Demand-Driven Cycamore Archetypes

Fuel Cycle Research and Development

Travis Knight
University of South Carolina

Collaborators
University of Illinois at Urbana-Champaign

Kenneth Keller, Federal POC
Ed Hoffman, Technical POC
Final Technical Report

Project 16-10512
Award DE-NE0008567
Demand Driven Cycamore Archetypes

Collaborators
Kathryn D. Huff
(University of Illinois at Urbana-Champaign)
Travis W. Knight
(University of South Carolina)

Contributors
Jin Whan Bae
Aditya Bhosale
Snehal Chandan
Gwendolyn J. Chee
Roberto E. Fairhurst
Robert R. Flanagan
Louis Kissinger
Gyu Tae Park
Anthony M. Scopatz

This research was performed using funding received from the DOE Office of Nuclear Energy’s Nuclear Energy University Program (Project 16-10512, DE-NE0008567) ‘Demand-Driven Cycamore Archetypes’. 
Executive Summary

Future nuclear fuel cycle options may present advantages over today’s once-through fuel cycle. Nuclear fuel cycle simulation tools assess the performance of those fuel cycles as well as the dynamics of long term technology transitions.

In many nuclear fuel cycle simulation tools, it has historically been the responsibility of the user to manually define facility deployment schemes and all facility parameters. While this is straightforward in simple fuel cycles, transitions from one fuel cycle to another can be more complex. In particular, deployment schemes for supportive fuel cycle facilities beyond the reactor become complex if the analyst desires to avoid gaps in the nuclear fuel and power supply chain during those transition scenarios. As nuclear fuel cycle analysis approaches questions regarding the feasibility and performance of the deployment schemes and technology choices during technology transition, automation of this historically manual process is necessary.

The main objective of this work was to develop and demonstrate Cyclus automation capabilities toward key nuclear fuel cycle transition scenarios. While deploying reactors to meet power demand is trivial, and existed in the earliest versions of CYCLUS, automated, predictive deployment and decommissioning of other facilities is more complex. These include mining, milling, enrichment, fuel fabrication, reprocessing, and others. For example, a balanced closed fuel cycle may require ensuring that there is enough fast reactor fuel for their operation and may drive deployment of a fleet of light water reactors. This concern comprises the main challenge that drove the project effort.

The Demand-Driven Cycamore Archetype project (NEUP-FY16-10512) aimed to develop CYCAMORE demand-driven deployment capabilities and thereby automate transition scenario definition. The developed software package, d3ploy, in the form of a CYCLUS Institution agent, deploys Facilities to meet the front-end and back-end demands of the fuel cycle. The University of South Carolina and the University of Illinois applied multiple algorithmic approaches to this challenge.

This project developed an in situ demand-driven development schedule calculation through non-optimizing, deterministic-optimizing, and stochastic-optimizing algorithms as CYCLUS archetypes and demonstrated these new archetypes in program-supporting fuel cycle transition scenarios. Both objectives were achieved. This report documents the results and deliverables obtained toward these achievements in detail.

The project began in October 2016 and ended on time and under budget in October 2019. The original collaborators included Principal
Investigator Anthony Scopatz (University of South Carolina) and Co-PI Prof. Kathryn Huff (University of Illinois at Urbana-Champaign.) Anthony Scopatz departed academia in 2017 and the PI-ship was transferred to Prof. Travis Knight at the University of South Carolina. Prof. Kathryn Huff remained the lead investigator at the collaborating institution, the University of Illinois, throughout the duration of the project.
Contents

