $PPF_{fuel} = 1.0872$, an oscillating TRISO distribution along the x-axis, and a packing fraction standard deviation of 0.017 across the one-third assembly (Figure 7.6b).

Section 7.2.4 concluded that for the AHTR plank model, the minimize PPF_{fuel} objective is driven by flattening thermal (Group 4) flux distribution. I compare the flux distributions for simulation a-1c's most-minimized PPF_{fuel} reactor model ($PPF_{fuel} = 1.0872$) and the reactor model in simulation a-1c's final generation with the highest $PPF_{fuel} = 1.2431$. Figure 7.10 shows the TRISO distributions for the compared reactor models.

Figure 7.11 compares the 4 energy group flux distributions between simulation a-1c's most-minimized PPF_{fuel} TRISO distribution and highest PPF_{fuel} TRISO distribution (both shown in Figure 7.10). In Figure 7.11, the reactor model with the highest PPF_{fuel} 's Group 4 flux dips in the center of the one-third assembly likely due to spatial self-shielding effects. In the highest PPF_{fuel}

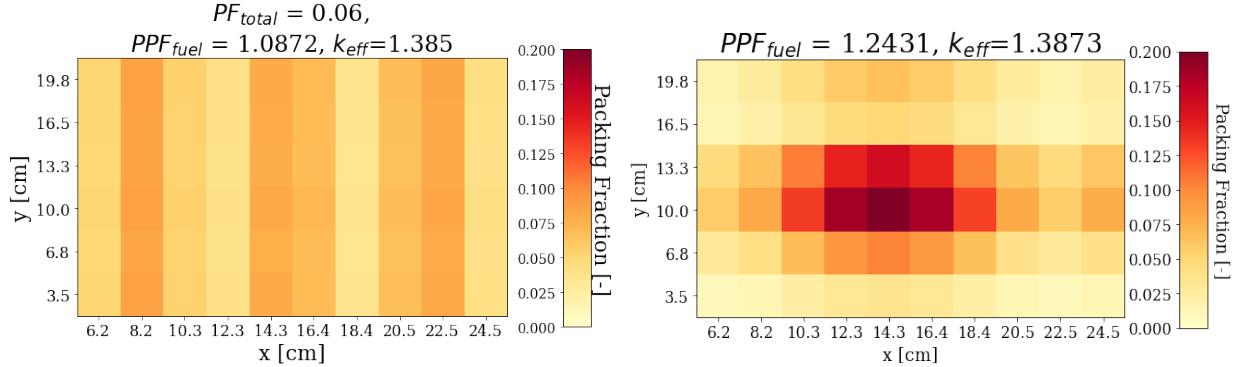


Figure 7.10: Simulation a-1c's most-minimized PPF_{fuel} TRISO distribution from Figure 7.6 (left) and the highest PPF_{fuel} TRISO distribution (right).

reactor model's Group 1 flux, there is a peak in fast neutrons born in the one-third assembly's center, the fast neutrons are moderated in the graphite matrix and graphite structure. The self-shielding neutrons are more likely absorbed at the fuel regions at the one-third assembly's top, bottom, and sides, near the pure graphite structure moderating regions. The outer sides of the one-third assembly absorb these neutrons and geometrically shield the one-third assembly's center from neutron flux, leading to a relatively lower group 4 thermal flux in the center.

Table 7.10 quantifies the total flux differences per energy group between the reactor models. I used a 100×100 mesh to tally the flux values for each energy group within the one-third assembly model.

Table 7.10: Flux value comparison between the two reactor models in Figure 7.10: simulation a-1c's most-minimized PPF_{fuel} reactor model with $PPF_{fuel} = 1.0872$ and simulation a-1c's reactor model with the highest $PPF_{fuel} = 1.2431$. Energy Group 1: $E > 9.1188 \times 10^{-3}$ MeV, Energy Group 2: $2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$ MeV, Energy Group 3: $1.8556 \times 10^{-5} < E < 2.9023 \times 10^{-5}$ MeV, Energy Group 4: $1.0 \times 10^{-12} < E < 1.8554 \times 10^{-6}$ MeV.

Energy Group	$\max(\phi)/\min(\phi)$ Most-minimized PPF_{fuel} TRISO Distribution	$\max(\phi)/\min(\phi)$ Highest PPF_{fuel} TRISO Distribution	% Difference
1	1.825	2.608	-30.00
2	1.341	1.386	-3.18
3	1.302	1.334	-2.43
4	1.319	1.331	-0.85

In energy group 4, the most-minimized PPF_{fuel} flux distribution is 0.85% flatter than the reactor model with the highest $PPF_{fuel} = 1.2431$. These results verify that, similar to the AHTR

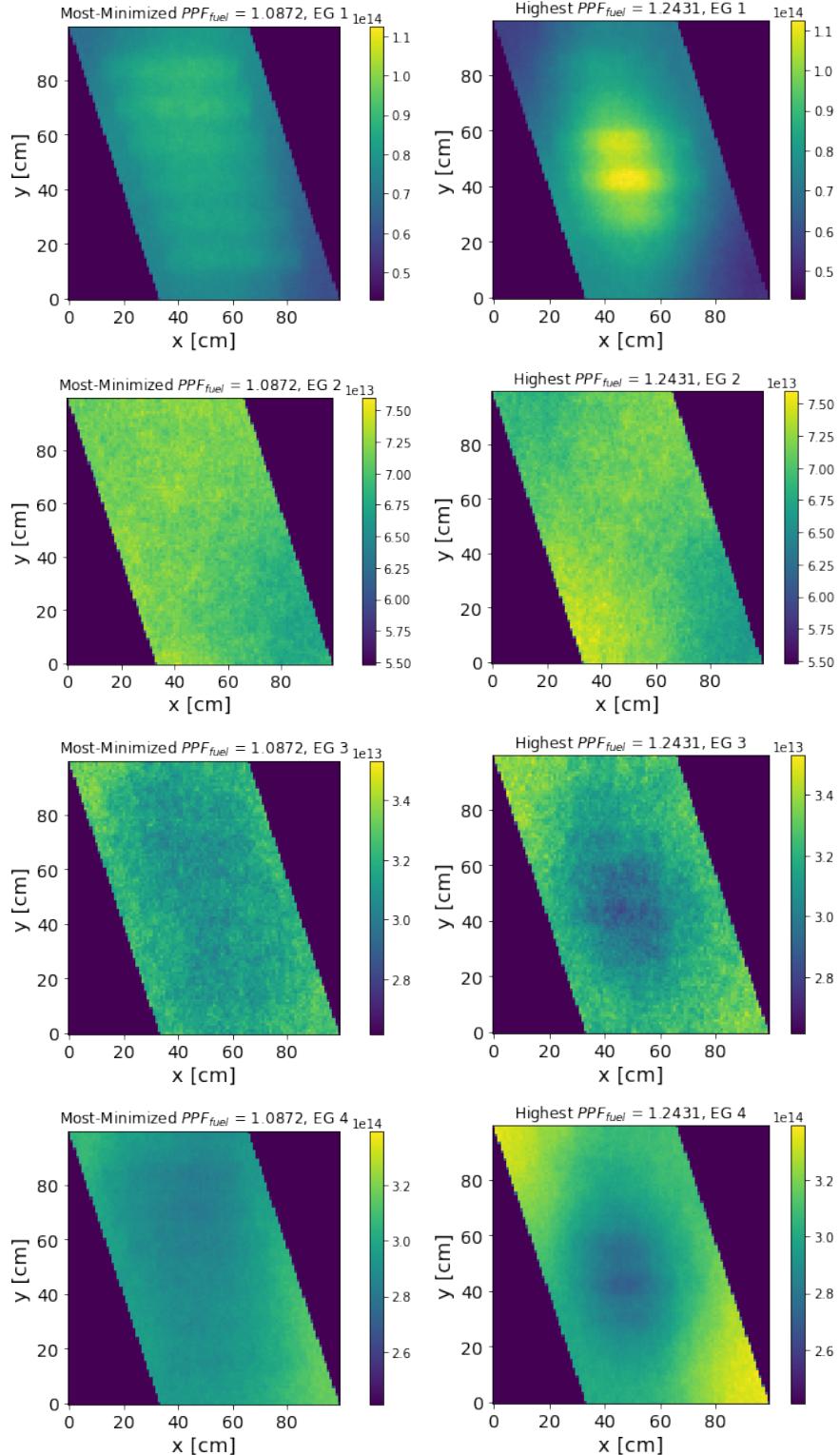


Figure 7.11: AHTR one-third assembly's flux comparison in units of $n \cdot cm^{-2} \cdot s^{-1}$ between the two reactor models in Figure 7.10: simulation a-1c's most-minimized PPF_{fuel} reactor model with $PPF_{fuel} = 1.0872$ and simulation a-1c's reactor model with the highest $PPF_{fuel} = 1.2431$. Energy Group 1: $E > 9.1188 \times 10^{-3}$ MeV, Energy Group 2: $2.9023 \times 10^{-5} < E < 9.1188 \times 10^{-3}$ MeV, Energy Group 3: $1.8556 \times 10^{-5} < E < 2.9023 \times 10^{-5}$ MeV, Energy Group 4: $1.0 \times 10^{-12} < E < 1.8554 \times 10^{-6}$ MeV.

plank model, the AHTR one-third assembly model's minimize PPF_{fuel} objective is also driven by flattening thermal (Group 4) flux distribution since $\max(\Phi_j) \div \text{ave}(\Phi_j) \propto PPF_{fuel}$ (Equation 6.5).

Simulation p-1f In Section 7.2.3's simulation a-1f, I conducted a single-objective optimization simulation to minimize fuel-normalized power peaking factor (PPF_{fuel}) by varying coolant channel shape. In simulation a-1f, ROLLO found no correlation between PPF_{fuel} and coolant channel shape (demonstrated in Figure 7.7).

Summary I verified that the minimize PPF_{fuel} objective for the AHTR one-third assembly model is also driven by flattening thermal (Group 4) flux distribution and, by extension, the fission distribution in the fuel. The minimize PPF_{fuel} objective influences PF_{total} and oscillations in the TRISO distribution. The objective does not correlate with the coolant channel shape.

7.2.5 Single-Objective Optimization Major Takeaways

Section 7.2 utilizes the AHTR plank characterizations in Chapter 6 to conduct a comparative analysis to verify if the same driving factors apply to the AHTR one-third assembly model's optimization objectives. In Section 7.2's single-objective optimization simulations: a-1a, a-1b, a-1c, a-1d, a-1e, a-1f; and Section 7.2.4's discussions, I verified that each of the one-third assembly objective follows the same driving factors as the AHTR plank optimization objectives and described each objective's relationship with each input parameter. I determined that ROLLO flattens TRISO distribution and maximizes the coolant channel shape's radius values (r_1, r_2, r_3, r_4, r_5) that are located close to the reactor model's temperature peak to achieve the minimize T_{max} objective. The minimize PF_{total} objective is driven by maximizing the one-third assembly's total fission reaction rate and influences oscillations in the TRISO distribution to achieve the objective. The minimize PPF_{fuel} objective is driven by flattening the one-third assembly's thermal flux distribution and influences PF_{total} and oscillations in the TRISO distribution to achieve the objective. Both the minimize PF_{total} and minimize PPF_{fuel} objectives do not correlate with the coolant channel shape. Simulation a-1b and a-1e results demonstrated that coolant channel shape variation does not have as high of an impact on T_{max} as TRISO distribution variation. Since the variations in coolant channel shape only impact the minimize maximum temperature (T_{max}) objective, I do not conduct

double-objective optimization for coolant channel shape variations.

7.3 AHTR One-Third Assembly: Double-Objective Optimization Results

This section reports the AHTR one-third assembly’s ROLLO double-objective optimization results. The previous section’s single-objective optimization results inform the double-objective optimization simulations in this section. Section 7.2 concluded that coolant channel shape variations only correlate with the minimize T_{max} objective, thus, I do not conduct double-objective optimization for coolant channel shape variations. I run double-objective optimization simulations that vary total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), for combinations of the three objectives (minimize PF_{total} , T_{max} , and PPF_{fuel}). Earlier in this chapter, Table 7.2 summarized the double-objective simulations: a-2a, a-2b, and a-2c.

As described in Section 2.3, multi-objective optimization returns multiple optimal solutions that meet each objective to varying degrees; this set of solutions is the Pareto front [66]. For each solution in the Pareto front, none of the objective functions can be improved without degrading another objective. An ideal optimization method for a multi-objective problem like reactor design optimization should find widely spread out reactor model solutions in the Pareto front [66]. Thus, I report on the optimal reactor models on the Pareto front for the multi-objective optimization problems in this section and Section 7.4. Several optimal solutions will be highlighted in the Pareto front figures.

To ensure that the multi-objective optimization problems are converged, I report the hypervolume values for each generation. As previously described in Section 4.4.2, the hypervolume indicator quantifies the Pareto front’s goodness (bigger = better). The hypervolume is calculated by finding the volume between the reference point and the objective values of the Pareto front’s reactor models. The reference point must be selected to ensure that every reactor model falls within the hypervolume; thus, I use a different reference point for each optimization problem. I use the hypervolume values to determine if the simulation converges across generations and do not compare them across different optimization simulations. The hypervolume is converging if the difference

between generations' hypervolume values is getting smaller.

For a single generation with 128 reactor models, OpenMC and Moltres software must be run for each reactor model, resulting in high computational cost. For each optimization simulation, I must balance convergence and computational cost. ROLLO's purpose is to help the human reactor designer narrow down reactor design the search space, therefore, the reactor designer uses the computational power available to them, to narrow down the search space as much as possible. Section 4.5.1 described that the reactor designer should plot the Pareto front and observe the hypervolume values to evaluate if they are confident about the final solution set. Thus, I determine if convergence criteria is met by evaluating if the difference between generations' hypervolume values are getting smaller. If a multi-objective optimization problem's hypervolume converges earlier than the five generations I intended to run (determined in Section 5.5.2), I stop the simulation at that generation. However, if the multi-objective optimization problem's hypervolume does not converge by generation 5, I run the problem for a few more generations until satisfactory convergence is achieved.

7.3.1 a-2a: Minimize PF_{total} and T_{max}

This section reports results from the double-objective optimization simulation a-2a; the objectives minimized are total fuel packing fraction (PF_{total}) and maximum temperature (T_{max}). Table 7.11 summarizes simulation a-2a's optimization problem parameters.

Table 7.11: Simulation a-2a Optimization Problem Parameters

Two Objectives: Simulation a-2a	
Objectives	Minimize PF_{total} Minimize T_{max}
Input parameter variations	$0.05 \leq PF_{total} \leq 0.07$ $\rho_{TRISO}(\vec{r}): 0 \leq a \leq 2, 0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r}): 0 \leq b \leq \frac{\pi}{2}, 0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 \leq c \leq 2\pi, 0 \leq f \leq 2\pi$
Constraints	$k_{eff} \geq 1.38$
Genetic algorithm parameters	Population size: 128 Generations: 5

Table 7.12 shows the hypervolume value at each generation, confirming that simulation a-2a converges by generation 5.

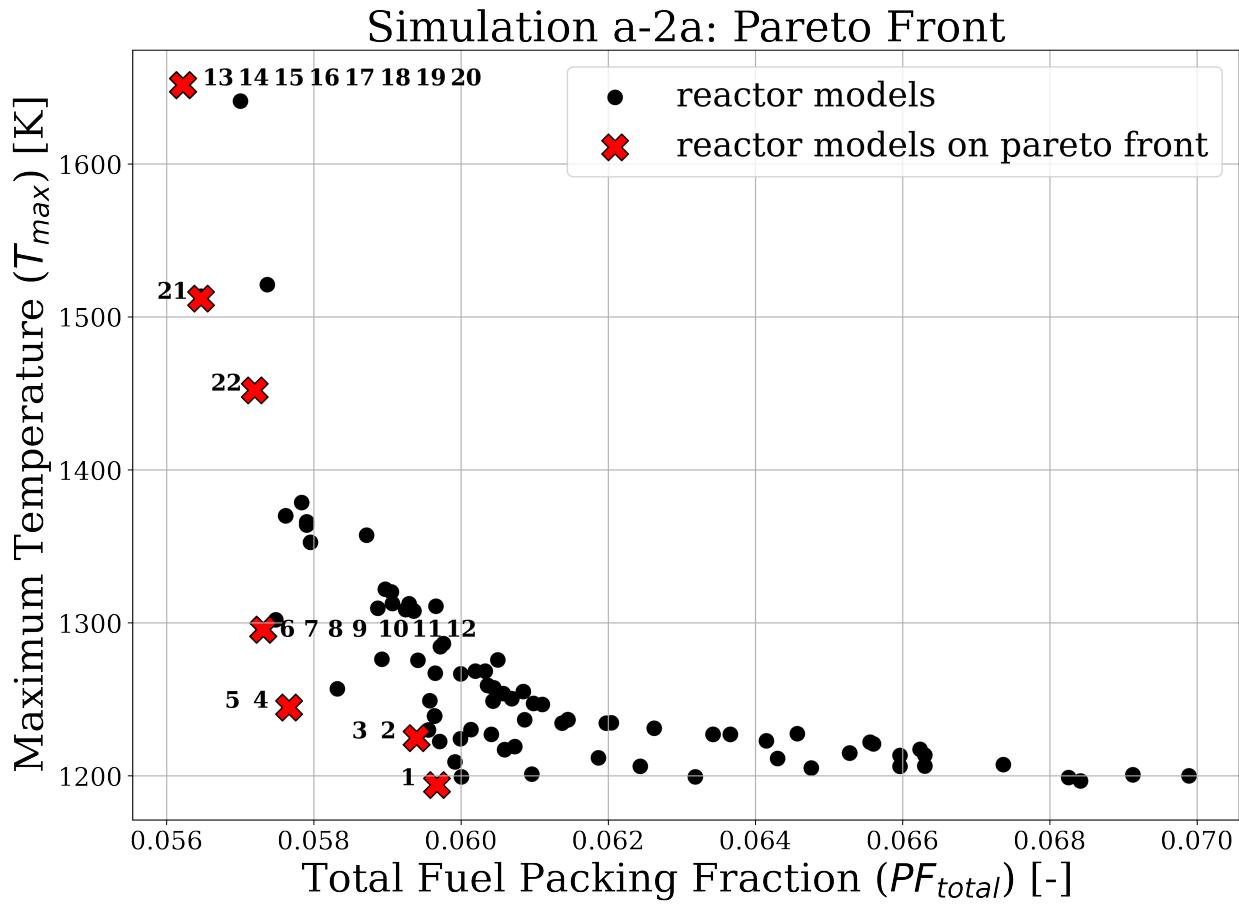
Table 7.12: Simulation a-2a hypervolume values at each generation.

Two Objectives: Simulation a-2a	
Reference point: (0.07, 1700)	
Generation	Hypervolume [-]
1	6.0090
2	6.0859
3	6.2220
4	6.3379
5	6.4664

Figure 7.12a shows a plot of the final generation’s reactor models’ PF_{total} against T_{max} ; crosses mark the reactor models that fall on the Pareto front. Figure 7.12b shows the 22 TRISO packing fraction distributions in the final generation, labeled numerically, that fall on the Pareto front.

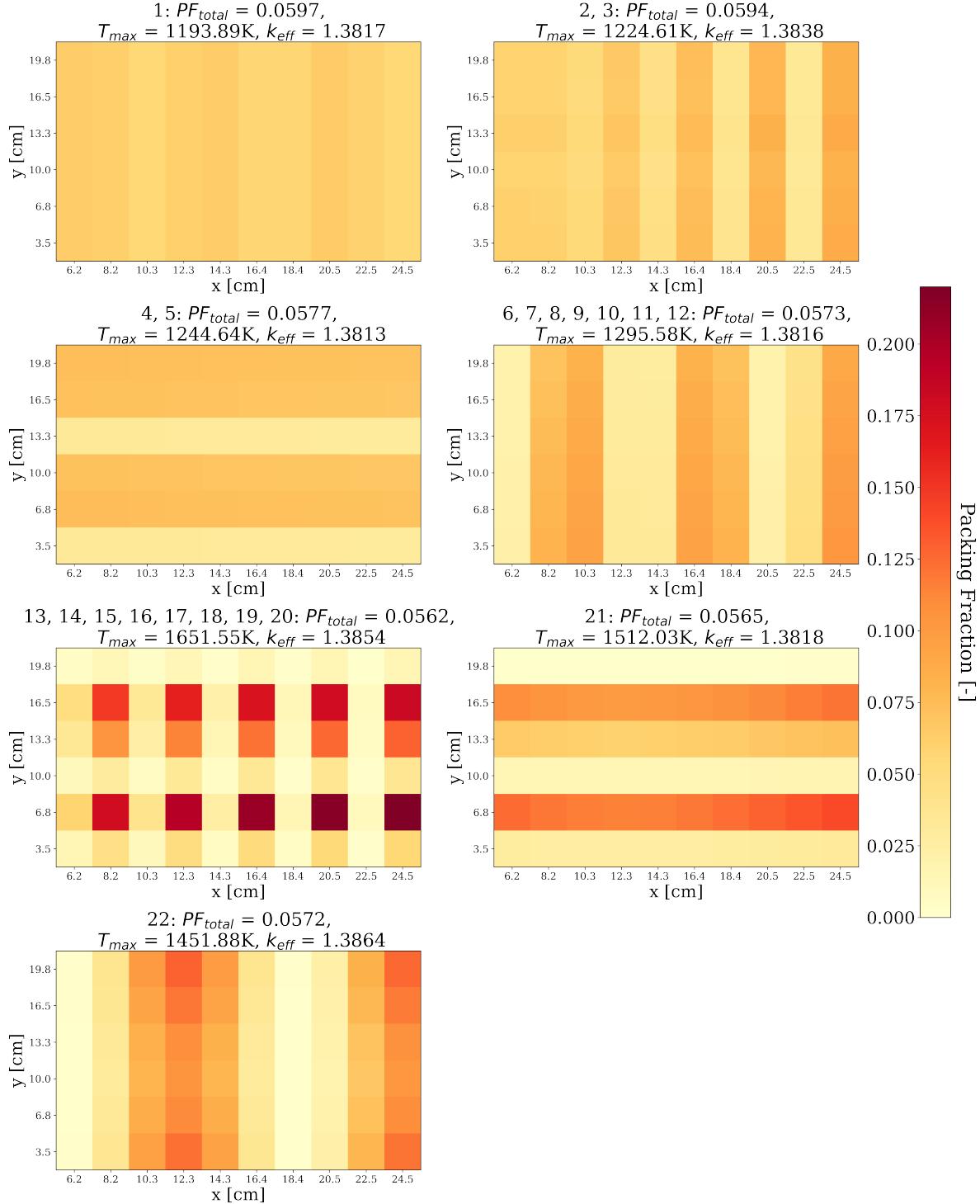
Figure 7.12a shows that minimize PF_{total} and minimize T_{max} are competing objectives. That is, by minimizing T_{max} we increase PF_{total} and by minimizing PF_{total} we increase T_{max} . In Figure 7.12, the one-third assembly model with the most-minimized PF_{total} and highest T_{max} are reactor models 13 to 20. These models have an oscillating TRISO distribution along the x-axis and y-axis, and a packing fraction standard deviation of 0.066 across the one-third assembly. Along the x-axis, the distribution peaks at the even fuel cell columns (at 8.2cm, and 12.3cm, 16.4cm, 20.5cm, and 24.5cm) and has minimum points at the odd fuel cell columns (at 6.2cm, 10.3cm, 14.3cm, 18.4cm, and 22.5cm). The even fuel cell columns have a ~ 0.18 y-axis variation with peaks of $PF \approx 0.21$.

In Figure 7.12, the one-third assembly model with the most-minimized T_{max} and highest PF_{total} is reactor model 1. Reactor model 1 has an almost constant TRISO packing fraction distribution with a packing fraction standard deviation of 0.004 across the one-third assembly. The one-third assembly model that minimizes both PF_{total} and T_{max} to an equal extent based on visual inspection of the Pareto Front (Figure 7.12a) are reactor models 4 and 5. Reactor models 4 and 5 have an oscillating TRISO distribution along the y-axis and a packing fraction standard deviation of 0.018 across the one-third assembly. Along the y-axis, the distribution peaks at the 2nd, 3rd, 5th, and 6th rows (at 6.8cm, 10.0cm, 16.5cm, and 19.8cm) with $PF \approx 0.07$ and has minimum points at the 1st and 4th rows (at 3.5cm and 13.3cm) with $PF \approx 0.03$. Section 7.3.4 discusses and explains simulation a-2a’s results.



(a) Plot of final generation's reactor models' PF_{total} against T_{max} . Crosses indicate the reactor models on the Pareto front. Annotated numbers on each cross correspond to TRISO distributions in the plot below.

Figure 7.12: Simulation a-2a – ROLLO double-objective optimization to minimize total fuel packing fraction (PF_{total}) and maximum temperature (T_{max}) in the one-third assembly. Input parameters varied: total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



(b) TRISO distribution for the 22 reactor models on the Pareto front. Numbered reactor models correspond to numbered crosses in the plot above. Note that some models have identical distributions, resulting in the 7 plots in this subfigure.