1 Objectives ................................................. 6
  1.1 CYCLUS Overview ...................................... 6
  1.2 Predictive Algorithms ................................. 7
  1.3 Demonstrations ...................................... 9
  1.4 Project Approach ................................... 10
  1.5 Literature Review .................................. 13
    1.5.1 Effort Performed ................................ 13
    1.5.2 Accomplishments and Deliverables .............. 14
  1.6 Stop and Restart Capabilities ....................... 14
    1.6.1 Effort Performed ................................ 14
    1.6.2 Accomplishments and Deliverables .............. 14
  1.7 Rectify Non-algorithmic Capability Gaps ........... 14
    1.7.1 Effort Performed ................................ 14
    1.7.2 Accomplishments and Deliverables .............. 15
  1.8 Toolkit Additions .................................. 15
    1.8.1 Effort Performed ................................ 15
    1.8.2 Accomplishments and Deliverables .............. 15
  1.9 Design Numerical Experiments for Non-Optimizing Algo-
      rithms ............................................. 17
    1.9.1 Effort Performed ................................ 18
    1.9.2 Accomplishments and Deliverables .............. 18
  1.10 Create d3ploy ...................................... 18
    1.10.1 Effort Performed ................................ 18
    1.10.2 Accomplishments and Deliverables .............. 18
  1.11 Design Numerical Experiments for Deterministic Optimiz-
       ing Algorithms .................................. 19
    1.11.1 Effort Performed ................................ 20
    1.11.2 Accomplishments and Deliverables .............. 20
  1.12 Implement Deterministic-Optimizing Methods: ARCH ... 20
    1.12.1 Effort Performed ................................ 20
    1.12.2 Accomplishments and Deliverables .............. 21
  1.13 Implement Deterministic Optimizing Algorithms: GCAM . 21
    1.13.1 Effort Performed ................................ 21
    1.13.2 Accomplishments and Deliverables .............. 21
  1.14 Design Numerical Experiments for Stochastic Optimizing Al-
       gorithms ........................................... 21
    1.14.1 Effort Performed ................................ 22
    1.14.2 Accomplishments and Deliverables .............. 22
1.15 Implement Deterministic Optimizing Algorithms: MARKAL 22
  1.15.1 Effort Performed ............................................. 22
  1.15.2 Accomplishments and Deliverables ...................... 22
1.16 Add Additional Capabilities for Predictive Methods ...... 23
  1.16.1 Effort Performed ............................................. 23
  1.16.2 Accomplishments and Deliverables ...................... 23
1.17 Demonstrate Capabilities in Transition Scenarios ........ 24
  1.17.1 Effort Performed ............................................. 26
  1.17.2 Accomplishments and Deliverables ...................... 26
1.18 Publish Comparison Among Algorithms ...................... 26
  1.18.1 Effort Performed ............................................. 26
  1.18.2 Accomplishments and Deliverables ...................... 28

2 List of Publications 28
  2.1 Journal Articles ................................................ 28
  2.2 Technical Reports .............................................. 29
  2.3 Theses ............................................................ 29
  2.4 Conference Papers .............................................. 29
  2.5 Software Packages .............................................. 31
  2.6 Other ............................................................ 31

List of Figures

1 Gap in capability: The user must define when support facilities are deployed. Above, a user-defined deployment scheme. 11
2 Bridging the gap: Developed demand-driven deployment capability in CYCLUS. This capability is named d3ploy. Above, a demand-driven deployment scheme enabled with d3ploy. 11
3 CycMap of the historic Cyclus U.S nuclear fuel cycle simulation [35] ...................................................... 17
4 Power generated between 1971 and 2016 from the CYCLUS simulation ...................................................... 17
5 Power generated between 1971 and 2016 as published by the Nuclear Energy Institute (NEI) [34] ...................... 17
6 d3ploy logic flow at every timestep in CYCLUS [16]. In this figure $D_p$ represents predicted demand, $S_p$ represents predicted supply, and $U = S_p - D_p$. .............................. 19
7 French Transition into an SFR Fleet ........................... 24
Deployment of French Sodium-Cooled Fast Reactors (SFRs) and total installed capacity.

The simulated nuclear power deployment scheme relies on used nuclear fuel collaboration among nations. The historical operation of EU reactors is followed by the French transition to SFRs. The steep transition from 2040 to 2060 reflects the scheduled decommissioning of reactors built in the 1975-2000 era of aggressive nuclear growth in France.

Time dependent undersupply of commodities for different prediction methods for the EG01-23 Transition Scenario with Constant Power Demand. The size of each cross is based on the size of the undersupply. Fewer crosses on plot indicates the method is more successful at preventing undersupply of each commodity.

Time dependent undersupply of commodities for different prediction methods for the EG01-24 Transition Scenario with Linearly Increasing Power Demand. The size of each cross is based on the size of the under capacity. Fewer crosses on plot indicates the method is more successful at preventing under capacity of each commodity.
1 Objectives

The main objective of this work was to develop and demonstrate CYCLUS automation capabilities toward key nuclear fuel cycle transition scenarios. This section and its subsections will provide background information as well as a full statement of each sub-objective comprising this broad objective. Each objective description will be accompanied by a description of the effort performed and the accomplishments achieved.