Figure 7.12: (contd.) Simulation a-2a – ROLLO double-objective optimization to minimize total fuel packing fraction (PF_{total}) and maximum temperature (T_{max}) in the one-third assembly. Input parameters varied: total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

7.3.2 a-2b: Minimize PF_{total} and PPF_{fuel}

This section reports results from the double-objective optimization simulation a-2b; the objectives minimized are total fuel packing fraction (PF_{total}) and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Table 7.13 summarizes simulation a-2b's optimization problem parameters.

Table 7.13: Simulation a-2b Optimization Problem Parameters

Two Objectives: Simulation a-2b	
Objectives	Minimize PF_{total} Minimize PPF_{fuel}
Input parameter variations	$0.05 \leq PF_{total} \leq 0.07$ $\rho_{TRISO}(\vec{r}): 0 \leq a \leq 2, 0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r}): 0 \leq b \leq \frac{\pi}{2}, 0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 \leq c \leq 2\pi, 0 \leq f \leq 2\pi$
Constraints	$k_{eff} \geq 1.38$
Genetic algorithm parameters	Population size: 128 Generations: 5

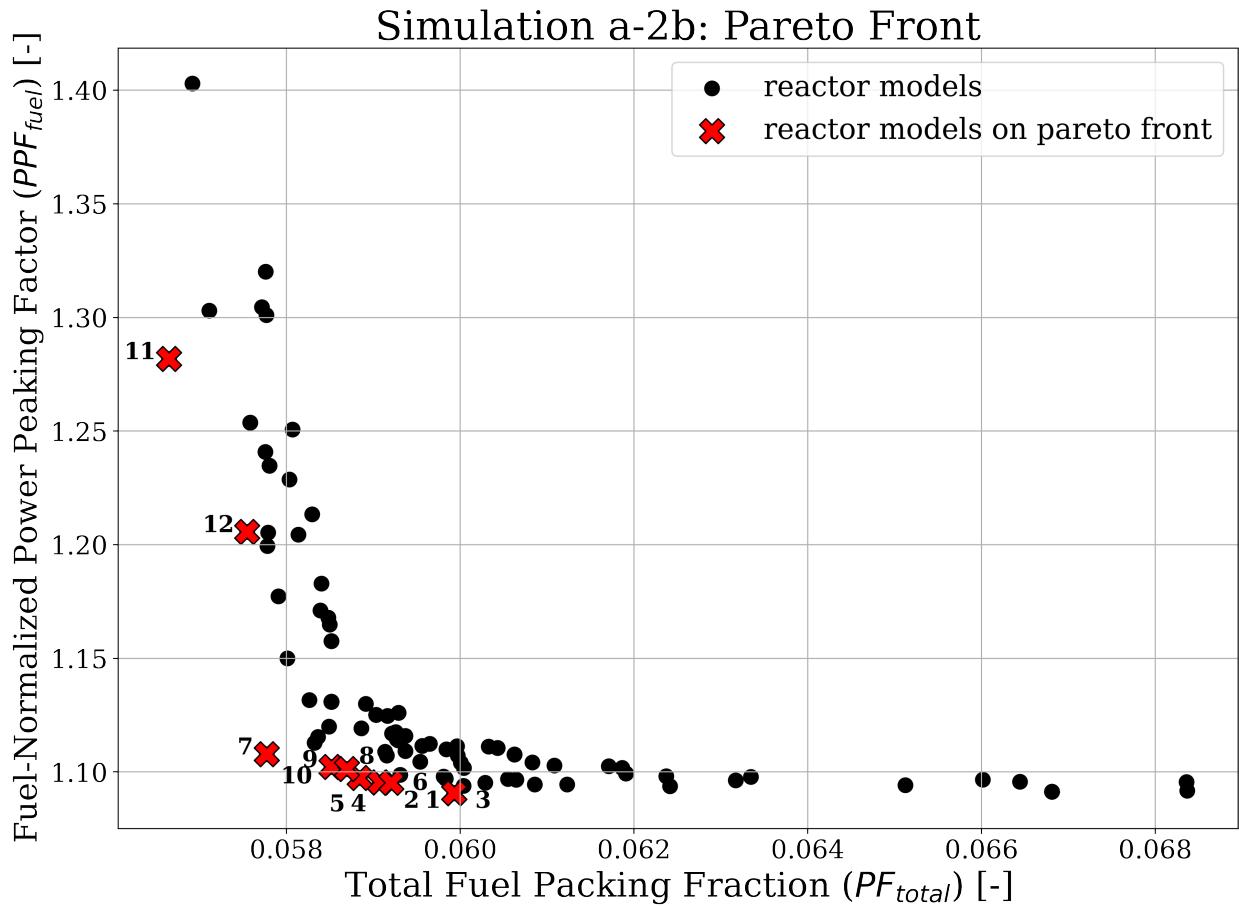
Table 7.14 shows the hypervolume value at each generation, confirming that simulation a-2b converges by generation 5.

Table 7.14: Simulation a-2b hypervolume values at each generation.

Two Objectives: Simulation a-2b	
Reference point: (0.07, 1.9)	
Generation	Hypervolume [-]
1	0.00989
2	0.00991
3	0.00997
4	0.01054
5	0.01058

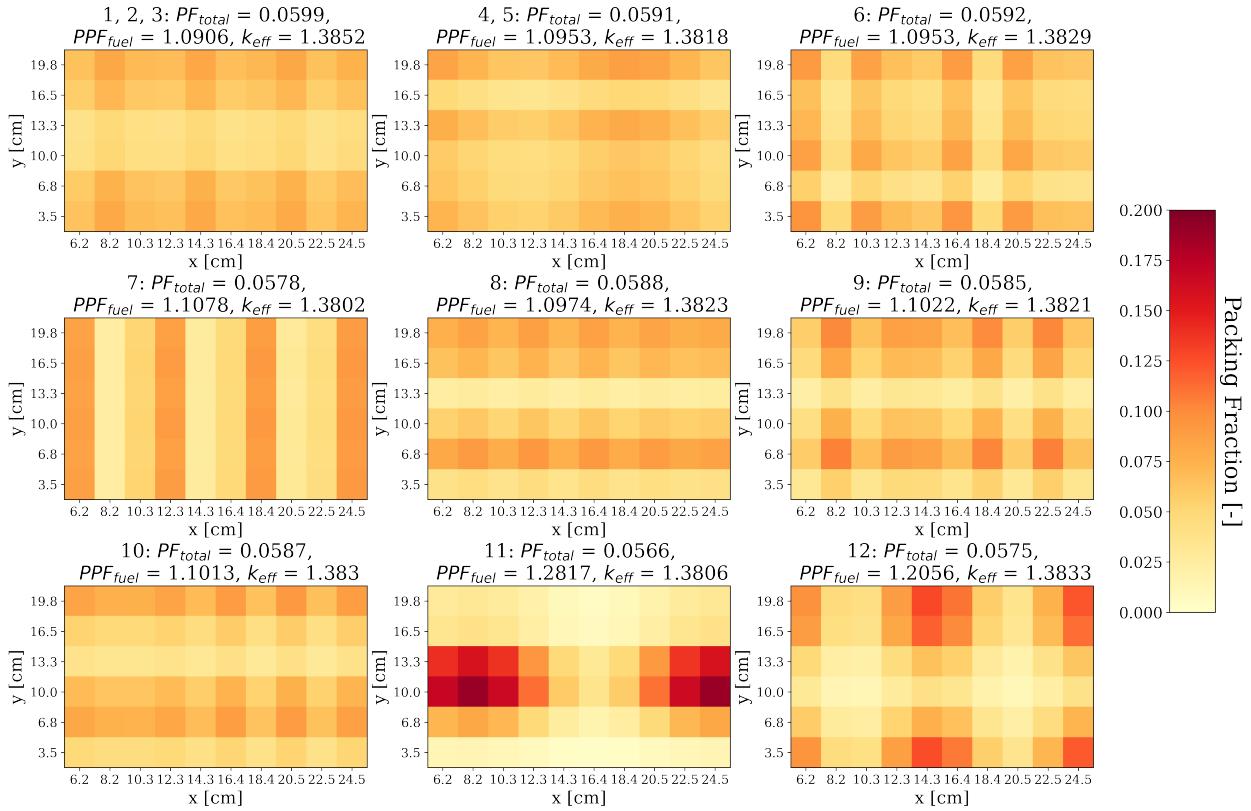
Figure 7.13a shows a plot of the final generation's reactor models' PF_{total} against PPF_{fuel} ; crosses mark the reactor models that fall on the Pareto front. Figure 7.13b shows the 12 TRISO packing fraction distributions, labeled numerically, that fall on the Pareto front.

Figure 7.13a shows that minimize PF_{total} and minimize PPF_{fuel} are competing objectives. In Figure 7.13, the one-third assembly model with the most-minimized PF_{total} and highest PPF_{fuel} is reactor model 11. Reactor model 11 has an oscillating TRISO distribution along the x-axis and y-axis, and a packing fraction standard deviation of 0.053 across the one-third assembly. Along the



(a) Plot of final generation's reactor models' PF_{total} against PPF_{fuel} . Crosses indicate the reactor models on the Pareto front. Annotated numbers on each cross correspond to TRISO distributions in Figure 7.13b.

Figure 7.13: Simulation a-2b – ROLLO double-objective optimization to minimize total fuel packing fraction (PF_{total}) and normalized power peaking factor (PPF_{fuel}) in the AHTR one-third assembly. Input parameters varied: PF_{total} and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



(b) TRISO distribution for the 12 reactor models on the Pareto front. Numbered reactor models correspond to numbered crosses in Figure 7.13a. Note that some models have identical distributions, resulting in the 9 plots in this subfigure.

Figure 7.13: (contd.) Simulation a-2b – ROLLO double-objective optimization to minimize total fuel packing fraction (PF_{total}) and normalized power peaking factor (PPF_{fuel}) in the AHTR one-third assembly. Input parameters varied: PF_{total} and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

y-axis, the distribution peaks at the 3rd and 4th fuel cell rows (at 10.0cm and 13.3cm) and has minimum points at the 1st, 5th, and 6th fuel cell rows (at 3.5cm, 16.5cm, 19.8cm). The 3rd and 4th rows have the largest x-axis variation of ~0.14 with peaks of $PF \approx 0.17$. The 1st, 5th, and 6th row has the smallest x-axis variation of ~0.02 with minimums of $PF \approx 0.005$.

In Figure 7.13, the one-third assembly model with the most-minimized PPF_{fuel} and highest PF_{total} is reactor model 1. Reactor model 1 has an oscillating TRISO distribution along the x-axis and y-axis and a packing fraction standard deviation of 0.013 across the one-third assembly. Along the x-axis, the distribution peaks at the 2nd, 5th, and 8th fuel cell columns (at 8.2cm, 14.3cm, and 20.5cm) and has minimum points at the 1st, 6th, and 9th (at 6.2cm, 16.4cm, and 22.5cm). The 2nd, 5th, and 8th columns have the largest y-axis variation of ~0.034 with peaks of $PF \approx 0.08$. The 1st, 6th, and 9th columns have the smallest y-axis variation of ~0.027 with minimums of $PF \approx 0.038$. On the y-axis, the distribution has peaks at the top and bottom row (at 3.5cm and 19.8cm) and has a minimum point in the center rows (at 10.0cm and 13.3cm). The top and bottom row have the largest x-axis variation of ~0.018 with peaks of $PF \approx 0.08$. The center rows have the smallest x-axis variation of ~0.011 with minimums of $PF \approx 0.038$.

The one-third assembly model that minimizes both PF_{total} and PPF_{fuel} to an equal extent based on visual inspection of the Pareto Front (Figure 7.13a) is reactor model 5. Like reactor models 11 and 1, reactor model 5 has an oscillating TRISO distribution along the x-axis and y-axis. Reactor model 5 has a packing fraction standard deviation of 0.013 across the one-third assembly. Along the x-axis, the distribution peaks at the 1st and 7th fuel cell columns (at 6.2cm and 18.4cm) and has minimum points in the 3rd, 4th, and 10th fuel cell columns (at 10.3cm, 12.3cm, and 24.5cm). Along the x-axis, all the columns have a similar x-axis variation of ~0.03. Along the y-axis, the distribution peaks at the 4th and 6th fuel cell rows (at 13.3cm and 19.8cm) and has minimum points at the 5th fuel row (at 16.5cm). The 4th and 6th rows have the largest x-axis variation of ~0.024 with peaks of $PF \approx 0.08$. The 5th row has the smallest x-axis variation of ~0.015 with minimums of $PF \approx 0.035$. Sections 7.3.4 discusses and explains simulation a-2b's results.

7.3.3 a-2c: Minimize T_{max} and PPF_{fuel}

This section reports results from the double-objective optimization simulation a-2c; minimized objectives are maximum temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Table 7.15 summarizes simulation a-2c's optimization problem parameters.

Table 7.15: Simulation a-2c Optimization Problem Parameters

Two Objectives: Simulation a-2c	
Objectives	Minimize T_{max} Minimize PPF_{fuel}
Input parameter variations	$\rho_{TRISO}(\vec{r})$: $0 \leq a \leq 2$, $0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r})$: $0 \leq b \leq \frac{\pi}{2}$, $0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r})$: $0 \leq c \leq 2\pi$, $0 \leq f \leq 2\pi$
Constraints	$k_{eff} \geq 1.38$ $PF_{total} = 0.06$
Genetic algorithm parameters	Population size: 128 Generations: 2

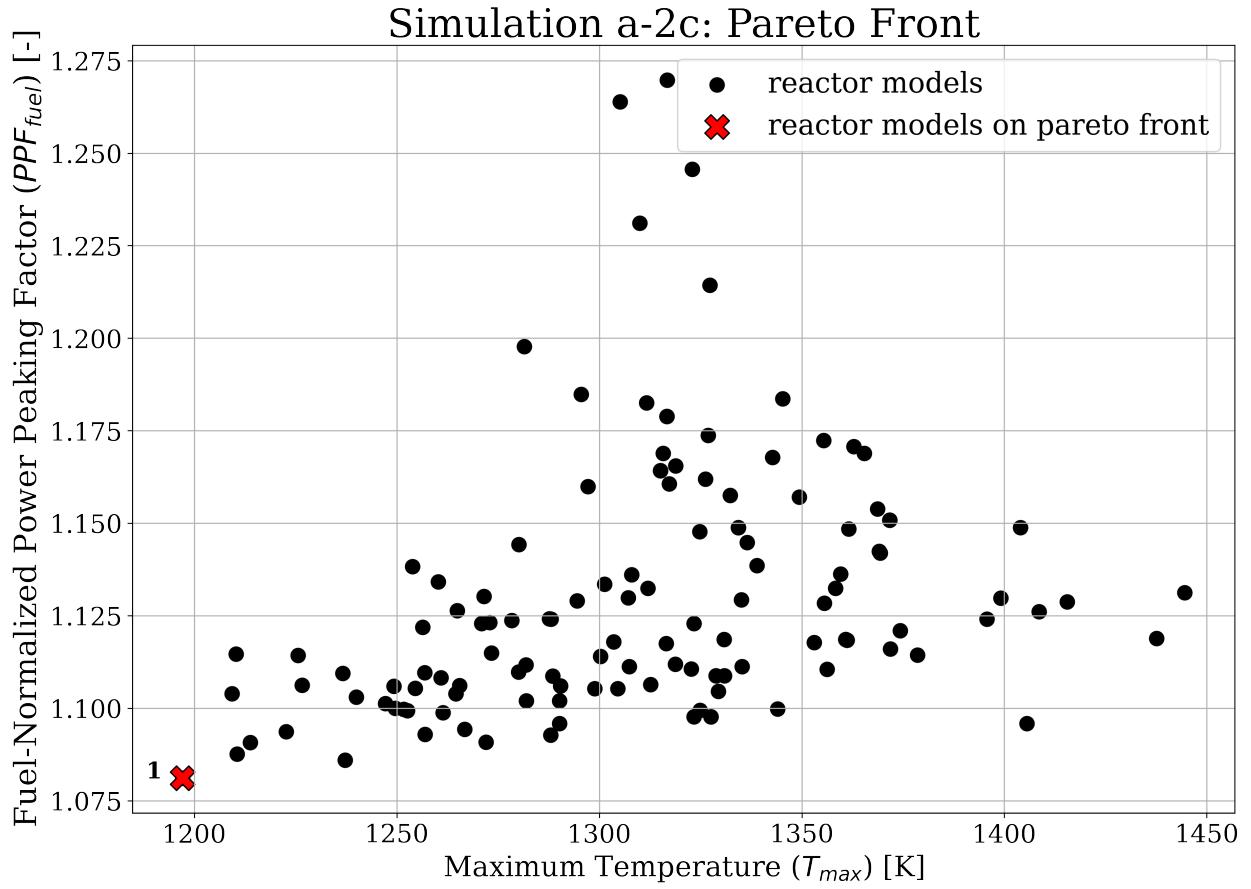
Table 7.16 shows the hypervolume value at each generation, confirming that simulation a-2c converges by generation 2.

Table 7.16: Simulation a-2c hypervolume values at each generation.

Two Objectives: Simulation a-2c	
Reference point: (1700, 1.5)	
Generation	Hypervolume [-]
1	210.685
2	210.685

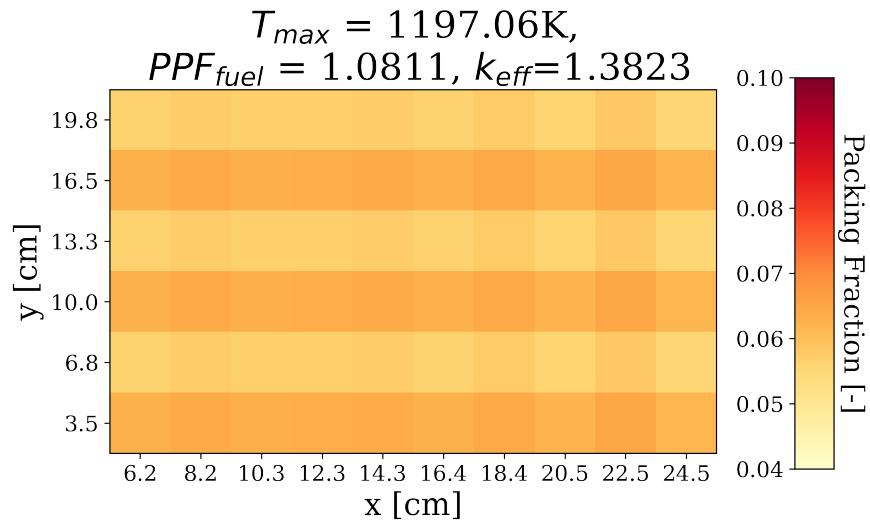
Figure 7.14a shows a plot of the final generation's reactor models' T_{max} against PPF_{fuel} ; crosses mark the optimal reactor model. Figures 7.14b and 7.14c show the optimal reactor model's TRISO packing fraction distribution and geometry.

Figure 7.14a shows that minimize T_{max} and minimize PPF_{fuel} are non-competing objectives, resulting in a single optimal reactor model. Figure 7.14b shows the TRISO distribution that best minimizes both T_{max} and PPF_{fuel} . The reactor model has a TRISO distribution that oscillates along the y-axis and slightly on the x-axis, and a packing fraction standard deviation of 0.033 across the one-third assembly. Along the y-axis, the distribution peaks at the odd rows (at 3.5cm, 10.0cm, and 16.5cm) with $PF \approx 0.06$ and has minimum points at the even rows (at 6.8cm, 13.3cm,

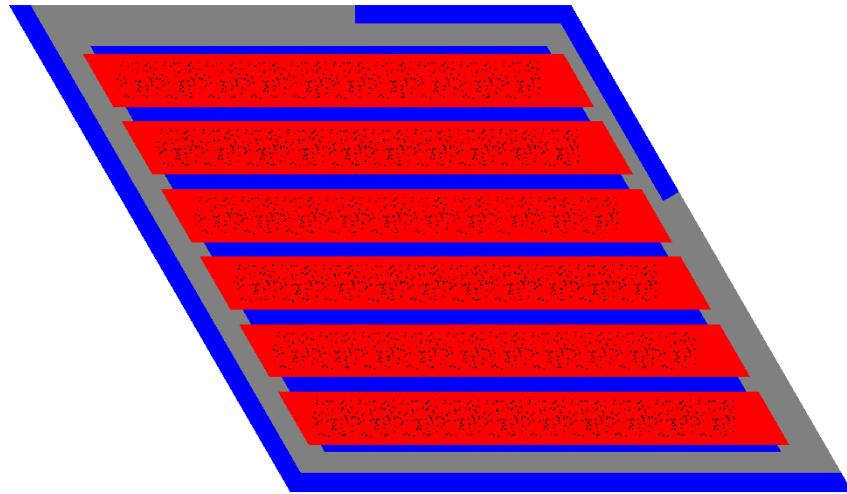


(a) Plot of final generation's reactor models' T_{max} against PPF_{fuel} . Crosses indicate the reactor models on the Pareto front. Annotated numbers on each cross correspond to TRISO distributions in the plot below.

Figure 7.14: Simulation a-2c – ROLLO double-objective optimization to minimize one-third assembly's maximum temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}) in the AHTR one-third assembly. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



(b) TRISO distribution for the 1 reactor model on the Pareto front.



(c) AHTR one-third assembly model with the most-minimized T_{max} and PPF_{fuel} (corresponds to the TRISO distribution in the above plot).

Figure 7.14: (contd.) Simulation a-2c – ROLLO double-objective optimization to minimize one-third assembly's maximum temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}) in the AHTR one-third assembly. Input parameters varied: TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

and 19.8cm) with $PF \approx 0.055$. Section 7.3.4 discusses and explains simulation a-2c's results.

7.3.4 Double-Objective Optimization Discussion

In this section, I explain how the driving factors and phenomena observed in the previous single-objective discussion (Section 7.2.4) combine to result in the optimal reactor models found by the double-objective optimization simulations.

Simulation a-2a

In Section 7.3.1's simulation a-2a, I conducted a two-objective optimization simulation to minimize total fuel packing fraction (PF_{total}) and maximum temperature (T_{max}) in a one-third assembly model by varying PF_{total} and TRISO distribution. In simulation a-2a, ROLLO found 13 reactor models on the Pareto Front (Figure 7.12a).

In simulation a-2a, ROLLO found that the one-third assembly models with the most-minimized PF_{total} objective are reactor models 3 and 4 (Figure 7.12b). Both reactor models have an oscillating TRISO distribution along the x-axis and y-axis. Figure 7.15 compares simulation a-2a's most-minimized PF_{total} reactor model 3 and simulation a-1a's most-minimized PF_{total} reactor model. Figure 7.15 shows that simulation a-2a's most-minimized PF_{total} reactor model, and simulation a-1a's most-minimized PF_{total} reactor model have similar distributions with peaks on the even fuel cell columns but at different amplitudes.

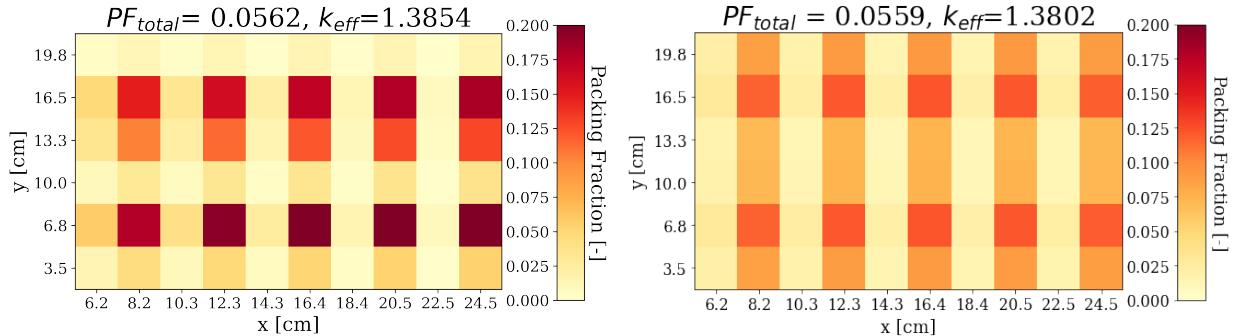


Figure 7.15: Simulation a-2a's most-minimized PF_{total} TRISO distribution from Figure 7.12 (left) and simulation a-1a's most-minimized PF_{total} TRISO distribution from Figure 7.1 (right).

In simulation a-2a, ROLLO found that the one-third assembly model with the most-minimized T_{max} objective, reactor model 9 (Figure 7.12b), has an almost constant TRISO distribution. Figure

7.16 compares simulation a-2a's most-minimized T_{max} reactor model 9 and simulation a-1b's most-minimized T_{max} reactor model. Figure 7.16 shows that simulation a-2a's most-minimized T_{max} reactor model, and simulation a-1b's most-minimized T_{max} reactor model have similar, PF_{total} values and almost constant TRISO distributions. Simulation a-2a's most-minimized T_{max} 's TRISO distribution is not as flat as simulation a-1b with packing fraction standard deviations of 0.004 and 0.0009, respectively.

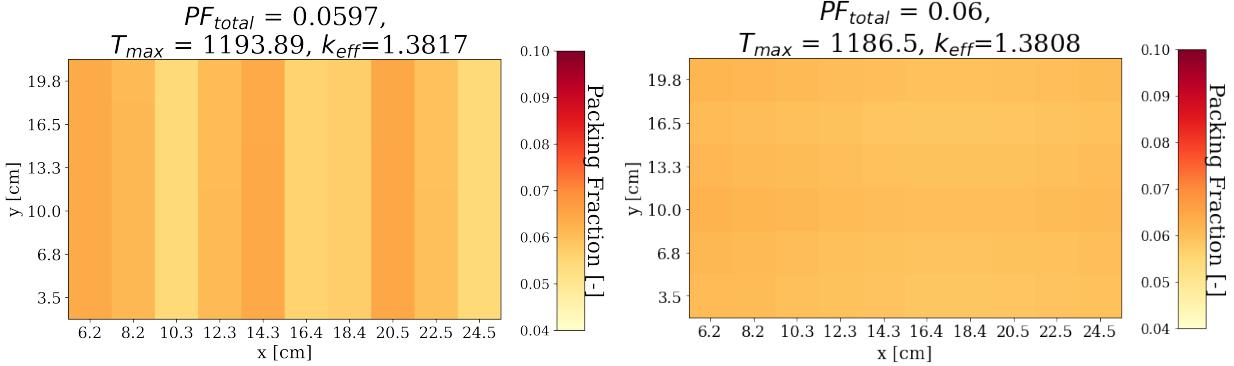


Figure 7.16: Simulation a-2a's most-minimized T_{max} TRISO distribution from Figure 7.12 (left) and simulation a-1b's most-minimized T_{max} TRISO distribution from Figure 7.4 (right).

Figure 7.17 shows reactor model 13, which minimized PF_{total} and T_{max} to an equal extent by balancing influences from both objectives.

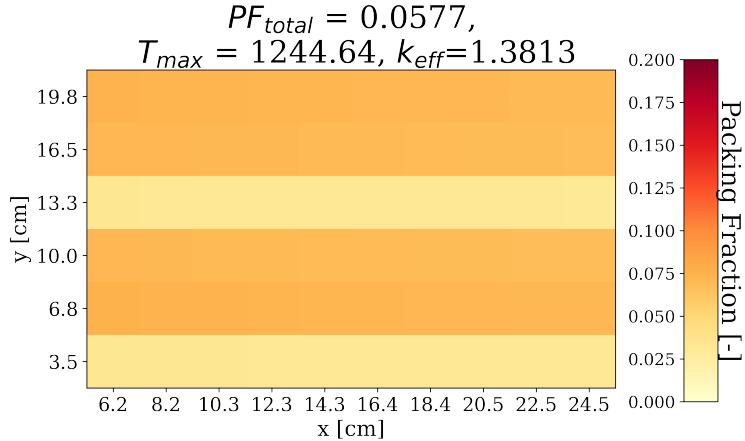


Figure 7.17: Simulation a-2a's reactor model 13 which minimized both PF_{total} and T_{max} to an equal extent (see Pareto Front in Figure 7.12).

The TRISO distributions on simulation a-2a's Pareto front in Figure 7.12 minimize both PF_{total} and T_{max} and vary between the two extreme cases: most-minimized PF_{total} and most-

minimized T_{max} . The minimize T_{max} objective influences the TRISO distribution's flatness, while the minimize PF_{total} objective influences the oscillating pattern. Section 7.2.4 described both influences in detail.

Simulation a-2b

In Section 7.3.2's simulation a-2b, I conducted a two-objective optimization simulation to minimize total fuel packing fraction (PF_{total}) and fuel-normalized power peaking factor (PPF_{fuel}) in a one-third assembly model by varying PF_{total} and TRISO packing fraction distribution. In simulation a-2b's final generation, ROLLO found 12 reactor models on the Pareto Front (Figure 7.13a).

In simulation a-2b, ROLLO found that the one-third assembly model with the most-minimized PF_{total} objective, reactor model 3 (Figure 7.13b), has an oscillating TRISO distribution along the x-axis and y-axis. Figure 7.18 compares simulation a-2b's reactor model 3 and simulation a-1a's most-minimized PF_{total} reactor model.

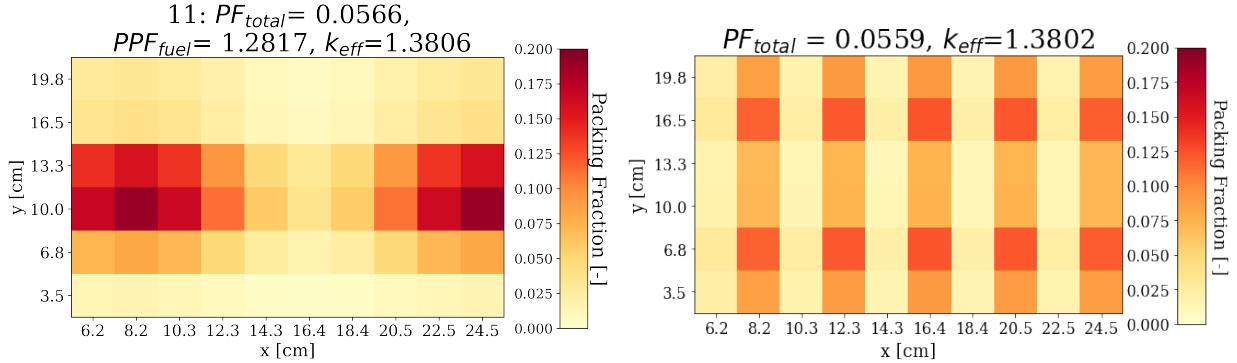


Figure 7.18: Simulation a-2b's most-minimized PF_{total} TRISO distribution from Figure 7.13 (left) and simulation a-1a's most-minimized PF_{total} TRISO distribution from Figure 7.1 (right).

Figure 7.18 shows that both models have similarly large packing fraction standard deviation of 0.053 and 0.04, respectively. However, they do not follow the same TRISO distribution pattern. Section 7.2.4 described that in the AHTR plank model, the minimize PF_{total} and minimize PPF_{fuel} objectives influence each other resulting in unexpected TRISO distributions at different PF_{total} values. This same effect also applies to the one-third assembly model, explaining why, unlike simulation a-2a, simulation a-2b's extreme most-minimized PF_{total} and most-minimized PPF_{fuel} do not follow similar TRISO distribution patterns as their single-objective counterparts.

In simulation a-2b, ROLLO found that the one-third assembly model with the most-minimized PPF_{fuel} objective, reactor model 1 (Figure 7.13b), has a TRISO distribution that oscillates along the y-axis and oscillates slightly along the x-axis. Figure 7.18 compares simulation a-2b's most-minimized PPF_{fuel} reactor model 1 and simulation a-1c's most-minimized PPF_{fuel} reactor model.

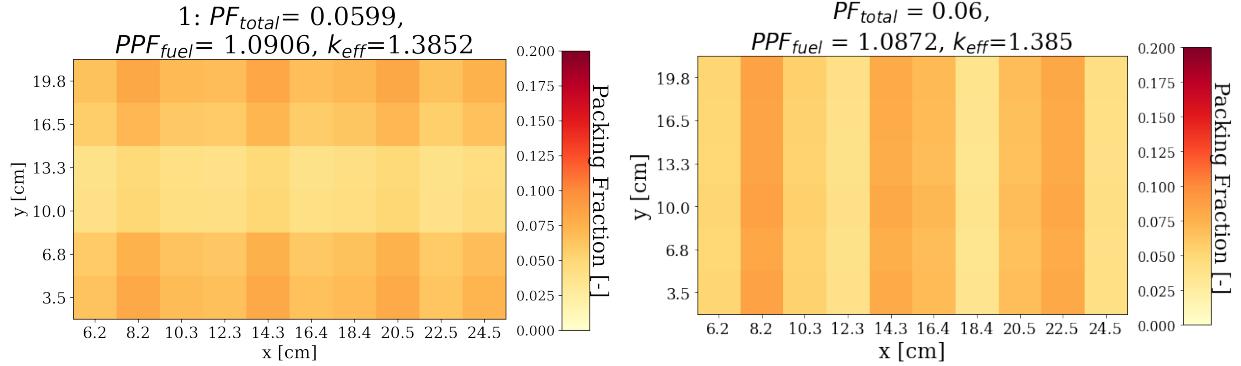


Figure 7.19: Simulation a-2b's most-minimized PPF_{fuel} TRISO distribution from Figure 7.13 (left) and simulation a-1c's most-minimized PPF_{fuel} TRISO distribution from Figure 7.6 (right).

Figure 7.19 shows that simulation a-2b's reactor model 1 and simulation a-1a's most-minimized PPF_{fuel} reactor model have similarly small packing fraction standard deviation of 0.013 and 0.017, respectively. However, they do not follow the same TRISO distribution pattern; this is attributed to the PF_{total} and PPF_{fuel} relationship resulting in unexpected TRISO distributions at different PF_{total} values, as mentioned previously. The relationship between the AHTR's PF_{total} and PPF_{fuel} merits future work of further sensitivity analysis. To better understand the reactor models on simulation a-2b's Pareto Front, I conduct an in-depth examination into the driving factors for the minimize PF_{total} and minimize PPF_{fuel} objectives.

Driving Factor Quantification Section 7.2.4 verified that, similar to the AHTR plank, the AHTR one-third assembly's minimize PF_{total} objective is driven by maximizing total fission reaction rates, and the minimize PPF_{fuel} objective is driven by flattening the thermal flux distribution. This section compares the total fission reaction rate and thermal flux flatness for 3 reactors models on simulation a-2b's Pareto Front (Figure 7.13a): reactor model 11 with most-minimized PF_{total} , reactor model 1 with most-minimized PPF_{fuel} , and reactor model 5 which minimizes both PF_{total} and PPF_{fuel} to an equal extent. Figure 7.20 shows the TRISO packing fraction distribution for

the 3 reactors models.

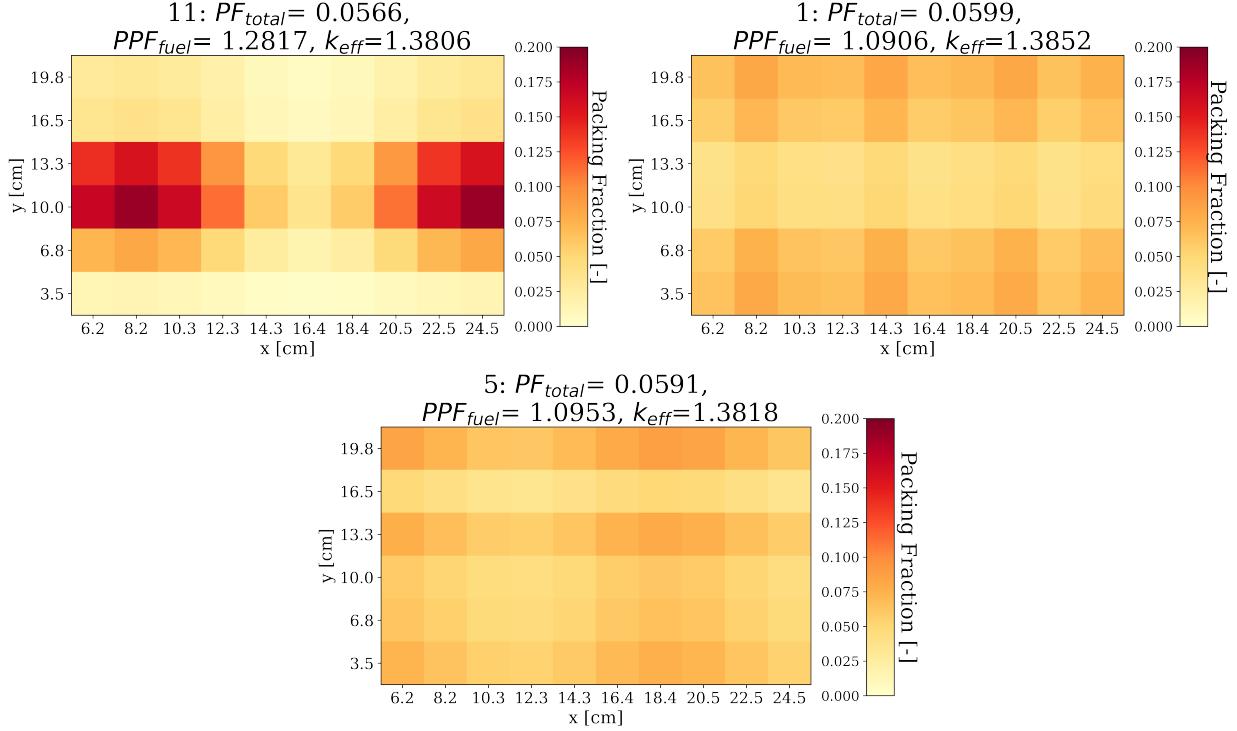


Figure 7.20: TRISO distributions for 3 reactor models on Simulation a-2b's Pareto Front (Figure 7.13a): reactor model 11 with most-minimized PF_{total} (top left), reactor model 1 with most-minimized PPF_{fuel} (top right), and reactor model 5 which minimizes both PF_{total} and PPF_{fuel} to an equal extent (bottom).

Table 7.17 shows the total fission reaction rate, and thermal flux flatness for the three reactor models.

Table 7.17: Total fission reaction rate, and thermal flux flatness ($\max(\phi_4)/\min(\phi_4)$) for 3 reactor models on simulation a-2b's Pareto Front (Figure 7.13a): reactor model 1 with most-minimized PPF_{fuel} , reactor model 11 with most-minimized PF_{total} , and reactor model 5 which minimizes both PF_{total} and PPF_{fuel} to an equal extent.

Most Minimized Parameter	Reactor Model	Fission [reactions/src]	% Diff	$\max(\phi_4)/\min(\phi_4)$	% Diff
Both	5	0.5471	-	1.2986	-
PF_{total}	11	0.5472	+0.017	1.3168	+1.40
PPF_{fuel}	1	0.5478	+0.12	1.2851	-1.03

The most minimized PPF_{fuel} reactor model 1 has the highest fission reaction rate, followed by the most minimized PF_{total} reactor model 11, and then reactor model 5 which minimizes both PF_{total} and PPF_{fuel} to an equal extent. Reactor model 1 has the highest fission reaction rate since

it has the highest PF_{total} . Reactor model 11 has a slightly higher fission reaction rate than reactor model 5, and they have k_{eff} values within error of each other despite reactor model 11 having a lower PF_{total} ($PF_{total,11} = 0.0566$ vs. $PF_{total,5} = 0.0591$). Section 7.2.4 verified that maximizing the total fission reaction rate drives the AHTR one-third assembly model's minimize PF_{total} objective. Therefore, reactor model 11's oscillating TRISO distribution enables a lower PF_{total} for the same k_{eff} as reactor model 5 since both have comparable total fission reaction rates.

The most minimized PPF_{fuel} reactor model 1 has the flattest thermal flux, followed by reactor model 5 which minimizes both PF_{total} and PPF_{fuel} to an equal extent, and then the most minimized PF_{fuel} reactor model 11. Section 7.2.4 verified that the AHTR one-third assembly model's minimize PPF_{fuel} objective is driven by flattening thermal (Group 4) flux distribution. Therefore, reactor model 1, with the flattest thermal flux distribution, most minimized PPF_{fuel} .

Simulation a-2c

In Section 7.3.3's simulation a-2c, I conducted a two-objective optimization simulation to minimize maximum temperature (T_{max}) and fuel-normalized power peaking factor (PPF_{fuel}) in a one-third assembly model by varying TRISO distribution. In simulation a-2c, ROLLO found one reactor model on the Pareto Front (Figure 7.14a), demonstrating that the minimize T_{max} and minimize PPF_{fuel} objectives are non-contrasting for the one-third assembly model.

Figure 7.21 compares the single reactor model on simulation a-2c's Pareto Front, simulation a-1b's most-minimized T_{max} reactor model, and simulation a-1c's most-minimized PPF_{fuel} reactor model. All reactor models have $PF_{total} = 0.06$. Figure 7.21 shows that the single reactor model on simulation a-2c's Pareto Front's TRISO distributions is more similar to simulation a-1b's most-minimized T_{max} TRISO distribution than simulation a-1c's most-minimized PPF_{fuel} TRISO distribution. Simulation a-1c's most-minimized PPF_{fuel} reactor model has a high $T_{max} = 1253.30$ K, while simulation a-1b's most-minimized T_{max} reactor model has a low $PPF_{fuel} = 1.104$. Therefore, influences from the minimize T_{max} objective results in the single reactor model on simulation a-2c's Pareto Front to have a TRISO distribution more similar to simulation a-1b's most-minimized T_{max} reactor model. The minimize T_{max} objective influences the TRISO distribution's flatness, while the minimize PPF_{fuel} objective influences the oscillating pattern. Section 7.2.4 described both

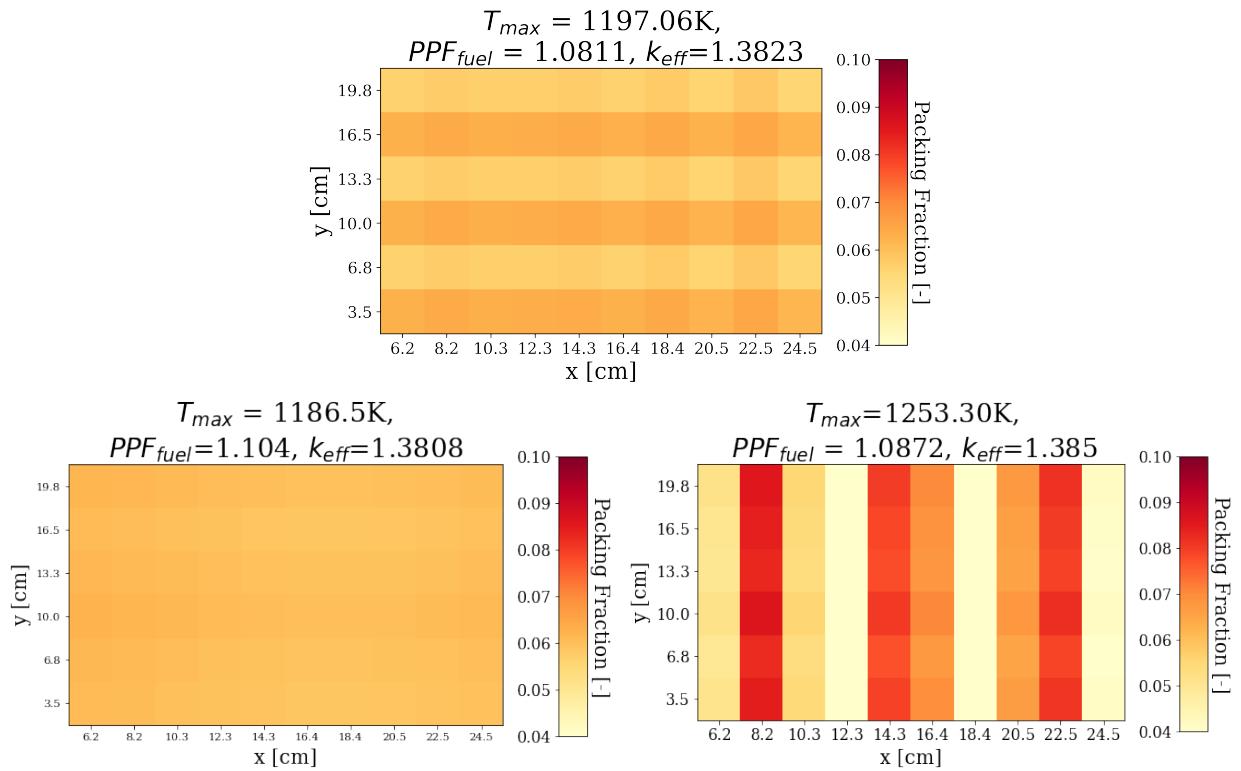


Figure 7.21: Simulation a-2c's single Pareto Front reactor model's TRISO distribution from Figure 7.14 (above), simulation a-1b's most-minimized T_{max} TRISO distribution from Figure 7.4 (lower left), and simulation a-1c's most-minimized PPF_{fuel} TRISO distribution from Figure 7.6 (lower right). All reactor models have $PF_{total} = 0.06$.

influences in detail.

7.3.5 Double-Objective Optimization Major Takeaways

The driving factors and influences from each objective come together in the a-2a and a-2b double-objective optimization simulations resulting in widely spread out optimal reactor models on their Pareto fronts. Simulation a-2c demonstrated that at $PF_{total} = 0.06$, minimize T_{max} and minimize PPF_{fuel} are non-competing objectives, resulting in a single optimal reactor model. In the double-objective optimization simulations, the minimize T_{max} objective continued to influence the flattening of the TRISO distribution. The results from simulation a-2b suggests that the minimize PF_{total} objective's driving factor maximize total fission reaction rate and minimize PPF_{fuel} objective's driving factor flattening thermal flux distribution influence each other resulting in unexpected TRISO distributions at different PF_{total} values.

7.4 AHTR One-Third Assembly: Triple-Objective Optimization Results

This section reports the AHTR one-third assembly's ROLLO triple-objective optimization results. I run two triple-objective optimization simulations that optimize all the objectives (minimize PF_{total} , T_{max} , and PPF_{fuel}). The first simulation varies total fuel packing fraction (PF_{total}) and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$). The second simulation varies PF_{total} , $\rho_{TRISO}(\vec{r})$, and coolant channel shape. Earlier in this chapter, Table 7.2 summarized the triple-objective simulations in this section: a-3a and a-3b. The following two subsections describing the optimization results are distinguished by the input parameters varied, since both simulations optimize for all three objectives.

7.4.1 a-3a: Variation of PF_{total} and $\rho_{TRISO}(\vec{r})$

This section reports results from the triple-objective optimization simulation a-3a, with all objectives minimized: total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. The input parameters

varied are PF_{total} and TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$). Table 7.18 summarizes simulation a-3a's optimization problem parameters.

Table 7.18: Simulation a-3a optimization problem parameters.

Three Objectives: Simulation a-3a	
Objectives	Minimize PF_{total} Minimize T_{max} Minimize PPF_{fuel}
Input parameter variations	$0.05 \leq PF_{total} \leq 0.07$ $\rho_{TRISO}(\vec{r}): 0 \leq a \leq 2, 0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r}): 0 \leq b \leq \frac{\pi}{2}, 0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 \leq c \leq 2\pi, 0 \leq f \leq 2\pi$
Constraints	$k_{eff} \geq 1.38$
Genetic algorithm parameters	Population size: 128 Generations: 5

Table 7.19 shows the hypervolume value at each generation, confirming that simulation a-3a converges by generation 5.

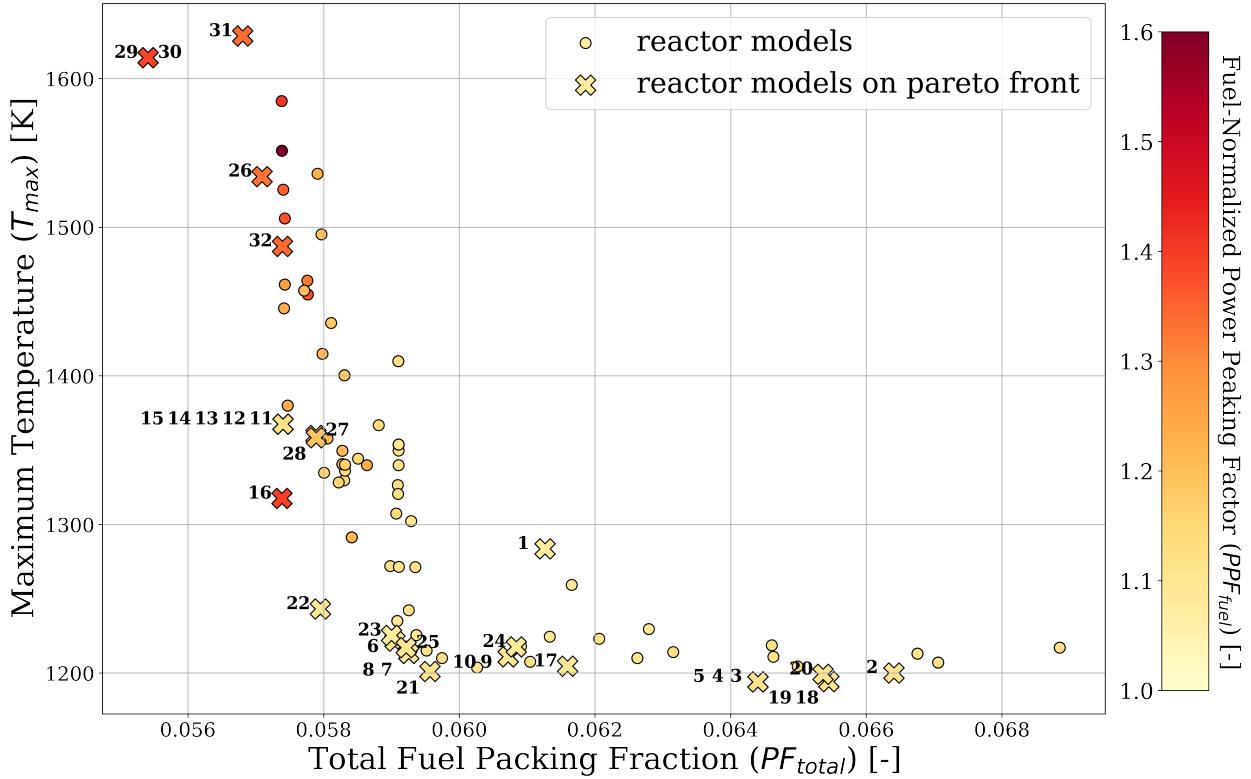
Table 7.19: Simulation a-3a hypervolume values at each generation.