1.1 CYCLUS Overview

Modeling of the nuclear fuel cycle (NFC) is often posed as a set of demands coupled with available technologies. Demands may be singular (“achieve 1% growth for total electricity production”) or multivariate (“achieve 1% growth and reach 10% uranium utilization”). Additionally, technologies may be constrained by the date on which they become deployable (“fast reactors first become available in 2050”). Most fuel cycle simulators approach such objective functions and constraints by wrapping realizations of the simulator in an external optimizer. This work aimed to bring demand and deployment decisions into the Nuclear Fuel Cycle (NFC) simulator itself, thereby simulating a more realistic process by which utilities, governments, and other stakeholders actually make facility deployment decisions.

In this work, we added in situ deployment scheduling capabilities to CYCLUS [30], the agent-based NFC simulator. Using these capabilities, we additionally evaluated various potential technology options via transition scenarios. CYCLUS simulations use three canonical agent categories, or archetypes. Facilities represent standard fuel cycle components such as reactors, repositories, mines, and others. Institutions represent agents that deploy facilities such as utility companies. Regions simulate countries and other geopolitical entities and can be used to establish tariffs, preferred trading partners, and enforce other top-level policy decisions.

Unlike in many other fuel cycle simulators, and because CYCLUS is agent-based, its regions and institutions have the agency to dynamically make and alter deployment decisions. Most importantly, each agent has ability to make their own predictions of the future based on current and past performance of their region and the performance of the entire simulation. This recursive prediction capability on each and every time step was an untapped feature of CYCLUS until this work was completed.

The region and institution archetypes produced here were to be devel-
oped as part of the CYCAMORE archetype library. These new archetypes will be a significant departure from the archetypes already in CYCAMORE. At first glance, the Growth Region may seem to implement some of the capabilities desired by a demand-driven system. However, the Growth Region has some critical limitations that make it unsuitable for the desired kind of simulations.

First, the Growth Region only is able to meet a growth curve for a specific prototype’s capacity. For certain demands, such as ensuring a certain quantity of fresh fuel, this poses no problem. For other demands, like “1% growth in annual electricity production,” then power would have to be shoehorned into a commodity even though it might not be a natural fit for the simulation at hand. However, for other demand types, such as “the amount of plutonium in a separations facility” or “waste heat,” the requirement of a tradeable commodity is too strict. The parameter with which the demand lies is derived from a commodity and is not itself a commodity. To generically handle demands, a new approach was needed.

Furthermore, the existing Growth Region has no integrated memory of past performance. It is therefore not able to make predictions about the future. It is only able to respond to current events according to a hard-coded, piecewise growth curve given at the start of the simulation. Moreover, the Growth Region is not able to adjust to the qualifiers for the curve it receives. For example, certain simulations may find it desirable to achieve at least 1% growth, allowing for overshooting this curve but never allowing to fall below it. Lastly, and perhaps most significantly, the Growth Region has no capability to deploy supporting fuel cycle facilities that enable a demand to be met. Take for instance the standard once through fuel cycle. Reactors may be deployed to meet a certain power demand. However, new mines and fuel fabrication facilities will also need to be deployed to ensure that reactors have sufficient fuel to run once they begin to produce power. The current and unrealistic solution to this problem is to simply have infinite capacity support facilities. This neglects one of the major advantages of CYCLUS, which is its ability to model, deploy, and track discrete facilities, such as those new facilities needed to meet capacity demands. The capabilities developed as part of this work leveraged these advantages in the deploy tool.

1.2 Predictive Algorithms

Many different algorithms and approaches may be taken to generate or adjust deployment schedules to meet a given demand. Since in situ
demand-driven deployment decisions have not been made in an NFC simulator before this work, the relative merits (performance characteristics, precision, accuracy) of different algorithms were not known a priori. This work, therefore, built out a few representative implementations that cover the space of possible algorithms. In our approach, demand-driven deployment models can be grouped into three categories: non-optimizing (NO), deterministic-optimizing (DO), and stochastic-optimizing (SO).