Three Objectives: Simulation a-3a	
Reference point: (0.07, 1700, 1.8)	
Generation	Hypervolume [-]
1	4.0925
2	4.2233
3	4.4002
4	4.4250
5	4.5312

Figure 7.22a shows a plot of the final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} ; crosses mark the reactor models that fall on the Pareto front. Figure 7.22b shows the 32 TRISO packing fraction distributions in the final generation, labeled numerically, that fall on the Pareto front.

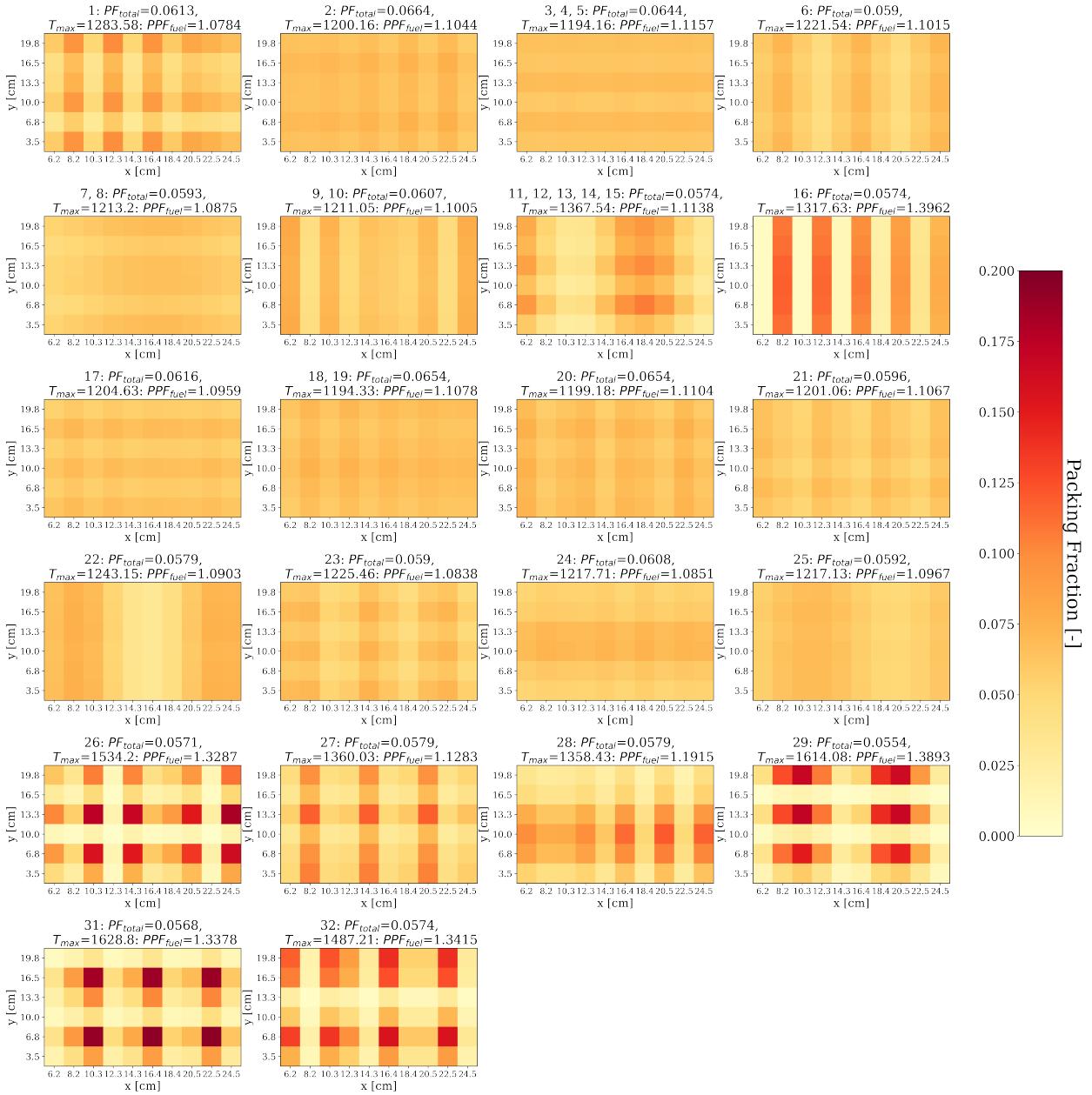
Figure 7.22 demonstrates that ROLLO successfully found 32 reactor models on simulation a-3a final generation's Pareto front. Figure 7.23 shows three reactor models on the Pareto front that most minimized each objective and one reactor model on the Pareto front that equally minimized all three objectives. I selected the equally minimized reactor model based on visual inspection of Figure 7.22 and selecting a reactor model close to the origin with a light yellow color dimension. Reactor model 30 most-minimized PF_{total} , reactor model 3 most-minimized T_{max} , reactor model 1

Simulation a-3a: Pareto Front



(a) Plot of final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} as a color dimension. Crosses indicate the reactor models on the Pareto front. Cross numbering corresponds to TRISO distributions in Figure 7.22b.

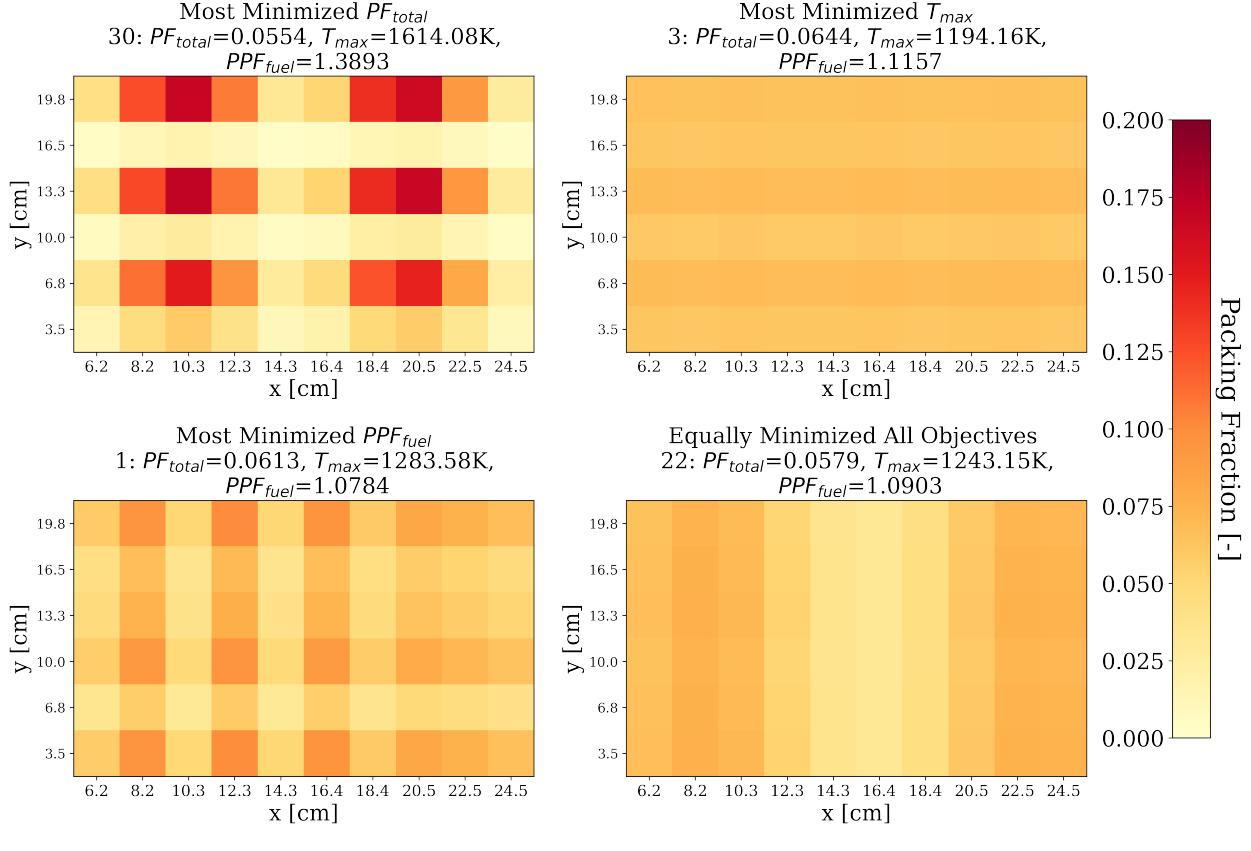
Figure 7.22: Simulation a-3a – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).



(b) TRISO distributions for the 32 reactor models on the Pareto front. Numbered reactor models correspond to numbered crosses in Figure 7.22a. Note that some models have identical distributions, resulting in the 22 plots in this subfigure.

Figure 7.22: (contd.) Simulation a-3a – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

most-minimized PPF_{fuel} , and reactor model 22 equally minimized all three objectives.

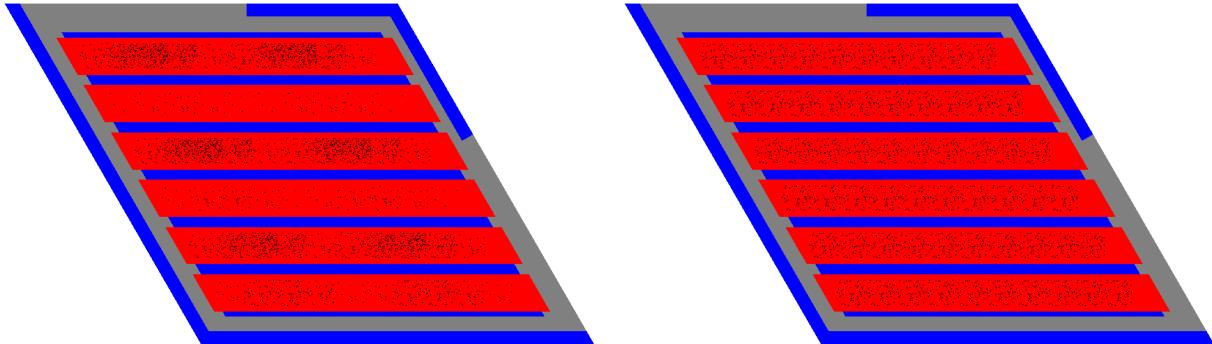


(a) TRISO packing fraction distributions.

Figure 7.23: AHTR one-third assembly models and TRISO distributions for the three reactor models on simulation a-3a's Pareto front that most minimized each objective, and one reactor model that equally minimized all three objectives. Simulation a-3a – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}) and normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

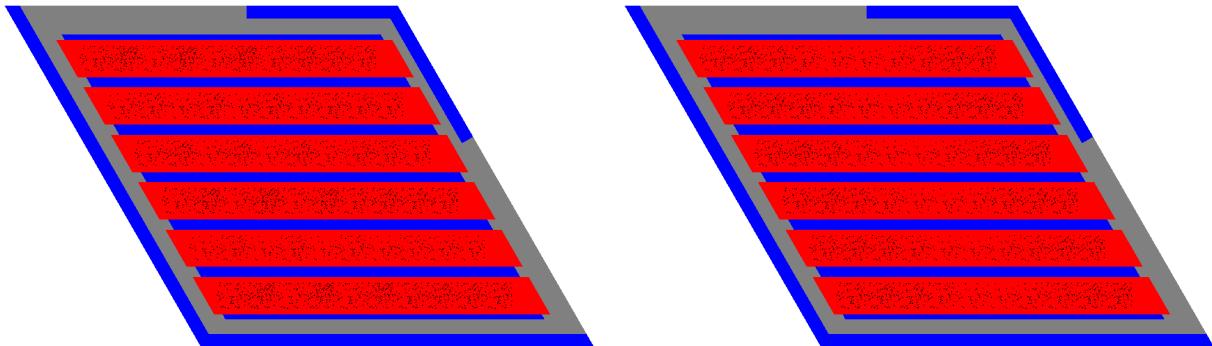
In the top left of Figure 7.23a, the one-third assembly model with the most-minimized PF_{total} is reactor model 30 (the corresponding geometry is illustrated in Figure 7.23b). Reactor model 30 has an oscillating TRISO distribution along the x-axis and y-axis, and a packing fraction standard deviation of 0.052 across the one-third assembly. Along the y-axis, the distribution peaks at the even fuel cell rows (at 6.8cm, 13.3cm, and 19.8cm) and has minimum points at the odd fuel cell rows (at 3.5cm, 10.0cm, and 16.5cm). The even fuel cell rows have a ~ 0.14 x-axis variation with peaks of $PF \approx 0.16$. The odd fuel cell rows have a ~ 0.02 x-axis variation with minimums of $PF \approx 0.005$.

In the top right of Figure 7.23a, the one-third assembly model with the most-minimized T_{max}



(b) AHTR one-third assembly model with the most-minimized PF_{total} (reactor model 30).

(c) AHTR one-third assembly model with the most-minimized T_{max} (reactor model 3).



(d) AHTR one-third assembly model with the most-minimized PPF_{fuel} (reactor model 1).

(e) AHTR one-third assembly model that equally minimized all objectives (reactor model 22).

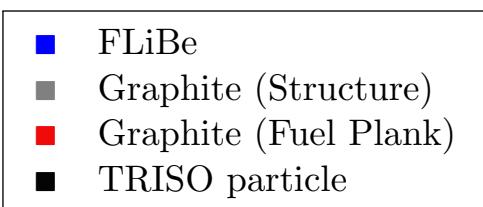


Figure 7.23: (contd.) AHTR one-third assembly models and TRISO distributions for the three reactor models on simulation a-3a's Pareto front that most-minimized each objective, and one reactor model that equally minimized all three objectives. Simulation a-3a – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}) and normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$).

is reactor model 3 (the corresponding geometry is illustrated in Figure 7.23c). Reactor model 3 has an almost constant TRISO packing fraction distribution with a packing fraction standard deviation of 0.003 across the one-third assembly.

In Figure 7.23a's bottom left, the one-third assembly model with the most-minimized PPF_{fuel} is reactor model 1 (the corresponding geometry is illustrated in Figure 7.23d). Reactor model 1 has an oscillating TRISO distribution along the x-axis and y-axis and a packing fraction standard deviation of 0.019 across the one-third assembly. Along the x-axis, the distribution peaks at the 2nd, 4th, and 6th fuel cell columns (at 8.2cm, 12.3cm, and 16.4cm). These three fuel cell columns have a ~ 0.04 y-axis variation with peaks of $PF \approx 0.1$. Along the y-axis, the distribution peaks at the 1st, 3rd, and 6th fuel cell rows (at 3.5cm, 10.0cm, and 19.8cm). These three fuel cell rows have a ~ 0.05 x-axis variation with peaks of $PF \approx 0.1$.

In the bottom right of Figure 7.23a, the one-third assembly model that equally minimized all three objectives is reactor model 22 (the corresponding geometry is illustrated in Figure 7.23e). Reactor model 22 has an oscillating TRISO distribution along x-axis and a packing fraction standard deviation of 0.015 across the one-third assembly. Along the x-axis, the distribution peaks at the 2nd, 9th, and 10th fuel cell columns (at 8.2cm, 22.5cm, and 24.5cm) with $PF \approx 0.07$. The distribution has minimum points at the 5th and 6th fuel cell columns (at 14.3cm, 16.4cm) with $PF \approx 0.03$. Section 7.4.3 discusses and explains simulation a-3a's results.

7.4.2 a-3b: Variation of PF_{total} , $\rho_{TRISO}(\vec{r})$, and Coolant Channel Shape

This section reports results from the triple-objective optimization simulation a-3b, the largest and final optimization problem run for the AHTR one-third assembly model. Simulation a-3b minimized all the objectives: total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}), and varied all the input parameters: total fuel packing fraction (PF_{total}), TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), and coolant channel shape (r_1, r_2, r_3, r_4 , and r_5). Previous optimization simulations varied combinations of the objectives and input parameters, but not all together. Simulation a-3b ran for 1528 node-hours on the Theta supercomputer with 6 generations and 128 reactor models per generation, which is five times the compute time per single-objective simulation, and larger than the compute time

per two-objective simulation, as reported in Table 7.22. Table 7.20 summarizes simulation a-3b's optimization problem parameters.

Table 7.20: Simulation a-3b optimization problem parameters.

Three Objectives: Simulation a-3b	
Objectives	Minimize PF_{total} Minimize T_{max} Minimize PPF_{fuel}
Input parameter variations	$0.05 \leq PF_{total} \leq 0.07$ $\rho_{TRISO}(\vec{r}): 0 \leq a \leq 2, 0 \leq d \leq 2$ $\rho_{TRISO}(\vec{r}): 0 \leq b \leq \frac{\pi}{2}, 0 \leq e \leq \frac{\pi}{2}$ $\rho_{TRISO}(\vec{r}): 0 \leq c \leq 2\pi, 0 \leq f \leq 2\pi$ coolant channel shape: $0.1 < r_1 < 0.35$ coolant channel shape: $0.1 < r_2 < 0.35$ coolant channel shape: $0.1 < r_3 < 0.35$ coolant channel shape: $0.1 < r_4 < 0.35$ coolant channel shape: $0.1 < r_5 < 0.35$
Constraints	$k_{eff} \geq 1.38$
Genetic algorithm parameters	Population size: 128 Generations: 6

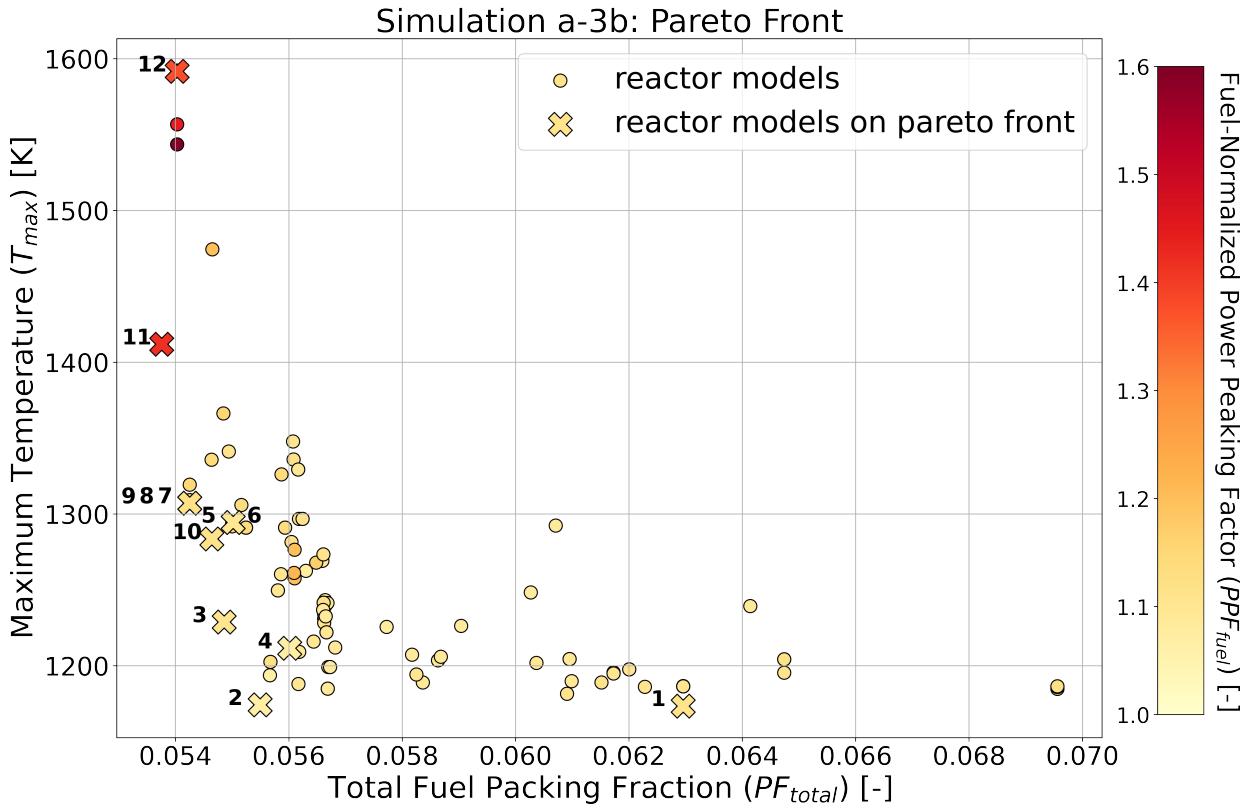
Table 7.21 shows the hypervolume value at each generation, confirming that simulation a-3b converges by generation 6.

Table 7.21: Simulation a-3b hypervolume values at each generation.

Three Objectives: Simulation a-3b	
Reference point: (0.07, 1700, 1.8)	
Generation	Hypervolume [-]
1	5.4961
2	5.6739
3	5.6876
4	5.8104
5	6.0023
6	6.0093

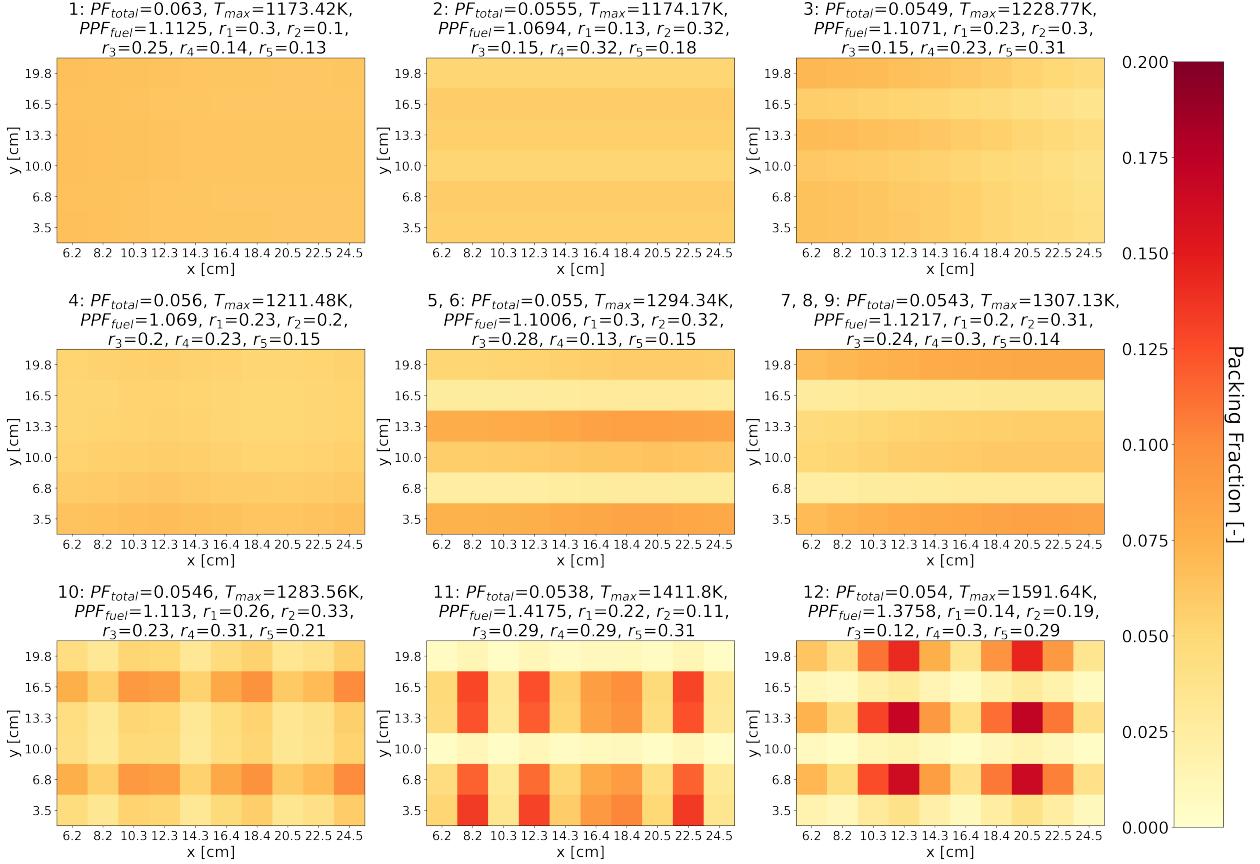
Figure 7.24a shows a plot of the final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} ; crosses mark the reactor models that fall on the Pareto front. Figure 7.24b shows the 12 TRISO packing fraction distributions in the final generation, labeled numerically, that fall on the Pareto front.

Figure 7.24 demonstrates that ROLLO found 12 reactor models on simulation a-3b final generation's Pareto front. Figure 7.25 shows three reactor models on the Pareto front that most



(a) Plot of final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} as a color dimension. Crosses indicate the reactor models on the Pareto front. Cross numbering correspond to TRISO distributions in Figure 7.24b.

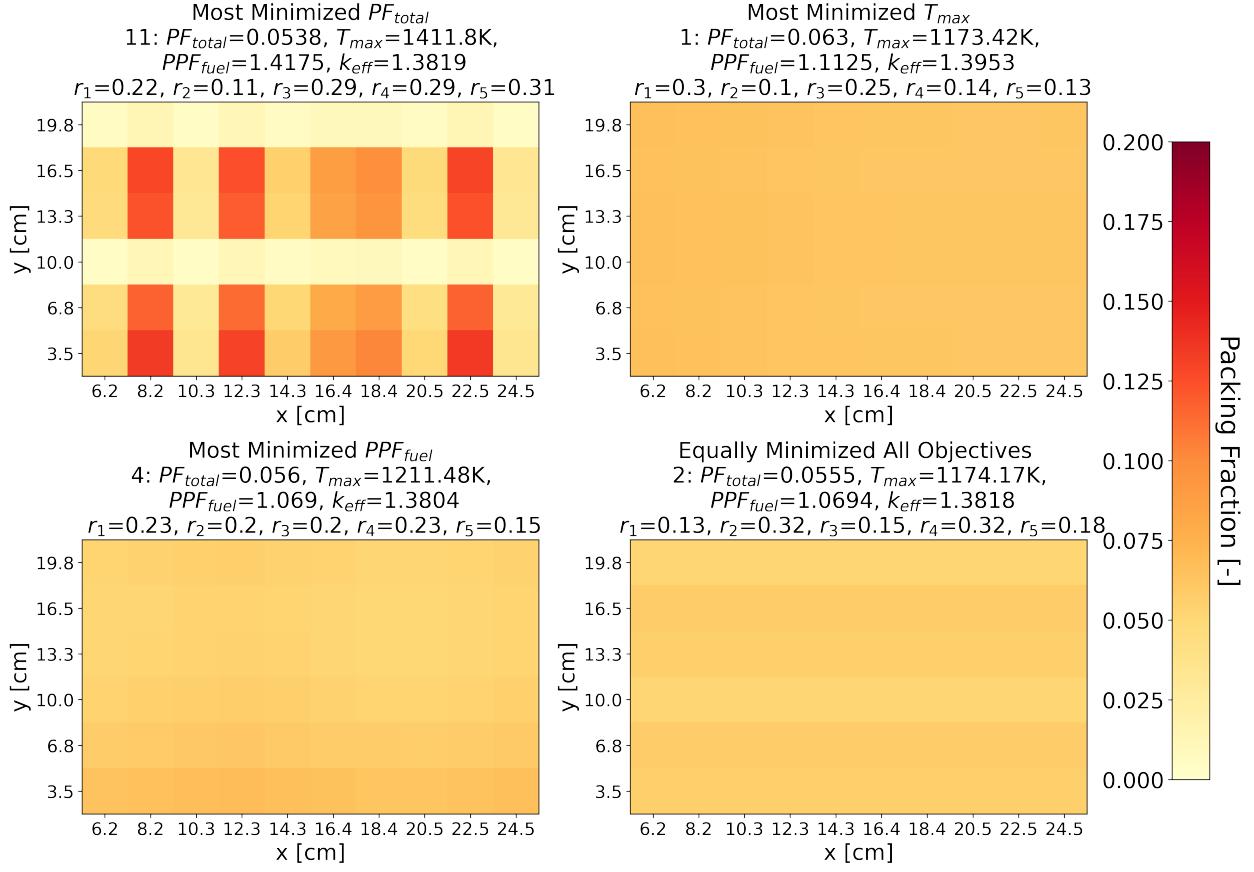
Figure 7.24: Simulation a-3b – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).



(b) TRISO distributions for the 12 reactor models on the Pareto front. Numbered reactor models correspond to numbered crosses in Figure 7.24a. Note that some models have identical distributions, resulting in the 9 plots in this subfigure.

Figure 7.24: (contd.) Simulation a-3b – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).

minimized each objective and one reactor model on the Pareto front that equally minimized all three objectives. I selected the equally minimized reactor model based on visual inspection of Figure 7.24 and selecting a reactor model close to the origin with a light yellow color dimension. Reactor model 11 most-minimized PF_{total} , reactor model 1 most-minimized T_{max} , reactor model 4 most-minimized PPF_{fuel} , and reactor model 2 equally minimized all three objectives.



(a) TRISO packing fraction distributions.

Figure 7.25: AHTR one-third assembly models and TRISO distributions for the 3 reactor models on simulation a-3b's Pareto front that most minimized each objective, and 1 reactor model that equally minimized all three objectives. Simulation a-3b – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).

In Figure 7.25a's top left, the one-third assembly model with the most-minimized PF_{total} is reactor model 11 (the corresponding geometry is illustrated in Figure 7.25b). Reactor model 11's TRISO distribution oscillates along the x-axis, slightly oscillates along the y-axis, and has

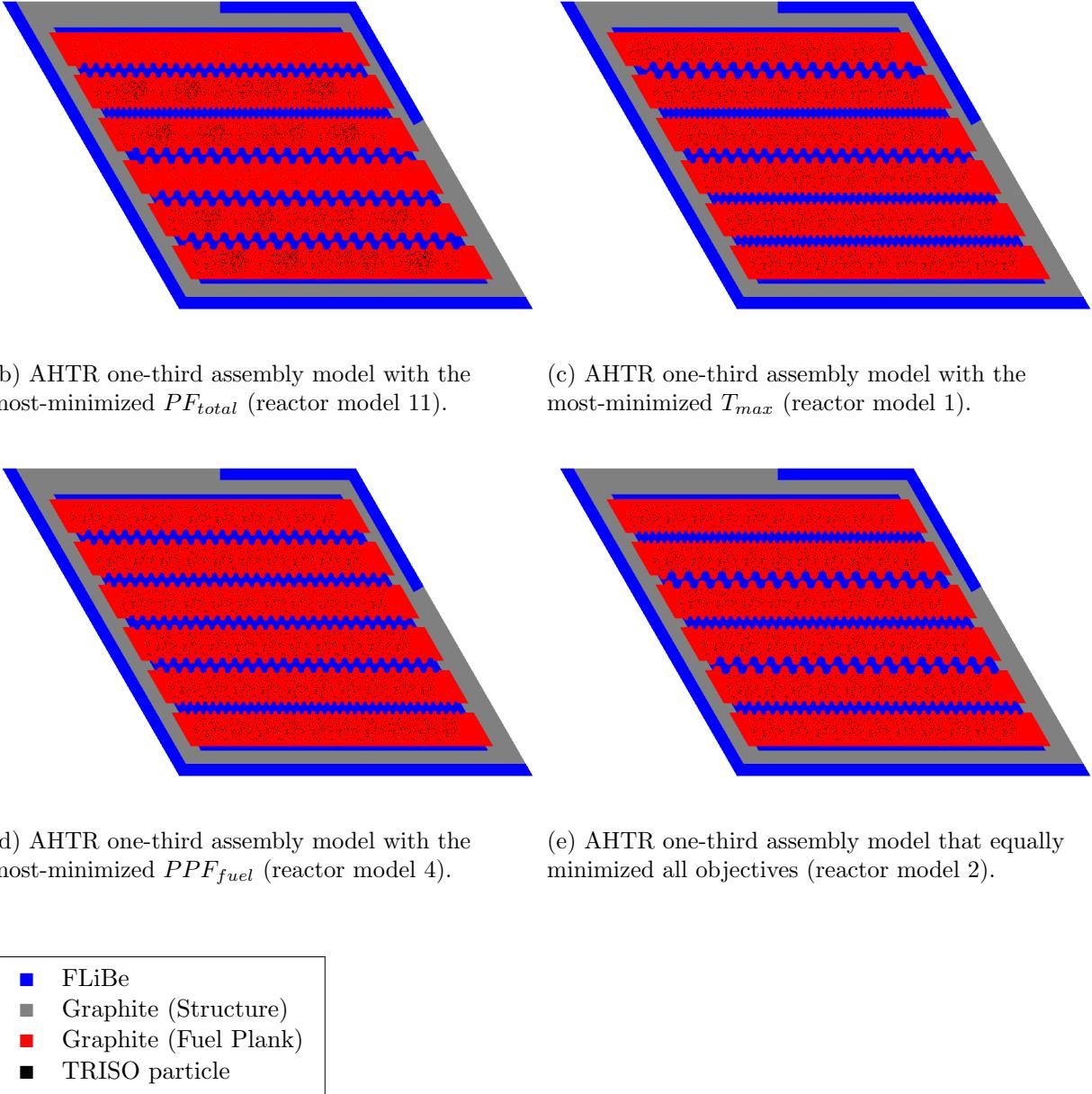


Figure 7.25: (contd.) AHTR one-third assembly models and TRISO distributions for the 3 reactor models on simulation a-3b's Pareto front that most-minimized each objective, and 1 reactor model that equally minimized all three objectives. Simulation a-3b – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).

a packing fraction standard deviation of 0.044 across the one-third assembly. Along the x-axis, the distribution peaks at the 2nd and 9th fuel cell columns (at 8.2cm, 12.3cm, and 22.5cm), and minimum points at the 1st and 8th fuel cell columns (at 10.3cm and 24.5cm). The 2nd and 9th columns have \sim 0.12 y-axis variation with peaks of $PF \approx 0.13$. The 1st and 8th columns have \sim 0.03 y-axis variation with minimums of $PF \approx 0.003$.

In the top right of Figure 7.25a, the one-third assembly model with the most-minimized T_{max} is reactor model 1 (the corresponding geometry is illustrated in Figure 7.25c). Reactor model 1 has an almost constant TRISO packing fraction distribution with a packing fraction standard deviation of 0.001 across the one-third assembly.

In Figure 7.25a's bottom right, the one-third assembly model with the most-minimized PPF_{fuel} is reactor model 4 (the corresponding geometry is illustrated in Figure 7.25d). Reactor model 4's TRISO distribution oscillates slightly along the y-axis, and has a packing fraction standard deviation of 0.005 across the one-third assembly. Along the y-axis, the distribution peaks at the 1st fuel cell row (at 3.5cm) with $PF \approx 0.065$. The distribution has minimums at the 4th, 5th, and 6th fuel cell rows (at 13.3cm, 16.5cm, and 19.8cm) with $PF \approx 0.052$.

In the bottom right of Figure 7.25a, the one-third assembly model that equally minimized all three objectives is reactor model 2 (the corresponding geometry is illustrated in Figure 7.25e). Reactor model 2's TRISO distribution oscillates slightly along the y-axis, and has a packing fraction standard deviation of 0.003 across the one-third assembly. Along the y-axis, the distribution peaks at the 2nd and 5th fuel cell row (at 6.8cm and 16.5cm) with $PF \approx 0.058$. The distribution has minimums at the 3rd and 6th fuel cell rows (at 10.0cm and 19.8cm) with $PF \approx 0.052$. Section 7.4.3 discusses and explains simulation a-3b's results.

7.4.3 Triple-Objective Optimization Discussion

ROLLO successfully found a wide spread of reactor models in each of the triple-objective optimization simulation's final generation Pareto fronts. In this section, I explain how the driving factors and phenomena observed in the previous single-objective discussion (Section 7.2.4) combine to result in the optimal reactor models found by the triple-objective optimization simulations.

Simulation a-3a

In Section 7.4.1's simulation a-3a, I conducted a triple-objective optimization simulation to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly model by varying PF_{total} and TRISO distribution in the x and y directions of the one-third assembly. ROLLO found 32 widely spread reactor models on simulation a-3a's Pareto front (Figure 7.22a).

In simulation a-3a, ROLLO found that the one-third assembly model with the most-minimized PF_{total} objective, reactor model 30 (Figure 7.23a), has an oscillating TRISO distribution along the x-axis and y-axis. Figure 7.26 compares simulation a-3a's reactor model 30 and simulation a-1a's most-minimized PF_{total} reactor model.

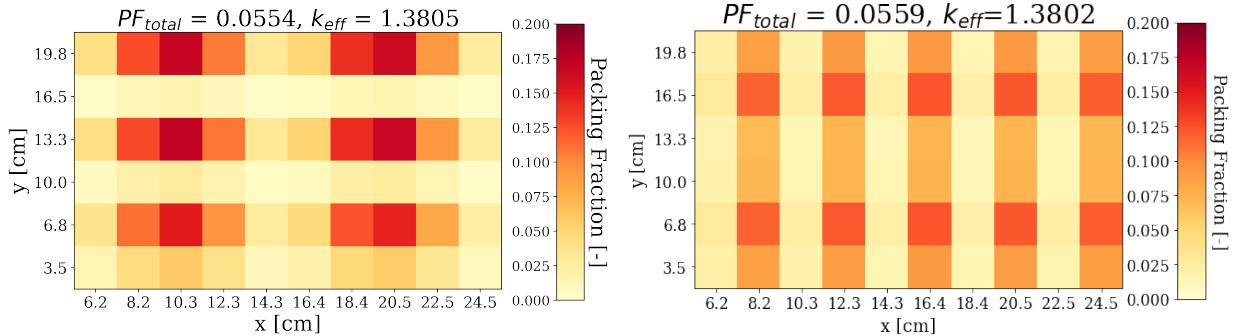


Figure 7.26: Simulation a-3a's most-minimized PF_{total} TRISO distribution from Figure 7.23 (left) and simulation a-1a's most-minimized PF_{total} TRISO distribution from Figure 7.1 (right).

Figure 7.26 shows that simulation a-3a's reactor model 30 and simulation a-1a's most-minimized PF_{total} reactor model have similarly large packing fraction standard deviation of 0.052 and 0.04, respectively. However, they do not follow the same TRISO distribution pattern; this is attributed to the PF_{total} and PPF_{fuel} relationship resulting in unexpected TRISO distributions at different PF_{total} values, as mentioned previously.

In simulation a-3a, ROLLO found that the one-third assembly model with the most-minimized T_{max} objective, reactor model 3 (Figure 7.23a), has an almost constant TRISO distribution. Figure 7.27 compares simulation a-3a's most-minimized T_{max} reactor model 3 and simulation a-1b's most-minimized T_{max} reactor model.

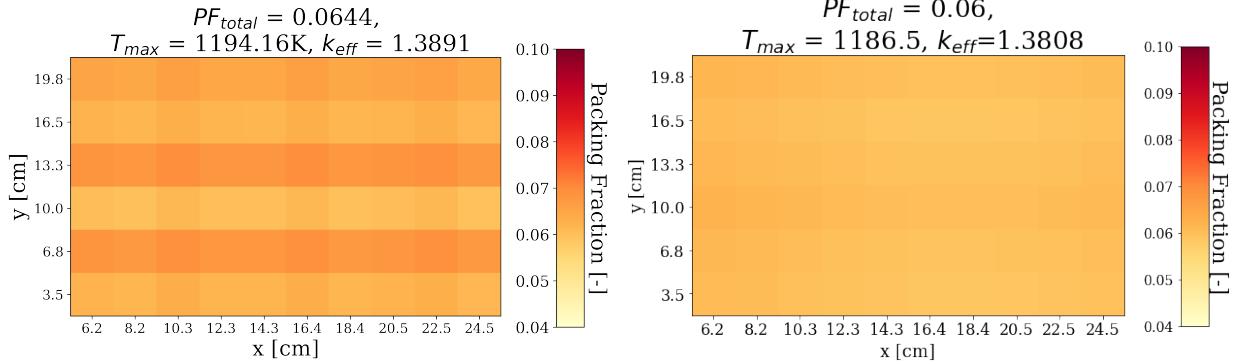


Figure 7.27: Simulation a-3a's most-minimized T_{max} TRISO distribution from Figure 7.23a (left) and simulation a-1b's most-minimized T_{max} TRISO distribution from Figure 7.4 (right).

Figure 7.27 shows that simulation a-3a's most-minimized T_{max} reactor model, and simulation a-1b's most-minimized T_{max} reactor model have similar, almost constant, TRISO distributions with packing fraction standard deviations of 0.003 and 0.0009, respectively. However, they have different PF_{total} values and simulation a-3a's most-minimized T_{max} 's TRISO distribution is not as flat as simulation a-1b.

In simulation a-3a, ROLLO found that the one-third assembly model with the most-minimized PPF_{fuel} objective, reactor model 1 (Figure 7.23a) has an oscillating TRISO distribution along the x-axis and y-axis. Figure 7.28 compares simulation a-3a's most-minimized PPF_{fuel} reactor model 1 and simulation a-1c's most-minimized PPF_{fuel} reactor model.

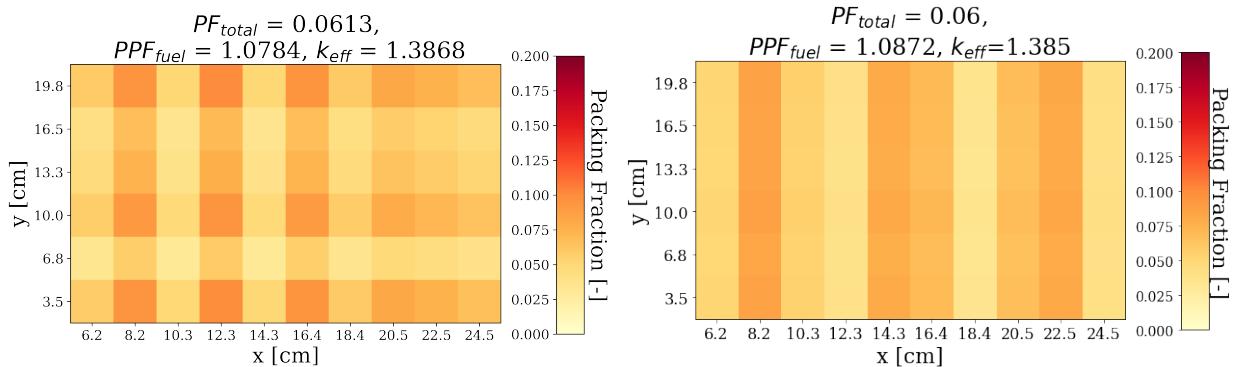


Figure 7.28: Simulation a-3a's most-minimized PPF_{fuel} TRISO distribution from Figure 7.23a (left) and simulation a-1c's most-minimized PPF_{fuel} TRISO distribution from Figure 7.6 (right).

Figure 7.28 shows that simulation a-3a's reactor model 1 and simulation a-1a's most-minimized PPF_{fuel} reactor model have similarly small packing fraction standard deviation of 0.019 and 0.017,

respectively. However, they do not follow the same TRISO distribution pattern; this is attributed to the aforementioned PF_{total} and PPF_{fuel} relationship resulting in unexpected TRISO distributions at different PF_{total} values.

Figure 7.29 shows reactor model 22, which minimized PF_{total} , T_{max} , and PPF_{fuel} to an equal extent by balancing influences from all objectives.

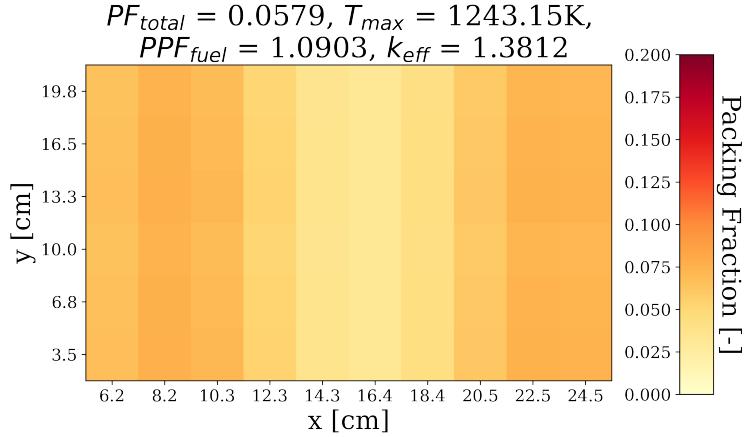


Figure 7.29: Simulation a-3a's reactor model 22 which minimized PF_{total} , T_{max} , and PPF_{fuel} to an equal extent (see Pareto Front in Figure 7.22a).

In all the reactor models on simulation a-3a's Pareto front (Figure 7.22a), the TRISO distribution flatness is influenced by the minimize T_{max} objective. The variations in TRISO distributions are influenced by both the minimize PF_{total} and minimize PPF_{fuel} objectives. However, as mentioned previously, the PF_{total} and PPF_{fuel} relationship result in unexpected TRISO distributions at different PF_{total} values. The minimize PF_{total} objective tries to maximize the fission reaction rate to enable a higher k_{eff} for a lower PF_{total} , and the PPF_{fuel} objective tries to flatten thermal flux.

Simulation a-3b

In Section 7.4.2's simulation a-3b, I conducted a triple-objective optimization simulation to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in a one-third assembly model by varying PF_{total} , TRISO distribution, and coolant channel shape (r_1, r_2, r_3, r_4, r_5). ROLLO found 12 reactor models on simulation a-3b's Pareto front (Figure 7.24a).

Compared to simulation a-3a in the previous section, simulation a-3b's reactor models have, on average, a lower T_{max} value due to coolant channel shape variation. In simulation a-3b, ROLLO found that the one-third assembly model with the most-minimized PF_{total} objective, reactor model 11 (Figure 7.25a), has an oscillating TRISO distribution along the x-axis and y-axis. Figure 7.26 compares simulation a-3b's reactor model 11 and simulation a-1a's most-minimized PF_{total} reactor model.

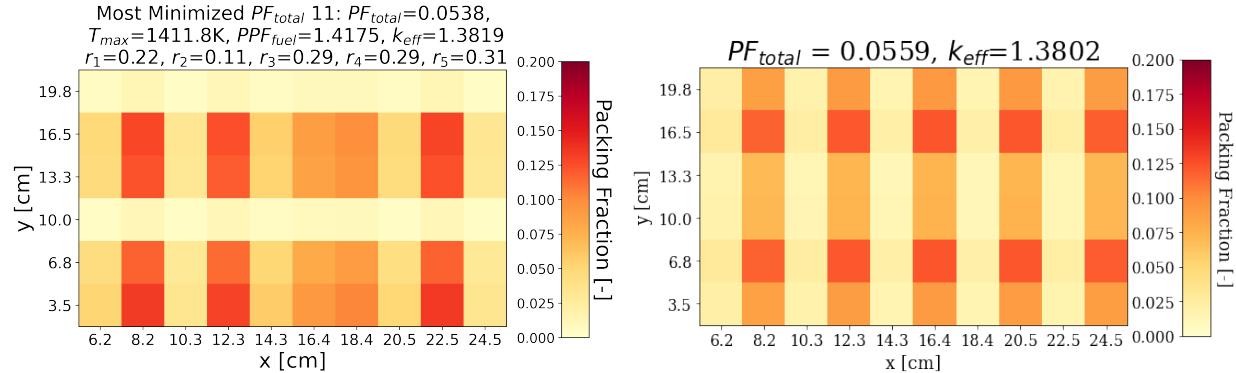


Figure 7.30: Simulation a-3b's most-minimized PF_{total} TRISO distribution from Figure 7.25 (left) and simulation a-1a's most-minimized PF_{total} TRISO distribution from Figure 7.1 (right).

Figure 7.30 shows that simulation a-3b's reactor model 11 and simulation a-1a's most-minimized PF_{total} reactor model have similarly large packing fraction standard deviation of 0.044 and 0.04, respectively. However, they do not follow the same TRISO distribution pattern. The PF_{total} and PPF_{fuel} relationship results in unexpected TRISO distributions at different PF_{total} values.

In simulation a-3b, ROLLO found that the one-third assembly model with the most-minimized T_{max} objective, reactor model 1 (Figure 7.23a), has an almost constant TRISO distribution. Figure 7.31 compares simulation a-3b's most-minimized T_{max} reactor model 1 and simulation a-1b's most-minimized T_{max} reactor model.

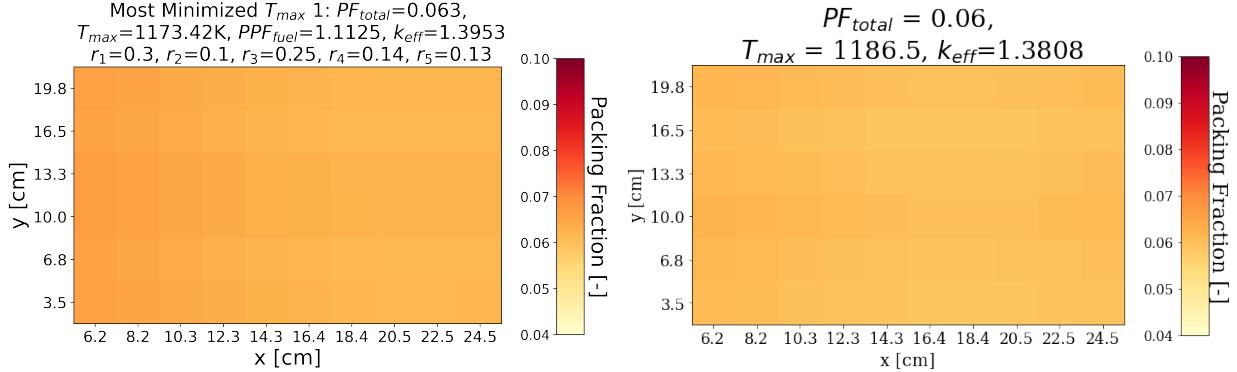


Figure 7.31: Simulation a-3b's most-minimized T_{max} TRISO distribution from Figure 7.23a (left) and simulation a-1b's most-minimized T_{max} TRISO distribution from Figure 7.4 (right).