Non-optimizing algorithms are defined as those that infer future deployment schedules solely from historical (to the simulator) supply-demand data. Notably, they do not run sub-simulations into the future and attempt to minimize the difference of the produced quantity to the demand. It is anticipated that these algorithms will have the fastest execution times, if perhaps also being the least precise. Algorithms that we propose to examine include Autoregressive Moving Average (ARMA)\[33, 42\] and Autoregressive Heteroskedasticity (ARCH) \[12, 40\] methods. Both ARMA and ARCH rely on an autoregressive model. This means that the predicted future values of a time series depend on the previous values of that same time series. ARMA applies a moving average model to the time series data as well. Adding the moving average aims to reduce statistical noise or shocks to the system. The ARMA method can be represented by an autoregressive term and a moving average term.

\[
X_i = c + \epsilon_t + \sum_{i=1}^{p} \theta_i X_{t-1} + \sum_{j=1}^{q} \phi_j \epsilon_{t-j} \quad (1)
\]

Here \(X_i\) is the next predicted value of the time series, \(c\) is an adjustable constant, \(\theta_i\) represents the \(i^{th}\) parameter, \(j\) represents the \(j^{th}\) parameter of the moving average system, and \(\epsilon\) represents white noise.

ARCH models use the previous variance terms to help calculate current error terms. This allows the method to determine the behavior of a time series that experiences periods of high volatility followed by periods of calm (and visa versa). ARCH expands on the AR method and specifically replaces the white noise error term with the following equation.

\[
\epsilon_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-1}^2 \quad (2)
\]

Here \(\alpha_i\) represents the \(i^{th}\) parameter of the conditional heteroskedastic model.
Deterministic optimization methods were also investigated. These methods, including both the Global Change Assessment Model (GCAM) [23] and MARKAL [24] were investigated. These methods are used to perform time series analysis on a wide variety of topics including economics, climate, and the energy sector. In this work, we have used linear programming models in conjunction with optimization functions for demand curve prediction.

Stochastic optimization techniques are the third group of methods that will be reviewed and implemented. Markov modeling processes work on time sets of data in which future events are independent of past events. This means that the markov process can predict the future state of a system, given just the current state. An example of these prediction methods is the Markov Switching model [1]. This model is composed of two equation types; measurement and transition.

Measurement equations determine how hidden states affect observed data, and transition equations define how the states evolve over time. A simple switching method can be represented with the following equations.

\[
y_t = \alpha_{0,0} + \alpha_{0,1}S_t + (\alpha_{1,0} + \alpha_{1,1}S_t)y_{t-1} + e_i
\]

Here \(y_t\) is the observable random variable being investigated, \(\alpha\) represents a drift term constant, \(S_t\) is a hidden state variable, \(e_i\) represents an error term. In this model, and \(P_{ij} = P(S_t = j|S_{t-1} = i)\) represents the probability of a transition in the hidden state variable.

Another stochastic method considered was Gaussian Process Regression. Such models assume a functional form for mean and the covariance functions. The hyperparameters of these forms are then fit to the historical data, which serves as a ‘training set’ in machine learning parlance. The mean function may then be used to predict the behavior of a system by evaluating at future points in simulation time.

1.3 Demonstrations

As this demand-driven deployment capability was added to CYCLUS, this work demonstrated via technology-level analyses of variations on four promising fuel cycle families identified by the U.S. Department of Energy (DOE) “Evaluation and Screening Report” [41]. Future efforts can build on this work by further evaluating the engineering design of fuel cycle facilities, the impact of technology choices on evaluation metrics, and expand
on conclusions generated in the evaluation and screening. This will utilize the available data in the Evaluation and Screening database. Possible further work may include similarly detailed comparative analysis of the fourteen potentially promising fuel cycles identified as by the Evaluation and Screening report as less promising.

This analysis will require fuel cycle simulation using a next generation fuel cycle simulator, such as CYCLUS, which is capable of flexible comparisons and facility-level customization and analysis. As a demonstration of the demand-driven implementation, this effort proposes to evaluate the impact of both fast and thermal critical reactor technologies and their associated reprocessing facilities within the following four evaluation groups:

EG23: Continuous recycle of U/Pu with new natural-U fuel in fast critical reactors

EG24: Continuous recycle of U/TRU with new natural-U fuel in fast critical reactors

EG29: Continuous recycle of U/Pu with new natural-U fuel in both fast and thermal critical reactors

EG30: Continuous recycle of U/TRU with new natural-U fuel in both fast and thermal critical reactors.