Figure 7.31 shows that simulation a-3b's most-minimized T_{max} reactor model, and simulation a-1b's most-minimized T_{max} reactor model have similar almost constant TRISO distributions with packing fraction standard deviations of 0.001 and 0.0009, respectively. However, they have slightly different PF_{total} values, and simulation a-3b has a slightly stronger gradient from left to right.

In simulation a-3b, ROLLO found that the one-third assembly model with the most-minimized PPF_{fuel} objective, reactor model 4 (Figure 7.25a), has a slightly oscillating TRISO distribution along the y-axis. Figure 7.28 compares simulation a-3b's most-minimized PPF_{fuel} reactor model 4 and simulation a-1c's most-minimized PPF_{fuel} reactor model.

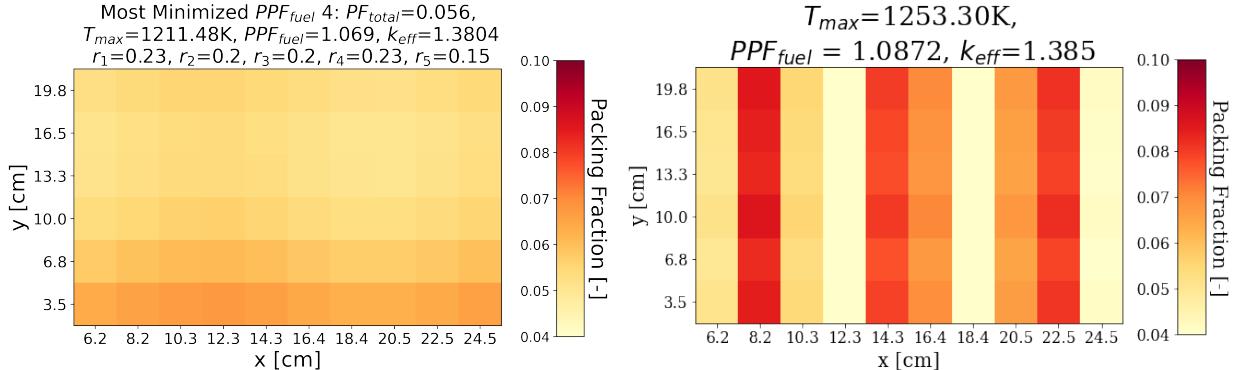


Figure 7.32: Simulation a-3b's most-minimized PPF_{fuel} TRISO distribution from Figure 7.25a (left) and simulation a-1c's most-minimized PPF_{fuel} TRISO distribution from Figure 7.6 (right).

Figure 7.32 shows that simulation a-3b's reactor model 4 and simulation a-1a's most-minimized PPF_{fuel} reactor models have small packing fraction standard deviation of 0.005 and 0.017, respectively. However, they do not follow the same TRISO distribution pattern; this is attributed to the

PF_{total} and PPF_{fuel} relationship resulting in unexpected TRISO distributions at different PF_{total} values, as mentioned previously.

Figure 7.33 shows reactor model 2, which minimized PF_{total} , T_{max} , and PPF_{fuel} to an equal extent by balancing influences from all objectives.

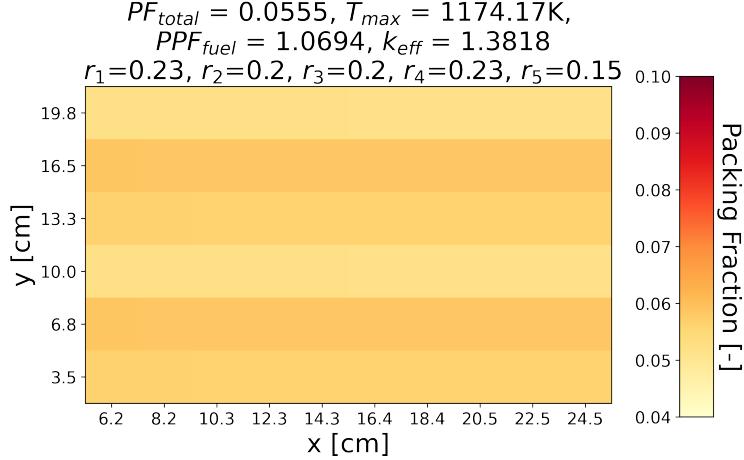
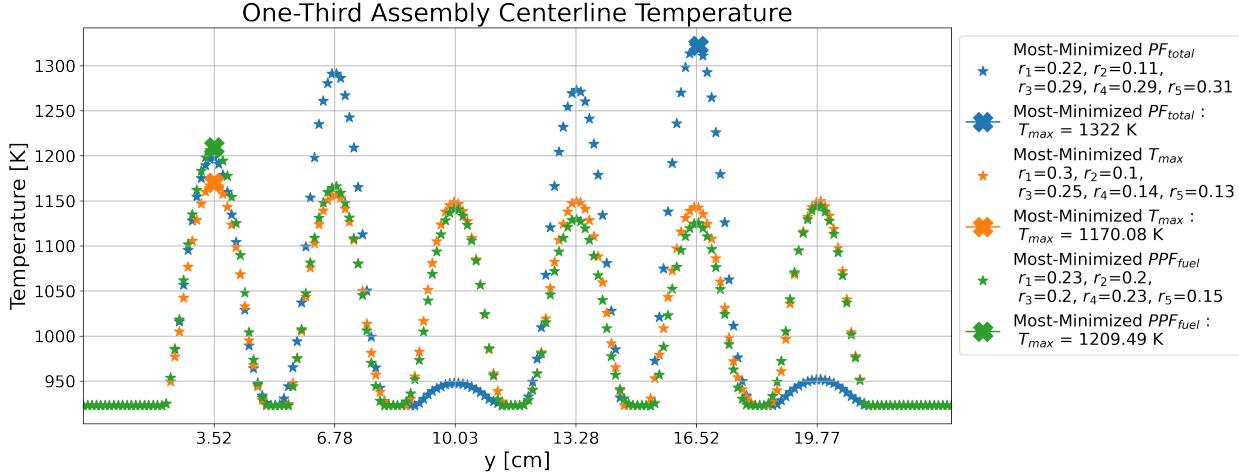


Figure 7.33: Simulation a-3b's reactor model 2 which minimized PF_{total} , T_{max} , and PPF_{fuel} to an equal extent (see Pareto Front in Figure 7.24a).

Similar to simulation a-3a, for all the reactor models on simulation a-3b's Pareto front (Figure 7.24a), the TRISO distribution flatness is strongly influenced by the minimize T_{max} objective. The variations in TRISO distributions are influenced by both the minimize PF_{total} and minimize PPF_{fuel} objectives. However, as mentioned previously, the PF_{total} and PPF_{fuel} relationship result in unexpected TRISO distributions at different PF_{total} values. The minimize PF_{total} objective tries to maximize the fission reaction rate to enable a higher k_{eff} for a lower PF_{total} , and the PPF_{fuel} objective tries to flatten thermal flux, resulting in the distribution observed in Figure 7.33.

Figure 7.34a shows the one-third assembly centerline temperatures for three reactors on simulation a-3b's Pareto front. Reactor model 11 with most-minimized PF_{total} peaks at 16.52cm with 1322K, reactor model 1 with most-minimized T_{max} peaks at 3.52cm with 1170K, and reactor model 4 with most-minimized PPF_{fuel} peaks at 3.52cm with 1209K. Section 7.2.4 concluded that the one-third assembly's FLiBe channels (corresponding to r_1, r_2, r_3, r_4, r_5) located closest to the temperature peaks are most important to minimizing T_{max} . However, all their radius values located closest to the temperature peaks in Figure 7.34a are not fully maximized (0.35cm). r_1, r_2, r_3, r_4 , and r_5 values correspond to the FLiBe channel at 18cm, 15cm, 12cm, 8cm, and 6cm, respectively.



(a) Centerline temperature. AHTR assembly's centerline is the white line in Figure 5.15.

Figure 7.34: Simulation a-3b's one-third assembly reactor models' temperature distribution. Reactor models are on simulation a-3b's Pareto front: reactor model 11 with most-minimized PF_{total} , reactor model 1 with most-minimized T_{max} , and reactor model 4 with most-minimized PPF_{fuel} . r_1, r_2, r_3, r_4 , and r_5 values correspond to the FliBe channel at 18cm, 15cm, 12cm, 8cm, and 6cm, respectively.

Figure 7.34a shows that reactor model 11 with most minimized PF_{total} temperature peaks in the 4th graphite plank (at 16.52cm) and has $r_1 = 0.22\text{cm}$ and $r_2 = 0.11\text{cm}$. Reactor model 1 with most minimized T_{max} temperature peaks in the 1st graphite plank (at 3.52cm) and has $r_5 = 0.13\text{cm}$, and reactor model 4 with most minimized PPF_{fuel} temperature peaks in the 1st graphite plank (at 3.52cm) and has $r_5 = 0.15\text{cm}$.

The unexpectedly small radius values could be due to the coolant channel shape not having a high impact on T_{max} compared to TRISO distribution; thus, ROLLO was more influenced by TRISO distribution when searching for optimal reactor models. During simulation a-3b's the genetic algorithm optimization process, variations in the coolant channel shape only minimize T_{max} by 10K, while variations in the TRISO distribution minimizes T_{max} by 150K.

Simulation a-3b's 128 individuals per generation could also be too small to explore reactor models with 12 input parameters. Simulation a-3b has 12 input parameters which is higher than all the other optimization simulations, which have 7 or fewer input parameters. Larger population size could enable ROLLO to explore more reactor model variations and potentially find even more optimal reactor models. Future work includes running simulation a-3b with a larger population size for more generations. Appendix B begins this preliminary future work.

7.4.4 Triple-Objective Optimization Major Takeaways

The triple-objective optimization simulations successfully found a wide spread of reactor models on their Pareto fronts that meet each objective to varying degrees. Simulation a-3b's multi-objective optimization shows the result of minimizing all three objectives (minimize PF_{total} , T_{max} , and PPF_{fuel}) while varying all the input parameters (PF_{total} , TRISO distribution, and coolant channel shape). Figure 7.24 shows the 12 widely spread out reactor models on simulation p-3b's Pareto front.

7.5 AHTR One-Third Assembly: Computational Cost Summary

Each ROLLO optimization simulation in this chapter runs the AHTR model *generations* \times *individuals* number of times with different geometries, resulting in high computationally cost. I report a summary of the computational cost to give readers a sense of how much computational power is required to run similar ROLLO reactor optimization simulations.

Optimization simulations are run on the Theta supercomputer at the Argonne Leadership Computing Facility under the Director's Discretionary Allocation Program [107]. Each Theta compute node has 64 processor cores with a nominal clock speed of 1.5GHz [107]. Each optimization simulation takes a different amount of node-hours due to differences in simulation software, tallies, and intermediate steps required. Table 7.22 reports the computational cost for each optimization simulation. One may refer to Table 7.2 for each simulation's parameters.

Table 7.22: Computational cost of Reactor evOLutionary aLgorithm Optimizer (ROLLO) simulations for optimizing Advanced High-Temperature Reactor (AHTR) one-third assembly. BW: BlueWaters Supercomputer, Theta: Theta supercomputer.

Num of Objs	Sim	Machine	Compute Cost Per Gen [node-hours]	Generations [#]	Total Compute Cost [node-hours]
1	a-1a	Theta	95.3	3	285.8
	a-1b	Theta	247.0	3	740.9
	a-1c	Theta	115.0	2	230.0
	a-1d	Theta	167.3	2	334.6
	a-1e	Theta	346.1	2	692.3
	a-1f	Theta	111.5	2	222.9
2	a-2a	Theta	250.2	5	1250.9
	a-2b	Theta	98.3	5	491.7
	a-2c	Theta	268.8	2	537.7
3	a-3a	Theta	273.6	5	1367.9
	a-3b	Theta	305.8	6	1830.0

7.6 Summary

This chapter described the Advanced High-Temperature Reactor (AHTR) one-third assembly's Reactor evOLutionary aLgorithm Optimizer (ROLLO) optimization results. I varied the following AHTR one-third assembly input parameters: Tristructural Isotropic (TRISO) packing fraction distribution ($\rho_{TRISO}(\vec{r})$), total fuel packing fraction (PF_{total}), and coolant channel shape; to minimize the following objectives: PF_{total} , maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly.

In Section 7.2's single-objective optimization simulations: a-1a, a-1b, a-1c, a-1d, a-1e, a-1f; and Section 7.2.4's discussions, I verified that each of the one-third assembly objective follows the same driving factors as the AHTR plank optimization objectives (Chapter 6) and described each objective's relationship with each input parameter. I determined that ROLLO flattens TRISO distribution and maximizes the coolant channel shape's radius values (r_1, r_2, r_3, r_4, r_5) that are located close to the reactor model's temperature peak to achieve the minimize T_{max} objective. The minimize PF_{total} objective is driven by maximizing the one-third assembly's total fission reaction rate and influences oscillations in the TRISO distribution to achieve the objective. The minimize PPF_{fuel} objective is driven by flattening the one-third assembly's thermal flux distribution and influences PF_{total} and oscillations in the TRISO distribution to achieve the objective. Both the minimize PF_{total} and minimize PPF_{fuel} objectives do not correlate with the coolant channel shape.

Simulation a-1b and a-1e results demonstrated that coolant channel shape variation does not have as high of an impact on T_{max} as TRISO distribution variation.

In Sections 7.3 and 7.4's multi-objective optimization simulations: a-2a, a-2b, a-2c, a-3a, a-3b; and the accompanying discussions in Section 7.3.4 and 7.4.3, I further analyzed how the objectives' combined effects resulted in the optimal reactor models found by each multi-objective optimization simulation. The multi-objective optimization simulations successfully found a wide spread of reactor models on their Pareto fronts that meet each objective to varying degrees. In the multi-objective optimization simulations, the minimize T_{max} objective continued to influence the flattening of the TRISO distribution and maximizing of the coolant channel shape's radius values (r_1, r_2, r_3, r_4, r_5) that are located close to the reactor model's temperature peak. Simulation a-2b results suggested that the minimize PF_{total} objective's driving factor maximize total fission reaction rate and minimize PPF_{fuel} objective's driving factor flattening thermal flux distribution influence each other resulting in unexpected TRISO distributions at different PF_{total} values.

Simulation a-3b is the final and largest multi-objective optimization problem that minimized all three objectives (PF_{total} , T_{max} , and PPF_{fuel}) while varying all the input parameters (PF_{total} , TRISO distribution, and coolant channel shape). Simulation a-3b ran for ~ 1800 node-hours on the Theta supercomputer with 6 generations and 128 reactor models per generation. Figure 7.24 shows the 12 reactor models on simulation a-3b's Pareto front that meet all three objectives. The reactor models on the Pareto Front have different PF_{total} , TRISO distributions, and coolant channel shapes, depending on the extent each objective is minimized due to the nature of multi-objective optimization that results in a tradeoff between objectives. These results demonstrate ROLLO's success in conducting a multi-objective global search of the large AHTR design space to find optimal reactor models that satisfy all the objectives. ROLLO also gives the reactor designer a sense of how sensitive each input parameter is in relation to the objectives. Once the ROLLO search is complete, reactor designers gain a better intuition of the model's reactor physics and can view the narrower reactor design space that meets their defined objectives. From there, reactor designers can determine the importance of each objective for their purposes, then conduct sensitivity analysis and use higher fidelity models to study the optimal design space further.

Chapter 8

Conclusions and Future Work

Additive manufacturing of reactor core components removes the geometric constraints imposed by conventional manufacturing, such as slabs as fuel planks and cylinders as fuel rods, enabling further optimization and improvement of core geometries. Wide-spread adoption of additive manufacturing methods in the nuclear industry could drastically reduce fabrication costs and deployment timelines, and improve reactor safety. Fully benefiting from the new ability to 3D print reactor components requires further research into generative reactor design optimization. This dissertation explores the new design space enabled by additive manufacturing by designing and applying the flexible and open-source Reactor evOLutionary aLgorithm Optimizer (ROLLO) tool to optimize the Advanced High-Temperature Reactor (AHTR) for non-conventional geometries and parameters. I successfully explored the AHTR’s arbitrary geometry design space by completing these three dissertation objectives:

1. I furthered our understanding of the AHTR design’s complexities through neutronics and temperature modeling by participating in the Organisation for Economic Co-operation and Development (OECD) Nuclear Energy Agency (NEA) Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark.
2. I developed the open-source ROLLO tool that enables generative reactor design using evolutionary algorithm optimization for non-conventional reactor geometries and fuel distributions.
3. I applied ROLLO to conduct generative AHTR design optimization. ROLLO generated AHTR designs with varying fuel amounts, fuel distributions, and coolant channel shapes that optimize for three key reactor performance metrics: minimize total fuel amount, maximize heat transfer, and minimize power peaking.

Chapter 3 addressed objective 1, chapter 4 addressed objective 2, and chapters 5, 6, and 7 addressed objective 3.

Chapter 3 reported the FHR benchmark Phase I-A and I-B results, demonstrating the AHTR’s passive safety behavior with negative temperature coefficients. A comparison of k_{eff} results between the reference case and the AHTR configuration with high heavy metal loading demonstrated that increased fuel packing does not always correspond with increased k_{eff} due to self-shielding effects. Chapter 3 also reported the AHTR full assembly temperature model results. The temperature distribution peaked in the fuel stripes near the spacers, highlighting to reactor designers that spacer material and location in the AHTR geometry impact temperature peaks. Through participation in the FHR benchmark, this dissertation contributes to deepening our understanding of the promising AHTR technology.

Chapter 4 described the ROLLO tool developed for this dissertation. ROLLO is a Python package that applies evolutionary algorithm optimization techniques to generate nuclear reactor designs that meet user-defined objectives and constraints based on user-defined input value ranges. ROLLO enables reactor designers to optimize any reactor model using robust evolutionary algorithm methods without going through the cumbersome process of setting up an evolutionary algorithm framework, selecting appropriate hyperparameters, and setting up parallelization. ROLLO is effective, flexible, accessible, parallel, reproducible, usable, and hosted on Github [12].

Chapter 5 described the modeling and optimization methodology of the AHTR plank and one-third assembly optimization conducted using the ROLLO software. I varied the following AHTR plank and one-third assembly input parameters: Tristructural Isotropic (TRISO) packing fraction distribution ($\rho_{TRISO}(\vec{r})$), total fuel packing fraction (PF_{total}), and coolant channel shape; to minimize the following objectives: total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}).

Chapter 6 reported the AHTR plank’s ROLLO optimization results. I characterized each objective’s driving factors and relationship with each input parameter from the results. I determined that the minimize PF_{total} objective is driven by maximizing the plank’s total fission reaction rate and influences oscillations in the TRISO distribution to achieve the objective. I determined that the minimize PPF_{fuel} objective is driven by flattening the plank’s thermal flux distribution and influ-

ences PF_{total} and oscillations in the TRISO distribution to achieve the objective. I determined that the minimize T_{max} objective flattens TRISO distribution and maximizes coolant channel shape's radius values to achieve the objective. The multi-objective optimization simulation results showed that the minimize PF_{total} and PPF_{fuel} driving factors influence each other resulting in unexpected TRISO distributions at different PF_{total} values. Further optimization of the AHTR design will benefit from awareness of the relationship between the objectives. The objectives' characterizations for the simple AHTR plank model provide insights to better understand Chapter 7's multi-objective optimization of the complex AHTR one-third assembly model.

Chapter 7 reported the AHTR one-third assembly's ROLLO optimization results. I verified that the one-third assembly objectives follow the same driving factors as the AHTR plank optimization objectives. Similar to the AHTR plank optimization results, these results demonstrated that the minimize PF_{total} and PPF_{fuel} driving factors influence each other resulting in unexpected TRISO distributions at different PF_{total} values. The results also demonstrated that the coolant channel shape variation did not have as high of an impact on T_{max} as the TRISO distribution variation. The final and largest optimization problem is the one-third assembly multi-objective optimization that minimized all three objectives (PF_{total} , T_{max} , and PPF_{fuel}) while varying all the input parameters (PF_{total} , TRISO distribution, and coolant channel shape), also known as simulation a-3b. Simulation a-3b ran for ~ 1800 node-hours on the Theta supercomputer with 6 generations and 128 reactor models per generation. Figure 7.24 showed the 12 reactor models on simulation a-3b's Pareto front that met all three objectives. The reactor models on the Pareto Front have different PF_{total} , TRISO distributions, and coolant channel shapes, depending on the extent each objective is minimized due to the nature of multi-objective optimization that results in a tradeoff between objectives. These results demonstrate that ROLLO's generative design optimization process provides the reactor designer with a set of equally good reactor models that minimize all objectives.

Chapters 6 and 7 demonstrated ROLLO's success in conducting multi-objective generative reactor design optimization. ROLLO conducted a global search of the large reactor design space and successfully generated optimal reactor models on the Pareto front that satisfy all the objectives. ROLLO's generative design optimization process also gives reactor designers unique insight into the

reactor design's problem domain and a sense of how sensitive each input parameter is in relation to the objectives. Once the ROLLO search is complete, reactor designers gain a better intuition for the model's reactor physics and can view the narrower reactor design space that meets their defined objectives. From this point, it is up to the reactor designer to determine the importance of each objective for their purposes, then conduct further sensitivity analysis and use higher fidelity models to study the optimal design space before selecting the final reactor model. By designing the ROLLO tool and demonstrating ROLLO's success in optimization of the AHTR beyond classical input parameters, this dissertation contributes to optimization tool development for reactors of the future.

8.1 Limitations and Future Work

This dissertation demonstrated results for the FHR benchmark's Phases I-A and I-B. The FHR benchmark is an ongoing project; thus, future work includes further contributions to the benchmark's Phases I-C, II, and III.

This dissertation demonstrated ROLLO's success in conducting multi-objective generative reactor design optimization. ROLLO could be further improved in the following ways: parallelization speed-up, JSON input file validation improvement, error message clarity, contingency kill switch implementation to minimize wasted compute time, unit testing expansion, and more nuclear reactor modeling software ROLLO coupling examples. These issues are detailed at <https://github.com/arfc/rollo/issues>.

This dissertation's generative design optimization of the AHTR model used sine distribution variations to govern the TRISO packing fraction distribution and cylinder radius variation to generate sinusoidal-like patterned coolant channel shapes. These input parameter variations are only one way to represent the AHTR geometry. There are other ways to represent the geometry that might give ROLLO more freedom to explore a larger design space and generate more optimized designs. Future reactor designers could consider topology optimization to optimize coolant channel shape further. The major limitation is the computational cost of modeling fluid flow in complex coolant channel designs.

Nuclear reactors represent physics interactions and feedbacks between thermal-hydraulics, neu-

tronics, material performance, structural mechanics, chemistry, and more. Multiphysics modeling is the practice of accounting for these physics interactions during design and analysis. This dissertation's generative design optimization modeled the AHTR model's beginning of life neutronics and temperature. Many multiphysics components were not included in this study; their inclusion in the AHTR optimization process might impact the optimal AHTR designs found by ROLLO. Future work includes taking into account the effect of burnup, structural mechanics, thermal stresses, chemistry, and higher fidelity thermal-hydraulics for AHTR model optimization. This relies on further developments in the reactor multiphysics modeling field through advanced coupling methods and narrowing down the important coupled physics interactions in the reactor types. The optimization work conducted in this dissertation utilized ~ 50000 node-hours on the BlueWaters and Theta supercomputers. Optimization work that models more complex reactor geometries with more multiphysics components will require even more compute time. Therefore, the feasibility of further optimization work also relies on software speed-ups and increased availability of compute time to conduct these simulations.

As additive manufacturing technology advances and the Transformational Challenge Reactor (TCR) program demonstrates the first 3D printed operational reactor, more reactor designers will begin to explore the vast design space enabled by 3D printing. ROLLO can be easily used to optimize any reactor type. Future work includes utilizing ROLLO to vary more reactor types for arbitrary geometries and parameters and optimize for objectives that address the many-faceted multiphysics interactions and feedbacks in the reactors.