The evaluation was based on the nine DOE-specified evaluation criteria, but sought to capture a greater range of insights at the facility technology level than were possible in the static and transition analyses conducted for the evaluation and screening. This detail included, but was not limited to, various options for reprocessing strategies, and alternative reactor design considerations at the single-facility technology level.

Notably, the evaluation groups under consideration (such as those proposed above) are subject to the input and needs of the sponsor. We will endeavor to work with the technical point of contact to identify future analysis needs and adjust the target and scope of this work as appropriate.

1.4 Project Approach

Because CYCLUS is Agent-Based, its regions and institutions have the agency to dynamically make and alter deployment decisions. Also, each agent can make their own predictions of the future based on current and past performance of the simulation. Accordingly, we embedded advanced
time series prediction algorithms to automatically deploy fuel cycle facilities for the user. This was implemented in d3ploy, an Institution agent.

Figure 1: Gap in capability: The user must define when support facilities are deployed. Above, a user-defined deployment scheme.

Figure 2: Bridging the gap: Developed demand-driven deployment capability in CYCLUS. This capability is named d3ploy. Above, a demand-driven deployment scheme enabled with d3ploy.

Technical deliverables, including publications affiliated with this work have been submitted along with previous quarterly reports and are summarized below.

Journal Articles 4 (+1 in review)

Full Conference Papers 5

Conference Summaries 7

Technical Reports 3

Theses 2 MS (+1 upcoming)

Educational deliverables included the support of graduate students and occasional undergraduates at UIUC and USC. These included transitions of skilled students to national laboratories after graduation. UIUC student
Jin Whan Bae received his MS and is now at Oak Ridge National Laboratory (ORNL) pursuing CYCLUS usability. University of Illinois at Urbana-Champaign (UIUC) student Gwendolyn Chee has completed an MS thesis related to this work and work conducted at Argonne National Laboratory (ANL) with Dr. Bo Feng. UIUC undergraduate Louis Kissinger participated in this project and spent a year as a baccalaureate researcher in 2018-2019 in the Mathematics and Computational Sciences division at ANL. Other students supported include Roberto Fairhurst, Gyu Tae Park, Snehal Chandan, and Aditya Bhosale. Finally, postdoctoral scholar Robert Flanagan was supported by this work for the duration of the project at University of South Carolina (USC).

Detailed objectives were developed in collaboration with Technical Point of Contact Edward Hoffman (ANL). Those detailed sub-objectives are listed below and have each been accomplished.

✓ Literature review of appropriate predictive algorithms.
✓ Add stop and restart capabilities to CYCLUS (bonus: HPC deployment addition).
✓ Identify and rectify non-algorithmic capability gaps (e.g. specific fuel cycle process archetypes) necessary for transition simulation.
✓ Create d3ploy, a demand-driven deployment software capability for CYCLUS.
✓ Add toolkit additions related to geospatial information.
✓ Design numerical experiments to test non-optimizing algorithms.
✓ Implement non-optimizing (NO) methods in d3ploy.
✓ Design numerical experiments (tests) for verifying Deterministic-Optimizing (DO) algorithms in the context of key transitions.
✓ Implement Deterministic Optimizing (DO) methods in d3ploy.
✓ Design numerical experiments (test) for verifying Stochastic-Optimizing (SO) algorithms in the context of key transitions.
✓ Implement Stochastic Optimizing (SO) methods in d3ploy.
✓ Add additional capabilities to the predictive methods. (Buffers, reprocessing complexity handing)

✓ Demonstrate and compare the new capability in the context of the Fuel Cycle Evaluation and Screening evaluation groups (EG) 23, 24, 29, 30.

Each item in the detailed project objectives above, and associated accomplishments, will be described in the following subsections.

1.5 Literature Review

By the end of Quarter 1 (month 3), the first deliverable of Phase 1, Thrust 1, a literature review, was due. In this task, led by UIUC a completed literature review of previous methods for developing predictive algorithms was delivered. This review included non-optimizing and discrete optimization methods. This effort focused on ARMA, ARCH, GCAM and MARKAL. Adding stochastic optimization methods (specifically Gaussian processes) to the literature review was the responsibility of USC, specifically aiming at how these algorithms can be leveraged with respect to nuclear specific metrics.

1.5.1 Effort Performed

Students and faculty reviewed literature from other fields including process engineering, control systems, energy analysis, and others. We found that automated deployment of supportive fuel cycle facilities is naïve or non-existent in most simulators, including CYCLUS, the simulator under development by the authors. Additionally, for the majority of simulators, automated deployment is limited to deploying reactors based on changes in power demand.

This study identified flexible, general, and performant algorithms available for application to simulating demand-driven deployment of nuclear fuel cycle facility capacity in a fuel cycle simulator. Accordingly, a review of current Nuclear Fuel Cycle (NFC) simulation tools was conducted to determine their current capabilities for demand-driven and transition scenarios. Additionally, we investigated promising algorithmic innovations that have been successful for similar applications in other domains such as economics and industrial engineering. Finally, the applicability of such algorithms in the context of challenging nuclear fuel cycle simulation questions was described.
1.5.2 Accomplishments and Deliverables

An abstract for the work was submitted, accepted, expanded into a full paper, and presented at the 2017 Global Nuclear Fuel Cycle Conference [31].

1.6 Stop and Restart Capabilities

By the end of Quarter 2 (month 6), the first deliverable of Phase 1, Thrust 2 was due. This sub-task required software development, led by USC to update CYCLUS in order to have start and stop capabilities. This involved updating the CYCLUS engine to allow for user-driven time periods for stopping a CYCLUS simulation for a variety of reasons. These include data analysis, simulation forking, decision making, and reproducibility.

1.6.1 Effort Performed

Software development in this context was led by a faculty member and a postdoctoral scholar at USC.

1.6.2 Accomplishments and Deliverables

Start-stop capabilities were implemented in the CYCLUS API. The Rickshaw toolkit can be found on github.

1.7 Rectify Non-algorithmic Capability Gaps

By the end of Quarter 3 (month 9), the second sub-task of Task 1.2 was due. This sub-task was led by UIUC. It involved test-running target transition analyses (EG 23, 24, 29, and 30) with existing CYCLUS & CYCAMORE capabilities to identify and rectify non-algorithmic capability gaps (e.g. specific fuel cycle process archetypes) necessary for complete simulation. Once identified, these gaps in capability were rectified.

1.7.1 Effort Performed

This effort involved open, transparent development of software (e.g. rickshaw from USC and geospatial information system (GIS) toolkit addition from UIUC) and transparent sharing of simulation data and results (e.g. preliminary transition scenario input files from UIUC). Simulations of transition scenarios were performed for all four of the “most promising”
fuel cycles. This process provided an excellent example of open science to peer researchers in other domains.

1.7.2 Accomplishments and Deliverables

Software capabilities were developed and are available on GitHub within the CYCLUS repository and a repository holding the transition scenario input and output files was also developed [8]. A milestone report was published to describe these capability gaps [4].

1.8 Toolkit Additions

By the end of quarter 3 (month 9), the first deliverable of Task 1.3 was due. Expand the CYCLUS toolkit to include an API that will allow agents to extract historical and current data from the Dynamic Resource Exchange. This API will allow for the ability to filter by commodity and archetype. Additionally, UIUC contributed toolkit additions related to geospatial information. On this topic, two papers were submitted, accepted, and presented to the GLOBAL nuclear fuel cycle conference in Seoul, South Korea [26, 38].

1.8.1 Effort Performed

The Rickshaw tool was developed to push the state-of-the-art of fuel cycle simulation by providing a standard and efficient way to execute common transition studies. While still offering the previous capabilities of parameter sweeps and stochastic searches. This included expanding the CYCLUS ecosystem to be able to execute and manage large numbers of simulations (up to millions to billions) and was supported by undergraduates at UIUC as well as researchers at USC.

A geospatial toolkit addition was also developed by a UIUC undergraduate.

1.8.2 Accomplishments and Deliverables

The new “rickshaw” code base capable of stochastically generating valid CYCLUS input files. Rickshaw is also able to run these input files and merge the resultant output files. This is needed to create training data sets for future stochastic optimizations studies. Documentation for Rickshaw can be
found at http://www.ergs.sc.edu/rickshaw-docs/. On this topic, two papers were submitted, accepted, and presented to the GLOBAL nuclear fuel cycle conference in Seoul, South Korea [38, 26].