References

- [1] D. Petti, “The future of nuclear energy in a carbon-constrained world,” *Massachusetts Institute of Technology Energy Initiative (MITEI)*, p. 272, 2018.
- [2] “U.S. Energy Information Administration (EIA),” *Direct Federal Financial Interventions and Subsidies in Energy in Fiscal Year 2016*, Apr. 2018. [Online]. Available: <https://www.eia.gov/analysis/requests/subsidy/>
- [3] “Climate change widespread, rapid, and intensifying,” *Intergovernmental Panel on Climate Change*, Aug. 2021. [Online]. Available: <https://www.ipcc.ch/2021/08/09/ar6-wg1-20210809-pr/>
- [4] *Climate Change and Nuclear Power 2018*, ser. Non-serial Publications. Vienna: International Atomic Energy Agency, 2018. [Online]. Available: <https://www.iaea.org/publications/13395/climate-change-and-nuclear-power-2018>
- [5] “Nuclear energy and climate change,” *World Nuclear Association*. [Online]. Available: <https://world-nuclear.org/nuclear-essentials/how-can-nuclear-combat-climate-change.aspx>
- [6] J. Simpson, J. Haley, C. Cramer, O. Shafer, A. Elliott, B. Peter, L. Love, and R. Dehoff, “Considerations for Application of Additive Manufacturing to Nuclear Reactor Core Components,” p. 43, May 2019.
- [7] K. A. Terrani, “Transformational Challenge Reactor demonstration program,” *Oak Ridge National Laboratory*, Nov. 2019. [Online]. Available: <http://tcr.ornl.gov>.
- [8] V. Sobes, B. Hiscox, E. Popov, M. Delchini, R. Archibald, P. Laiu, B. Betzler, and K. Terrani, “Artificial Intelligence Design Of Nuclear Systems Empowered By Advanced Manufacturing,” 2020, p. 8.
- [9] A. Bergeron and J. B. Crigger, “Early progress on additive manufacturing of nuclear fuel materials,” *Journal of Nuclear Materials*, vol. 508, pp. 344–347, 2018, ISBN: 0022-3115 Publisher: Elsevier.
- [10] Autodesk®, “Autodesk® Fusion 360™,” 2020. [Online]. Available: <https://www.autodesk.com/products/fusion-360/overview>
- [11] N. D. See, S. Cetiner, and B. R. Betzler, “Design Optimization of the Transformational Challenge Reactor Outlet Plenum,” *Nuclear Science and Engineering*, pp. 1–20, Jan. 2022. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/00295639.2021.2011571>
- [12] G. J. Y. Chee, “rollo: Reactor Evolutionary Algorithm Optimizer,” May 2021. [Online]. Available: <https://pypi.org/project/rollo/>

- [13] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, and K. Smith, “OpenMC: A state-of-the-art Monte Carlo code for research and development,” *Annals of Nuclear Energy*, vol. 82, no. Supplement C, pp. 90–97, Aug. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S030645491400379X>
- [14] A. Lindsay, G. Ridley, A. Rykhlevskii, and K. Huff, “Introduction to Moltres: An application for simulation of Molten Salt Reactors,” *Annals of Nuclear Energy*, vol. 114, pp. 530–540, Apr. 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0306454917304760>
- [15] G. I. I. Forum, “A Technology Roadmap Updtate for Generation IV Nuclear Energy Systems,” US DOE Nuclear Energy Research Advisory Committee and the Generation IV International Forum, Tech. Rep. GIF-002-00, Jan. 2014.
- [16] C. W. Forsberg, K. A. Terrani, L. L. Snead, and Y. Katoh, “Fluoride-Salt-Cooled High-Temperature Reactor (FHR) with Silicon-Carbide-Matrix Coated-Particle Fuel,” in *Transactions of the American Nuclear Society*, vol. 107, 2012, p. 907. [Online]. Available: <http://info.ornl.gov/sites/publications/files/Pub37875.pdf>
- [17] M. I. T. Facilitators, U. M. Facilitators, and U. B. Facilitators, “Fluoride-Salt-Cooled, High-Temperature Reactor (FHR) Development Roadmap and Test Reactor Performance Requirements White Paper,” 2013. [Online]. Available: <http://fhr.nuc.berkeley.edu/wp-content/uploads/2013/08/12-004-FHR-Workshop-4-Report-Final.pdf>
- [18] B. Petrovic, K. Ramey, and I. Hill, “Benchmark Specifications for the Fluoride-salt High-temperature Reactor (FHR) Reactor Physics Calculations,” Nuclear Energy Agency, Tech. Rep., Mar. 2021.
- [19] P. K. Romano and B. Forget, “The OpenMC Monte Carlo particle transport code,” *Annals of Nuclear Energy*, vol. 51, pp. 274–281, Jan. 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306454912003283>
- [20] M. B. Chadwick, M. Herman, P. Obložinský, M. E. Dunn, Y. Danon, A. C. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D. A. Brown, R. Capote, A. D. Carlson, Y. S. Cho, H. Derrien, K. Guber, G. M. Hale, S. Hoblit, S. Holloway, T. D. Johnson, T. Kawano, B. C. Kiedrowski, H. Kim, S. Kunieda, N. M. Larson, L. Leal, J. P. Lestone, R. C. Little, E. A. McCutchan, R. E. MacFarlane, M. MacInnes, C. M. Mattoon, R. D. McKnight, S. F. Mughabghab, G. P. A. Nobre, G. Palmiotti, A. Palumbo, M. T. Pigni, V. G. Pronyaev, R. O. Sayer, A. A. Sonzogni, N. C. Summers, P. Talou, I. J. Thompson, A. Trkov, R. L. Vogt, S. C. van der Marck, A. Wallner, M. C. White, D. Wiarda, and P. G. Young, “ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data,” *Nuclear Data Sheets*, vol. 112, no. 12, pp. 2887–2996, Dec. 2011.
- [21] C. W. Forsberg, P. F. Peterson, and P. S. Pickard, “Molten-salt-cooled advanced high-temperature reactor for production of hydrogen and electricity,” *Nuclear Technology*, vol. 144, no. 3, pp. 289–302, 2003, ISBN: 0029-5450 Publisher: Taylor & Francis.
- [22] M. K. M. Ho, G. H. Yeoh, and G. Braoudakis, “Molten salt reactors,” in *Materials and processes for energy: communicating current research and technological developments*, 2013th ed., ser. Energy Book Series, A. Méndez-Vilas, Ed. Badajoz, Spain: Formatex Research Center, 2013, no. 1, pp. 761–768, <http://www.formatex.info/energymaterialsbook/> <http://www.energymaterialsbook.org/chapters.html>.

- [23] M. W. Rosenthal, P. R. Kasten, and R. B. Briggs, “Molten-Salt Reactors - History, Status, and Potential,” *Nuclear Applications and Technology*, vol. 8, no. 2, pp. 107–117, Feb. 1970. [Online]. Available: <https://doi.org/10.13182/NT70-A28619>
- [24] R. O. Scarlat, M. R. Laufer, E. D. Blandford, N. Zweibaum, D. L. Krumwiede, A. T. Cisneros, C. Andreades, C. W. Forsberg, E. Greenspan, L.-W. Hu, and P. F. Peterson, “Design and licensing strategies for the fluoride-salt-cooled, high-temperature reactor (FHR) technology,” *Progress in Nuclear Energy*, vol. 77, pp. 406–420, Nov. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0149197014001838>
- [25] R. O. Scarlat and P. F. Peterson, “The current status of fluoride salt cooled high temperature reactor (FHR) technology and its overlap with HIF target chamber concepts,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 733, pp. 57–64, 2014, ISBN: 0168-9002 Publisher: Elsevier. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168900213007055>
- [26] D. L. Krumwiede, R. O. Scarlat, J. K. Choi, T. M. Phan, and P. F. Peterson, “Three-dimensional modeling of the pebble-bed fluoride-salt-cooled, high-temperature reactor (PB-FHR) commercial plant design,” in *Proceedings of the American Nuclear Society 2013 Winter Meeting*, 2013.
- [27] L. Liu, D. Zhang, Q. Lu, K. Wang, and S. Qiu, “Preliminary neutronic and thermal-hydraulic analysis of a 2 MW Thorium-based Molten Salt Reactor with Solid Fuel,” *Progress in Nuclear Energy*, vol. 86, pp. 1–10, Jan. 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0149197015300779>
- [28] D. E. Holcomb, D. Ilas, V. K. Varma, A. T. Cisneros, R. P. Kelly, and J. C. Gehin, “Core and refueling design studies for the advanced high temperature reactor,” *ORNL/TM-2011/365, Oak Ridge National Laboratory, Oak Ridge, TN*, 2011.
- [29] V. K. Varma, D. E. Holcomb, F. J. Peretz, E. C. Bradley, D. Ilas, A. L. Qualls, and N. M. Zaharia, “AHTR mechanical, structural, and neutronic preconceptual design,” *Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States)*, Tech. Rep., 2012.
- [30] S. R. Greene, J. C. Gehin, D. E. Holcomb, J. J. Carbajo, D. Ilas, A. T. Cisneros, V. K. Varma, W. R. Corwin, D. F. Wilson, and G. L. Yoder Jr, “Pre-conceptual design of a fluoride-salt-cooled small modular advanced high-temperature reactor (SmAHTR),” *Oak Ridge National Laboratory, ORNL/TM-2010/199*, 2010.
- [31] K. M. Ramey and B. Petrovic, “Monte Carlo modeling and simulations of AHTR fuel assembly to support V&V of FHR core physics methods,” *Annals of Nuclear Energy*, vol. 118, pp. 272–282, Aug. 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0306454918301816>
- [32] F. Rahnema, D. Diamond, D. Serghiuta, and P. Burke, “Phenomena, gaps, and issues for neutronics modeling and simulation of FHRs,” *Annals of Nuclear Energy*, vol. 123, pp. 172–179, Jan. 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306454918304572>

- [33] F. Rahnema, B. Petrovic, C. Edgar, D. Zhang, P. Avigni, M. Huang, and S. Terlizzi, “The Current Status of the Tools for Modeling and Simulation of Advanced High Temperature Reactor Neutronics Analysis,” Georgia Institute of Technology, Tech. Rep., Dec. 2015. [Online]. Available: <http://hdl.handle.net/1853/55803>
- [34] D. E. Holcomb, G. F. Flanagan, G. T. Mays, W. D. Pointer, K. R. Robb, and G. L. Yoder Jr, “Fluoride salt-cooled high-temperature reactor technology development and demonstration roadmap,” Oak Ridge National Laboratory, Oak Ridge, TN, Tech. Rep., Sep. 2013.
- [35] D. Zhang and F. Rahnema, “Integrated Approach to Fluoride High Temperature Reactor Technology and Licensing Challenges (FHR-IRP),” Georgia Inst. of Technology, Atlanta, GA (United States), Tech. Rep., Mar. 2019. [Online]. Available: <https://doi.org/10.2172/1505504>
- [36] A. T. Cisneros and D. Ilas, “Neutronics and Depletion Methods for Parametric Studies of Fluoride Salt Cooled High Temperature Reactors with Slab Fuel Geometry and Multi-Batch Fuel Management Schemes,” *Nuclear Technology*, vol. 183, p. 11, 2013.
- [37] L. Mei, X. Cai, D. Jiang, J. Chen, Y. Zhu, Y. Liu, and X. Wang, “The investigation of thermal neutron scattering data for molten salt Flibe,” *Journal of Nuclear Science and Technology*, vol. 50, no. 7, pp. 682–688, 2013, iISBN: 0022-3131 Publisher: Taylor & Francis.
- [38] Y. Zhu and A. I. Hawari, “Thermal neutron scattering cross section of liquid FLiBe,” *Progress in Nuclear Energy*, vol. 101, pp. 468–475, 2017, iISBN: 0149-1970 Publisher: Elsevier.
- [39] C. A. Gentry, “Development of a Reactor Physics Analysis Procedure for the Plank-Based and Liquid Salt-Cooled Advanced High Temperature Reactor,” Ph.D. dissertation, University of Tennessee - Knoxville, 2016. [Online]. Available: https://trace.tennessee.edu/utk_graddiss/3695
- [40] J. Leppanen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho, “The Serpent Monte Carlo code: Status, development and applications in 2013,” *Annals of Nuclear Energy*, vol. 82, pp. 142–150, Aug. 2014.
- [41] P. Turinsky, R. Al-Chalabi, P. Engrand, H. Sarsour, F. Faure, and W. Guo, “NESTLE: Few-group neutron diffusion equation solver utilizing the nodal expansion method for eigenvalue, adjoint, fixed-source steady-state and transient problems,” Tech. Rep. EGG-NRE-11406, 10191160, Jun. 1994. [Online]. Available: <https://doi.org/10.2172/10191160>
- [42] K. Koebke, “A new approach to homogenization and group condensation,” International Atomic Energy Agency (IAEA), Tech. Rep., 1980, iAEA-TECDOC-231 INIS Reference Number: 12619334.
- [43] H.-C. Lin, “Thermal Hydraulics System-Level Code Validation and Transient Analyses for Fluoride Salt-Cooled High-Temperature Reactors,” Ph.D. dissertation, University of Michigan, 2020. [Online]. Available: <https://hdl.handle.net/2027.42/155105>
- [44] B. A. Lindley, J. G. Hosking, P. J. Smith, D. J. Powney, B. S. Tollit, T. D. Newton, R. Perry, T. C. Ware, and P. N. Smith, “Current status of the reactor physics code WIMS and recent developments,” *Annals of Nuclear Energy*, vol. 102, pp. 148–157, 2017, iISBN: 0306-4549 Publisher: Elsevier.

- [45] B. Petrovic, K. Ramey, I. Hill, E. Losa, M. Elsawi, Z. Wu, C. Lu, J. Gonzales, D. Novog, and G. Chee, “Preliminary Results of the NEA FHR Benchmark Phase IA and IB (Fuel Element 2-D Benchmark),” in *The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Raleigh, NC*, 2021.
- [46] B. Petrovic and K. M. Ramey, “FHR/AHTR NEA Benchmark,” Georgia Tech, Atlanta, Oct. 2019. [Online]. Available: http://montecarlo.vtt.fi/mtg/2019_Atlanta/Petrovic1.pdf
- [47] X.- M. C. Team, “MCNP — A General Monte Carlo N-Particle Transport Code, Version 5,” LANL, Los Alamos, NM, Tech. Rep. LA-UR-03-1987, Apr. 2003.
- [48] R. Macfarlane, D. W. Muir, R. M. Boicourt, A. C. Kahler III, and J. L. Conlin, “The NJOY nuclear data processing system, version 2016,” Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2017.
- [49] S. M. Park, “Advancement and Verification of Moltres for Molten Salt Reactor Safety Analysis,” Master’s thesis, University of Illinois at Urbana-Champaign, Urbana, IL, Aug. 2020. [Online]. Available: <https://www.ideals.illinois.edu/handle/2142/108542>
- [50] I. Gibson, D. W. Rosen, and B. Stucker, *Additive manufacturing technologies*. Springer, 2014, vol. 17. [Online]. Available: <https://doi.org/10.1007/978-3-030-56127-7>
- [51] A. Standard, “Standard terminology for additive manufacturing technologies,” *ASTM International F2792-12a*, 2012.
- [52] A. Uriondo, M. Esperon-Miguez, and S. Perinpanayagam, “The present and future of additive manufacturing in the aerospace sector: A review of important aspects,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 229, no. 11, pp. 2132–2147, 2015, iISBN: 0954-4100 Publisher: SAGE Publications Sage UK: London, England.
- [53] “Printed titanium parts expected to save millions in Boeing Dreamliner costs,” *Reuters*, Apr. 2017. [Online]. Available: <https://www.reuters.com/article/us-norsk-boeing-idUSKBN17C264>
- [54] “Transformation In 3D: How A Walnut-Sized Part Changed The Way GE Aviation Builds Jet Engines | GE News,” 2018. [Online]. Available: <http://www.ge.com/news/reports/transformation-3d-walnut-sized-part-changed-way-ge-aviation-builds-jet-engines>
- [55] “Roadmap for Regulatory Acceptance of Advanced Manufacturing Methods in the Nuclear Energy Industry,” Nuclear Energy Institute, NEI Report. [Online]. Available: <https://www.nrc.gov/docs/ML1913/ML19134A087.pdf>
- [56] N. Sridharan, M. N. Gussev, and K. G. Field, “Performance of a ferritic/martensitic steel for nuclear reactor applications fabricated using additive manufacturing,” *Journal of Nuclear Materials*, vol. 521, pp. 45–55, 2019, iISBN: 0022-3115 Publisher: Elsevier.
- [57] B. Betzler, B. Ade, A. Wysocki, P. Jain, P. Chessier, M. Greenwood, and K. Terrani, “Transformational Challenge Reactor preconceptual core design studies,” *Nuclear Engineering and Design*, vol. 367, p. 110781, Oct. 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0029549320302752>

- [58] M. P. Trammell, B. C. Jolly, M. D. Richardson, A. T. Schumacher, and K. A. Terrani, “Advanced Nuclear Fuel Fabrication: Particle Fuel Concept for TCR,” Tech. Rep., 2019.
- [59] J. Rosales, I. J. van Rooyen, and C. J. Parga, “Characterizing surrogates to develop an additive manufacturing process for U₃Si₂ nuclear fuel,” *Journal of Nuclear Materials*, vol. 518, pp. 117–128, 2019, iISBN: 0022-3115 Publisher: Elsevier.
- [60] T. Koyanagi, K. Terrani, S. Harrison, J. Liu, and Y. Katoh, “Additive manufacturing of silicon carbide for nuclear applications,” *Journal of Nuclear Materials*, vol. 543, p. 152577, 2020, iISBN: 0022-3115 Publisher: Elsevier.
- [61] C. Sauder, “Ceramic matrix composites: nuclear applications,” *Ceramic matrix composites: materials, modeling and technology*, pp. 609–646, 2014, publisher: Wiley Online Library.
- [62] L. L. Snead, T. Nozawa, Y. Katoh, T.-S. Byun, S. Kondo, and D. A. Petti, “Handbook of SiC properties for fuel performance modeling,” *Journal of nuclear materials*, vol. 371, no. 1-3, pp. 329–377, 2007, iISBN: 0022-3115 Publisher: Elsevier.
- [63] K. A. Terrani, J. O. Kiggans, C. M. Silva, C. Shih, Y. Katoh, and L. L. Snead, “Progress on matrix SiC processing and properties for fully ceramic microencapsulated fuel form,” *Journal of Nuclear Materials*, vol. 457, pp. 9–17, 2015, iISBN: 0022-3115 Publisher: Elsevier.
- [64] J. Byrne, P. Cardiff, A. Brabazon, and M. O'Neill, “Evolving parametric aircraft models for design exploration and optimisation,” *Neurocomputing*, vol. 142, pp. 39–47, Oct. 2014. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S092523121400530X>
- [65] H. A. Simon, *The sciences of the artificial*. MIT press, 2019.
- [66] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.
- [67] W. F. Sacco and C. M. Pereira, “Two stochastic optimization algorithms applied to nuclear reactor core design,” *Progress in Nuclear Energy*, vol. 48, no. 6, pp. 525–539, 2006, iISBN: 0149-1970 Publisher: Elsevier.
- [68] W. F. Sacco, H. Alves Filho, N. Henderson, and C. R. de Oliveira, “A Metropolis algorithm combined with Nelder–Mead Simplex applied to nuclear reactor core design,” *Annals of Nuclear Energy*, vol. 35, no. 5, pp. 861–867, 2008, iISBN: 0306-4549 Publisher: Elsevier.
- [69] F. Y. Odeh and W. S. Yang, “Core design optimization and analysis of the Purdue Novel Modular Reactor (NMR-50),” *Annals of Nuclear Energy*, vol. 94, pp. 288–299, 2016, iISBN: 0306-4549 Publisher: Elsevier.
- [70] C. D. Fletcher and R. R. Schultz, “RELAP5/MOD3 code manual,” Nuclear Regulatory Commission, Tech. Rep., 1992.
- [71] D. J. Kropaczek and R. Walden, “Large-Scale Application of the Constraint Annealing Method for Pressurized Water Reactor Core Design Optimization,” *Nuclear Science and Engineering*, vol. 193, no. 5, pp. 523–536, 2019, iISBN: 0029-5639 Publisher: Taylor & Francis.

- [72] ——, “Constraint annealing method for solution of multiconstrained nuclear fuel cycle optimization problems,” *Nuclear Science and Engineering*, vol. 193, no. 5, pp. 506–522, 2019, ISBN: 0029-5639 Publisher: Taylor & Francis.
- [73] B. R. Betzler, D. Chandler, D. H. Cook, E. E. Davidson, and G. Ilas, “Design optimization methods for high-performance research reactor core design,” *Nuclear Engineering and Design*, vol. 352, p. 110167, 2019, ISBN: 0029-5493 Publisher: Elsevier.
- [74] C. M. Pereira and C. M. Lapa, “Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem,” *Annals of Nuclear Energy*, vol. 30, no. 5, pp. 555–565, 2003, ISBN: 0306-4549 Publisher: Elsevier.
- [75] C. M. Pereira and W. F. Sacco, “A parallel genetic algorithm with niching technique applied to a nuclear reactor core design optimization problem,” *Progress in Nuclear Energy*, vol. 50, no. 7, pp. 740–746, Sep. 2008. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S014919700800019X>
- [76] S. Kamalpour and H. Khalafi, “SMART reactor core design optimization based on FCM fuel,” *Nuclear Engineering and Design*, p. 110970, 2020, ISBN: 0029-5493 Publisher: Elsevier.
- [77] A. Kumar and P. V. Tsvetkov, “A new approach to nuclear reactor design optimization using genetic algorithms and regression analysis,” *Annals of Nuclear Energy*, vol. 85, pp. 27–35, Nov. 2015. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0306454915002285>
- [78] K. Meintjes, ““Generative Design” – What’s That? - CIMdata.” [Online]. Available: <https://www.cimdata.com/en/news/item/8402-generative-design-what-s-that>
- [79] S. Krish, “A practical generative design method,” *Computer-Aided Design*, vol. 43, no. 1, pp. 88–100, Jan. 2011. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0010448510001764>
- [80] G. Renner and A. Ekárt, “Genetic algorithms in computer aided design,” *Computer-aided design*, vol. 35, no. 8, pp. 709–726, 2003.
- [81] D. E. Goldberg, “Genetic algorithms in search,” *Optimization, and MachineLearning*, 1989, publisher: Addison Wesley Publishing Co. Inc.
- [82] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002, ISBN: 1089-778X Publisher: IEEE.
- [83] D. E. Goldberg, K. Deb, and D. Thierens, “Toward a better understanding of mixing in genetic algorithms,” *Journal of the Society of Instrument and Control Engineers*, vol. 32, no. 1, pp. 10–16, 1993, ISBN: 0453-4662 Publisher: The Society of Instrument and Control Engineers.
- [84] A. Ng, K. Katanforoosh, and Y. B. Mourri, “Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization,” 2021. [Online]. Available: <https://www.coursera.org/learn/deep-neural-network>

- [85] J. Jordan, “Hyperparameter tuning for machine learning models.” Nov. 2017. [Online]. Available: <https://www.jeremyjordan.me/hyperparameter-tuning/>
- [86] G. Chee, “arfcc/fhr-benchmark,” 2021. [Online]. Available: <https://github.com/arfc/fhr-benchmark>
- [87] NCSA, “About Blue Waters: The National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign,” 2017, <http://www.ncsa.illinois.edu/>. [Online]. Available: <http://www.ncsa.illinois.edu/>
- [88] M. Torabi, A. Lashkari, S. F. Masoudi, and S. Bagheri, “Neutronic analysis of control rod effect on safety parameters in Tehran Research Reactor,” *Nuclear Engineering and Technology*, vol. 50, no. 7, pp. 1017–1023, 2018, iISBN: 1738-5733 Publisher: Elsevier.
- [89] A. Lindsay, “Moltres, software for simulating Molten Salt Reactors,” University of Illinois at Urbana-Champaign, 2017, <https://github.com/arfc/moltres>. [Online]. Available: <https://github.com/arfc/moltres>
- [90] J. A. Bucholz, “SCALE: A modular code system for performing standardized computer analyses for licensing evaluation,” Oak Ridge National Lab., TN (USA), NUREG NUREG/CR-0200-Vol.2-Bk.2; ORNL/NUREG/CSD-2-Vol.2-Bk.2 ON: DE82013370; TRN: 82-013156, 1982. [Online]. Available: <http://www.osti.gov/scitech/biblio/5360496>
- [91] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579>
- [92] K. M. Ramey, “Methodology for Coupled Reactor Physics, Thermal Hydraulics, and Depletion of a Fluoride Salt-Cooled High-Temperature Reactor (FHR),” Ph.D. dissertation, Georgia Institute of Technology, 2021.
- [93] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, “DEAP: Evolutionary algorithms made easy,” *Journal of Machine Learning Research*, vol. 13, no. Jul, pp. 2171–2175, 2012.
- [94] J. Blank and K. Deb, “Pymoo: Multi-objective optimization in python,” *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020, iISBN: 2169-3536 Publisher: IEEE.
- [95] A. Garrett, “inspyred: Bio-inspired Algorithms in Python,” URL: <https://pypi.python.org/pypi/inspyred> (visited on 11/28/2016), 2014.
- [96] C. S. Perone, “Pyevolve: a Python open-source framework for genetic algorithms,” *Acm Sigevolution*, vol. 4, no. 1, pp. 12–20, 2009.
- [97] C. Gagné and M. Parizeau, “Open BEAGLE: A New Versatile C++ Framework for Evolutionary Computation.” in *GECCO Late Breaking Papers*. Citeseer, 2002, pp. 161–168.
- [98] G. J. Y. Chee, “Documentation for rollo (Reactor Evolutionary Algorithm Optimizer),” 2021. [Online]. Available: <https://gwenchee.github.io/rollo/>
- [99] Ronacher, “Welcome to Jinja2 - Jinja2 Documentation (2.10),” 2018. [Online]. Available: <http://jinja.pocoo.org/docs/2.10/>

- [100] R. Smallshire, “multiprocessing_on_dill 3.5.0a4 : A friendly fork of multiprocessing which uses dill instead of pickle.” [Online]. Available: https://github.com/sixty-north/multiprocessing_on_dill
- [101] D. Ackley, *A connectionist machine for genetic hillclimbing*. Springer Science & Business Media, 2012, vol. 28.
- [102] S. Surjanovic and D. Bingham, “Ackley Function,” *Virtual Library of Simulation Experiment: Test Functions and Datasets.*, 2013. [Online]. Available: <https://www.sfu.ca/~ssurjano/ackley.html>
- [103] T. T. Binh and U. Korn, “MOBES: A multiobjective evolution strategy for constrained optimization problems,” in *The third international conference on genetic algorithms (Mendel 97)*, vol. 25, 1997, p. 27.
- [104] M. Bassi, E. S. d. Cursi, E. Pagnacco, and R. Ellaia, “Statistics of the Pareto front in Multi-objective Optimization under Uncertainties,” *Latin American Journal of Solids and Structures*, vol. 15, Nov. 2018, publisher: Associação Brasileira de Ciências Mecânicas. [Online]. Available: <http://www.scielo.br/j/lajss/a/zF63wFPmYFjmKZ8sWkg6MFy/?format=html&lang=en>
- [105] A. Blanchard, K. R. Yates, J. F. Zino, D. Biswas, S. C. Aiken, D. E. Carlson, H. Hoang, and D. Heemstra, “Updated Critical Mass Estimates for Plutonium-238,” Savannah River Site, Tech. Rep., 1999.
- [106] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, “The Hypervolume Indicator: Problems and Algorithms,” *arXiv:2005.00515 [cs]*, May 2020, arXiv: 2005.00515. [Online]. Available: <http://arxiv.org/abs/2005.00515>
- [107] “Theta/ThetaGPU,” *Argonne Leadership Computing Facility*, 2022. [Online]. Available: <https://www.alcf.anl.gov/alcf-resources/theta>
- [108] A. Novak, *Multiscale Thermal-Hydraulic Methods for Pebble Bed Reactors*. University of California, Berkeley, 2020.