The geospatial toolkit addition enabled a unique simulation that demonstrated the past and current the United States fuel cycle. A CYCLUS simulation of the U.S. nuclear fuel cycle was created using historic United States reactor deployment data obtained from the Power Reactor Information System (PRIS) database (via the International Atomic Energy Agency (IAEA)). In this simulation, facilities present in the simulation included a mine, mill, enrichment plant, fuel fabrication facility, 112 historic commercial reactors in the U.S, a dry storage facility, and a final waste repository. Additional assumptions included:

- **Refueling time**: 1 month
- **Cycle length**: 18 months
- **Single Spent Fuel Recipe**: 33 or 51 GWDt/MTU burnup (depletion calculations done using ORIGEN)
- **Assembly size, Core size, Batch size**: dependent on the reactor type
- **Power capacity, Location**: specific to each reactor from PRIS data

The four transition scenarios were simulated with optimization parameters seeking to minimize undersupply and under capacity of all commodities.

1. EG01-23 \( (P(t) = P_0) \)
2. EG01-24 \( (P(t) = P_0 + rt) \)
3. EG01-29 \( (P(t) = P_0) \)
4. EG01-30 \( (P(t) = P_0 + rt) \)

This was achieved with the following process:

1. Comparison of prediction methods for each of 4 scenarios is conducted to determine the best method.
2. Sensitivity analysis of power supply buffer is conducted to determine best buffer size.
3. Using best prediction method, look ahead rate, buffer size, demonstrate deploying reactor and supporting facilities to meet power demand for 4 scenarios.

The results were visualized in figures 3, 4, and 5.

Figure 3: CycMap of the historic Cyclus U.S nuclear fuel cycle simulation [35]

Figure 4: Power generated between 1971 and 2016 from the CYCLUS simulation

Figure 5: Power generated between 1971 and 2016 as published by the NEI [34]

These results, and others, were presented at the ANS student conference by undergraduates at UIUC.

1.9 Design Numerical Experiments for Non-Optimizing Algorithms

Due in Quarter 3 (month 9), Task 1.3 was led by UIUC. This task aimed to design numerical experiments for validating Non-Optimizing (NO) algorithms in the context of the EG 24, 24, 29, 30. These were be implemented as unit and integration tests.
1.9.1 Effort Performed

Expected performance of the implemented algorithms were defined in toy problems. These toy problems comprised numerical experiments (tests) against which the algorithm performance would be measured once implemented.

1.9.2 Accomplishments and Deliverables

The designed experiments were implemented as unit tests in the d3ploy codebase and were coordinated with our continuous integration system. A report was submitted to Nuclear Energy University Programs (NEUP) describing this process [6].

1.10 Create d3ploy

Due in Quarter 5 (month 15), the first sub-task in Phase/Thrust 2.4a was due. This task aimed to develop a CYCLUS module that implements the ARMA method in the d3ploy model. It fundamentally required the creation of d3ploy, as the ARMA method was the first predictive module to be added to d3ploy. This module performed demand prediction of the amount of material that will be available in the future. Additionally, it provided uncertainties for those predictions. It was used to demonstrate decision making on when to deploy new facilities to meet demand and supply curves.

1.10.1 Effort Performed

The ARMA method was implemented in d3ploy until it passed the non-optimizing tests developed in the previous sub-task. Significant software development on the part of both UIUC and USC was performed collaboratively, transparently, and reproducibly, to achieve this accomplishment.

1.10.2 Accomplishments and Deliverables

The ARMA method was implemented in d3ploy and the software tool, d3ploy was released. A paper on this topic “Using Autoregressive Moving Average Method to Determine the Deployment of Nuclear Fuel Cycle Facilities” [25] was presented at PHYSOR 2018: Reactors Physics paving the way towards more efficient systems in Cancun, Mexico, April 22-26, 2018. The logic of d3ploy appears below in 6:
1.11 Design Numerical Experiments for Deterministic Optimizing Algorithms

Due in Quarter 6 (month 18), subtask 2.4a included the design of numerical experiments for verifying Deterministic-Optimizing (DO) algorithms in the context of the “most promising” fuel cycles. These were evaluation groups (EG) 23, 24, 29, 30. These were implemented as unit and integration
tests. UIUC was responsible for leading this effort.

1.11.1 Effort Performed

Expected performance of the implemented deterministic optimizing algorithms were defined in toy problems. These toy problems emphasized capabilities that deterministic optimizing methods would presumably achieve beyond the capabilities of the non-optimizing methods. The toy problems comprised numerical experiments (tests) against which the algorithm performance would be measured once implemented.