Appendix A: Reproducibility

Reproducible science refers to the concept that all research outputs should be repeatable using a complete computational environment consisting of the software application, input files, and data results [108]. This section describes the complete computational environment used to obtain the results in this dissertation.

The UIUC team's FHR benchmark results described in Chapter 3 are available on Github at:

<https://github.com/arfc/fhr-benchmark>

Instructions on how to reproduce the benchmark results are included in READMEs within the **fhr-benchmark** directory.

The ROLLO project described in Chapter 4 is hosted on Github at:

<https://github.com/arfc/rollo>

Instructions on how to install the ROLLO Python package and documentation are provided in the README.

The Latex source files and figures used to compile this dissertation document, and the input files and data results for the AHTR temperature model and optimization simulations are available on Github at:

<https://github.com/arfc/2022-chee-dissertation>

Table A.1 provides paths for the tests, input files, and data files referenced in this dissertation in order of appearance.

Table A.1: Paths to tests, input files, and data files for the simulations performed in this dissertation.

Section	Description	Location
3.3.1	FHR Benchmark Phase I-A Input Files	fhr-benchmark/phase1a
3.3.2	FHR Benchmark Phase I-B Input Files	fhr-benchmark/phase1b
3.4	FHR Benchmark Temperature Model	2022-chee-dissertation/data/fhr-benchmark-temp-model
4.4	ROLLO Verification Problems	rollo/examples
5.4.1	AHTR Plank Key Neutronics Parameter Verification	2022-chee-dissertation/data/ahtr-opt-plank/verification
5.4.2	AHTR One-Third Assembly Key Neutronics Parameter Verification	2022-chee-dissertation/data/ahtr-opt-assem/verification
5.5	ROLLO Hyperparameter Tuning Studies	2022-chee-dissertation/data/rollo-hyperparameters
6.2	AHTR Plank Single-Objective Optimization Results	2022-chee-dissertation/data/ahtr-opt-plank/single-obj
6.3	AHTR Plank Two-Objective Optimization Results	2022-chee-dissertation/data/ahtr-opt-plank/two-obj
6.4	AHTR Plank Three-Objective Optimization Results	2022-chee-dissertation/data/ahtr-opt-plank/three-obj
7.2	AHTR One-Third Assembly Single-Objective Optimization Results	2022-chee-dissertation/data/ahtr-opt-assem/single-obj
7.3	AHTR One-Third Assembly Two-Objective Optimization Results	2022-chee-dissertation/data/ahtr-opt-assem/two-obj
7.4	AHTR One-Third Assembly Three-Objective Optimization Results	2022-chee-dissertation/data/ahtr-opt-assem/three-obj

Appendix B: Simulation a-3b-256

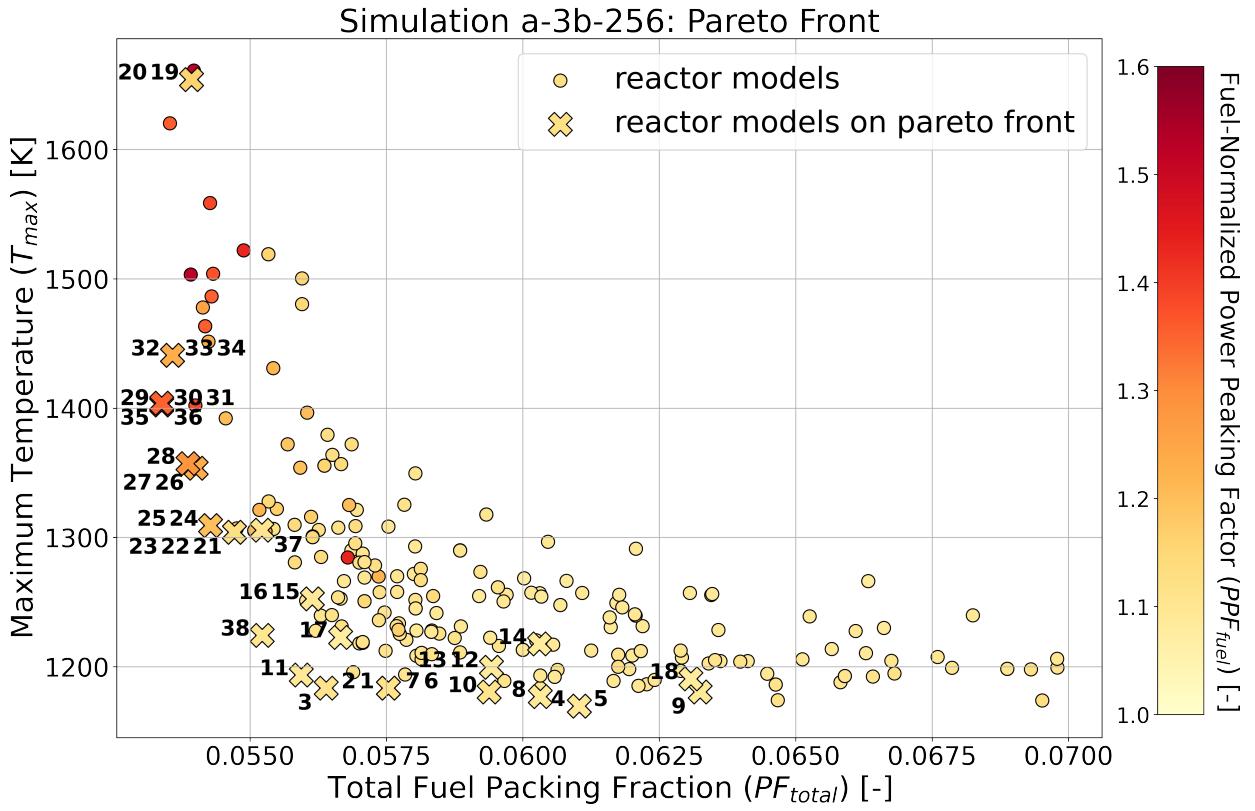
Simulation a-3b-256 has the exact same optimization problem parameters as simulation a-3b (Table 7.20) except for an increase in population size to 256 individuals. Table B.1 shows the hypervolume value at each generation. The compute time required to run simulation a-3b-256 for four generations is 3359 node-hours. I ran simulation a-3b-256 for only four generations due to limited computational resources.

Table B.1: Simulation a-3b-256 hypervolume values at each generation.

Three Objectives: Simulation a-3b-256	
Reference point: (0.07, 1700, 1.8)	
Generation	Hypervolume [-]
1	5.7867
2	5.8019
3	5.9412
4	5.9675

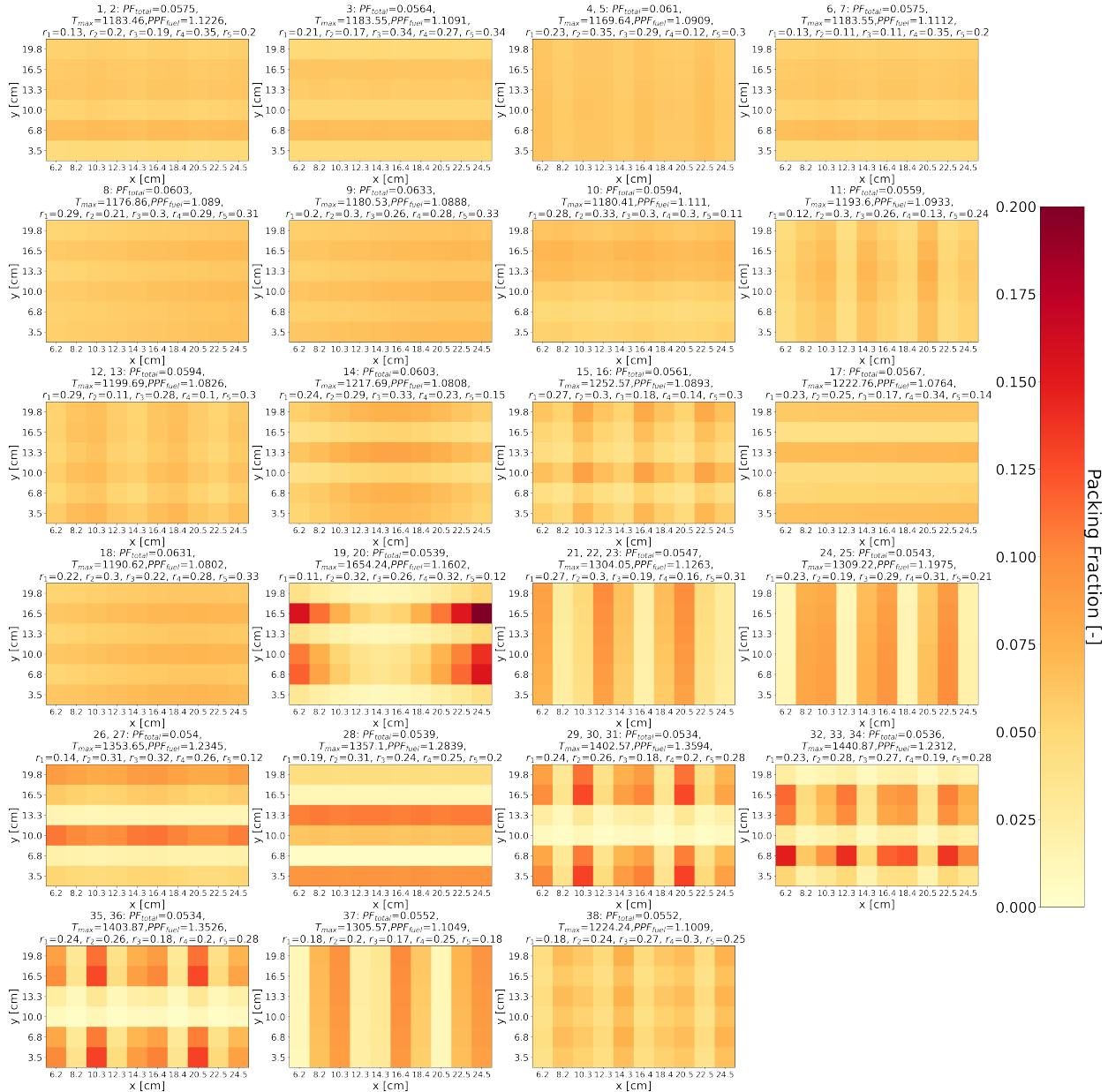
Figure B.1a shows a plot of the final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} ; crosses mark the reactor models that fall on the Pareto front. Figure B.1b shows the 38 TRISO packing fraction distributions in the final generation, labeled numerically, that fall on the Pareto front.

Figure B.1 demonstrates that ROLLO found 38 reactor models on simulation a-3b-256 final generation's Pareto front. Figure B.2 shows three reactor models on the Pareto front that most minimized each objective and one reactor model on the Pareto front that equally minimized all three objectives. I selected the equally minimized reactor model by visually studying Figure B.1 and selecting a reactor model close to the origin with a light yellow color dimension. Reactor model 36 most-minimized PF_{total} , reactor model 4 most-minimized T_{max} , reactor model 17 most-minimized PPF_{fuel} , and reactor model 11 equally minimized all three objectives.



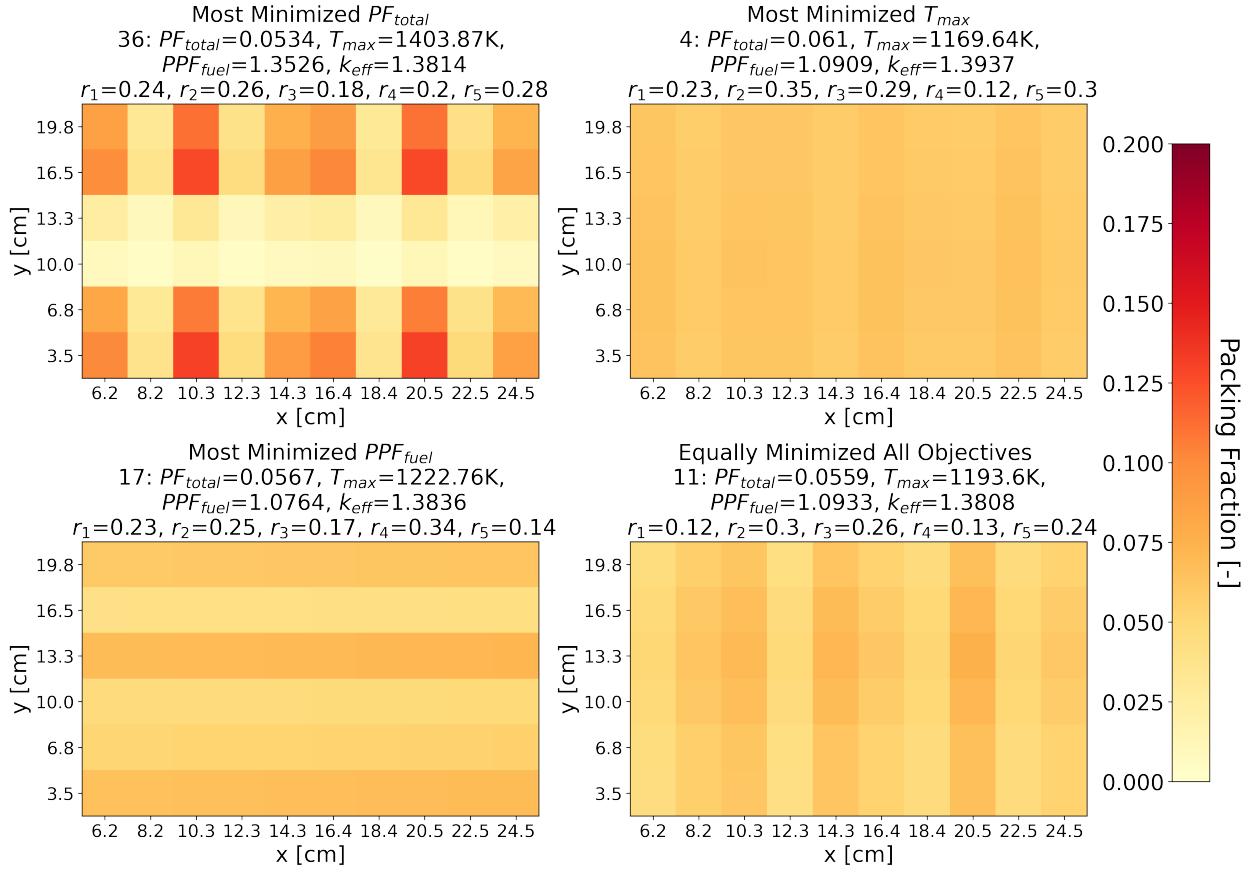
(a) Plot of final generation's reactor models' PF_{total} against T_{max} against PPF_{fuel} as a color dimension. Crosses indicate the reactor models on the Pareto front. Cross numbering correspond to TRISO distributions in Figure 7.24b.

Figure B.1: Simulation a-3b-256 – ROLLO triple-objective optimization with 256 population size to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).



(b) TRISO distributions for the 38 reactor models on the Pareto front. Numbered reactor models correspond to numbered crosses in Figure B.1a. Note that some models have identical distributions, resulting in the 23 plots in this subfigure.

Figure B.1: (contd.) Simulation a-3b-256 – ROLLO triple-objective optimization with 256 population size to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).



(a) TRISO packing fraction distributions.

Figure B.2: AHTR one-third assembly models and TRISO distributions for the 3 reactor models on simulation a-3b-256's Pareto front that most-minimized each objective, and 1 reactor model that equally minimized all three objectives. Simulation a-3b – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).

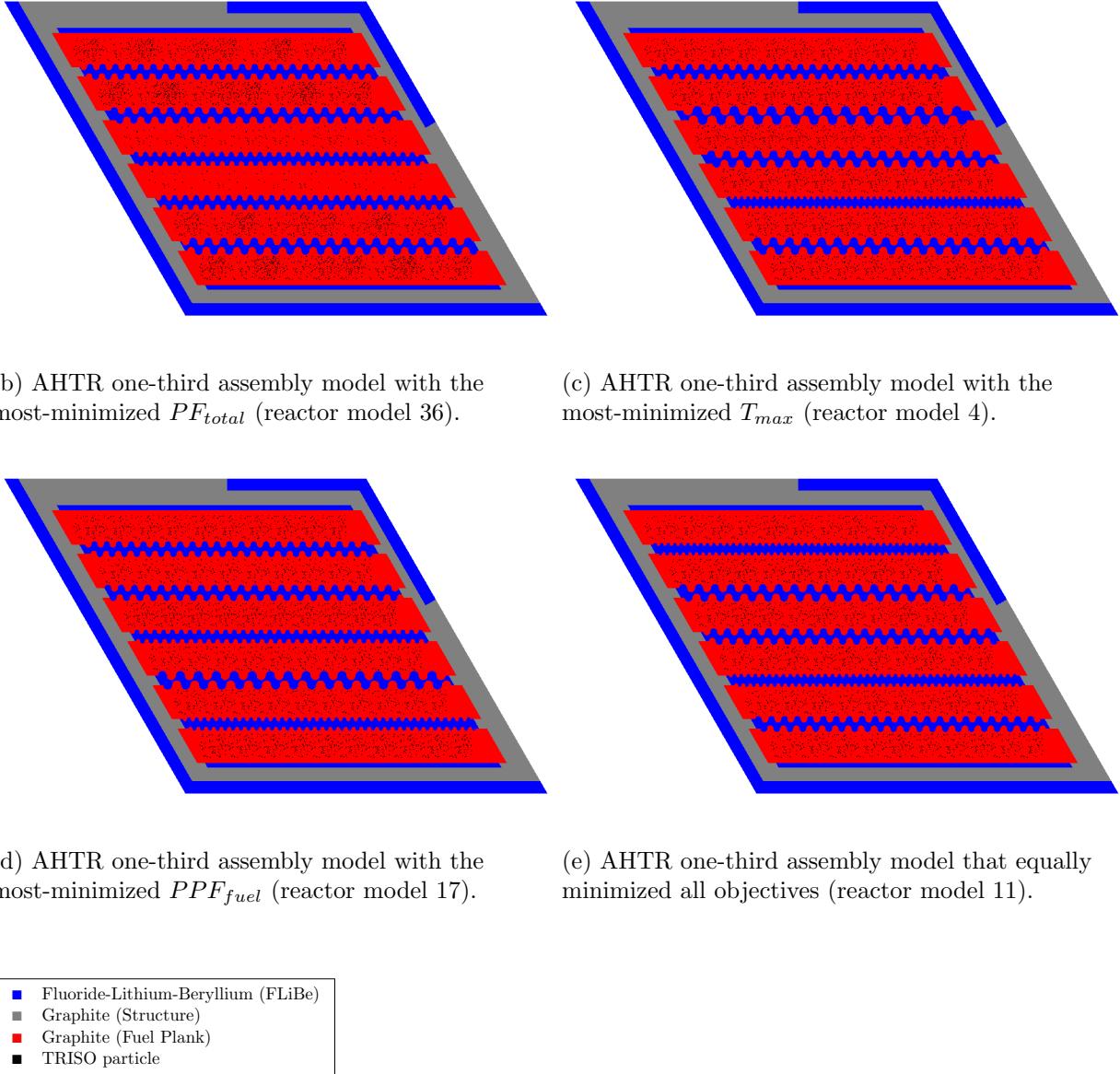
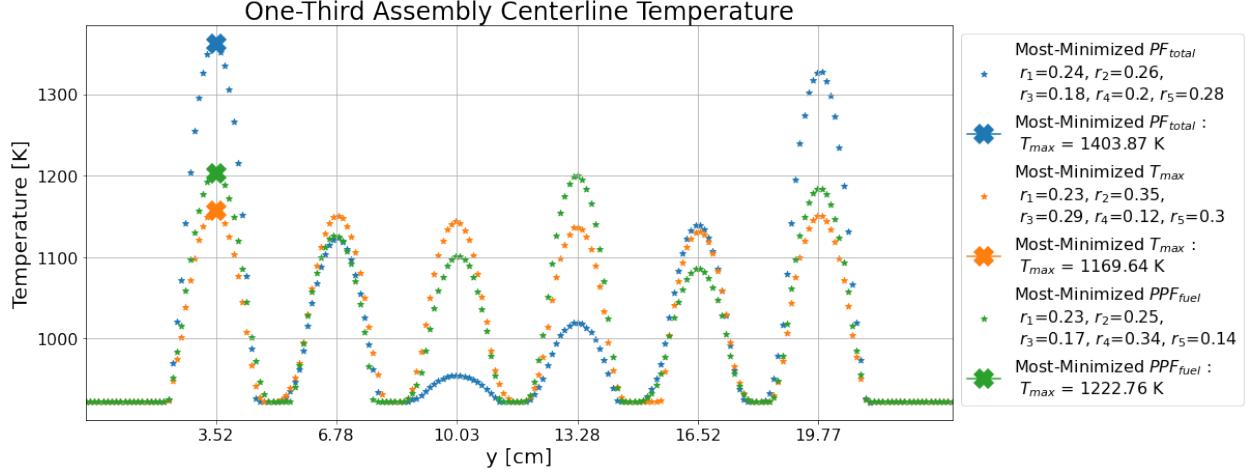


Figure B.2: (contd.) AHTR one-third assembly models and TRISO distributions for the 3 reactor models on simulation a-3b-256's Pareto front that most-minimized each objective, and 1 reactor model that equally minimized all three objectives. Simulation a-3b – ROLLO triple-objective optimization to minimize total fuel packing fraction (PF_{total}), maximum temperature (T_{max}), and fuel-normalized power peaking factor (PPF_{fuel}) in the one-third assembly. Input parameters varied: total fuel packing fraction PF_{total} , TRISO packing fraction distribution ($\rho_{TRISO}(\vec{r})$), coolant channel shape (r_1, r_2, r_3, r_4, r_5).

Figure B.3 shows the one-third assembly centerline temperatures for three reactors on simulation a-3b-356's Pareto front: reactor model 36 with most-minimized PF_{total} , reactor model 4 with most-minimized T_{max} , and reactor model 17 with most-minimized PPF_{fuel} . r_1 , r_2 , r_3 , r_4 , and r_5 values correspond to the FliBe channel at 18cm, 15cm, 12cm, 8cm, and 6cm, respectively.



(a) Centerline temperature. AHTR assembly's centerline is the white line in Figure 5.15.

Figure B.3: Simulation a-3b-256's one-third assembly reactor models' temperature distribution. Reactor models are on simulation a-3b-256's Pareto front: reactor model 36 with most-minimized PF_{total} , reactor model 4 with most-minimized T_{max} , and reactor model 17 with most-minimized PPF_{fuel} . r_1 , r_2 , r_3 , r_4 , and r_5 values correspond to the FliBe channel at 18cm, 15cm, 12cm, 8cm, and 6cm, respectively.

Figure B.3a shows that all three reactor models peak in the 1st graphite plank (at 3.52cm) with r_1 values of $\sim 0.23\text{cm}$. The larger radius values closer to temperature peaks enables lower T_{max} values in simulation a-3b-256 compared to simulation a-3b's equivalent reactor models (Figure 7.34a). This suggests that simulation a-3b-256's larger population size enabled ROLLO to explore more reactor model variations and find even more optimal reactor models that further minimized T_{max} . Future work includes running simulation a-3b-256 for even more generations.