1.11.2 Accomplishments and Deliverables

The designed experiments were implemented as unit tests in the deploy codebase and were coordinated with our continuous integration system. A report was submitted to NEUP describing this process [6] and students presented this work at the 2018 American Nuclear Society Student Conference. In support of this work, an undergraduate at UIUC developed reactor depletion recipes for sodium cooled fast reactors and presented those to the ANS student conference as well. All of these results can be found online at https://github.com/arfc. Additionally, Prof. Huff presented this work at the University of Michigan Nuclear Engineering and Radiological Sciences Colloquium.

1.12 Implement Deterministic-Optimizing Methods: ARCH

Due in Quarter 6 (month 18), sub-task 2.4b was led by USC. Like 2.1a, this task saw the release of an Auto-Regressive Conditional Heterodastic method module used to predict the future behavior of demand and supply curves of the active simulation. Due to the reliance of the ARCH method on conditionals, this milestone also investigated how these conditionals impact the behavior of the system.

1.12.1 Effort Performed

The ARCH method was implemented in deploy until it passed the non-optimizing tests developed in the previous sub-task. Significant software development on the part of both UIUC and USC was performed collaboratively, transparently, and reproducibly, to achieve this accomplishment.
1.12.2 Accomplishments and Deliverables

The ARCH method was implemented in d3ploy and the software tool, d3ploy was released.

1.13 Implement Deterministic Optimizing Algorithms: GCAM

In Quarter 7 (month 21) the first portion of sub-task 2.5a was due. This was led by the team at USC. This milestone aimed to incorporate the GCAM system into a Cyclus module. To accomplish this, a Cyclus module that makes external calls to the GCAM software was to be investigated. If feasible, this capability was to be developed and demonstrated. Using Cyclus to make external calls to an outside software through a module had never been explicitly attempted so far. As such part of this milestone emphasized testing how amenable the Cyclus engine is to this behavior and how best to accomplish this.

1.13.1 Effort Performed

While incorporating GCAM, specifically, was infeasible due to its restrictive API, similar external modules from the scikit-learn library \[36\], a machine learning library in python, were wrapped into this tool.

1.13.2 Accomplishments and Deliverables

The incorporation of external library wrapping was presented to the 2018 Technical Workshop on Fuel Cycle Simulation in Paris, France by postdoctoral scholar Robert Flanagan in the talk “Deployment of Fuel Cycle Facilities Using Supply and Demand Curves.”

1.14 Design Numerical Experiments for Stochastic Optimizing Algorithms

Due in quarter 8 (month 24), the second portion of sub-task 2.5a was due. This subtask required the design of numerical experiments for verifying Stochastic-Optimizing (SO) algorithms in the context of the EG 23, 24, 29, 30 transitions. These were implemented as unit and integration tests.
1.14.1 Effort Performed

Expected performance of the stochastic optimizing algorithms were defined in toy problems before the implementation of those algorithms. These toy problems emphasized capabilities that stochastic optimizing methods would presumably achieve beyond the capabilities of the deterministic optimizing methods. The toy problems comprised numerical experiments (tests) against which the algorithm performance would be measured once implemented.

1.14.2 Accomplishments and Deliverables

The designed experiments were implemented as unit tests in the deploy codebase and were coordinated with our continuous integration system. A report was submitted to NEUP describing this process [6] and students presented this work at the 2018 American Nuclear Society Student Conference[14]. In support of this work, an undergraduate at UIUC developed reactor depletion recipes for sodium cooled fast reactors and presented those to the ANS student conference as well [32]. All of these results can be found online at https://github.com/arfc.

1.15 Implement Deterministic Optimizing Algorithms: MARKAL

In quarter 8 (month 24), sub-task 2.5b was due. In this task, stochastic optimizing algorithms were to be implemented and used. This effort was to be led by USC. However, similar to the infeasibility of GCAM, the MARKAL API did not allow this integration. Alternative, more modern external tools incorporating the same deterministic optimizing algorithms were incorporated instead.

1.15.1 Effort Performed

An effort was made at USC to integrate MARKAL, but its API was incompatible with wrapping in this manner. Instead, equivalent algorithms were implemented via the alternative external libraries in scikit-learn.

1.15.2 Accomplishments and Deliverables

The incorporation of external library wrapping was presented to the 2018 Technical Workshop on Fuel Cycle Simulation in Paris, France by post-

1.16 Add Additional Capabilities for Predictive Methods

In quarter 10 (month 30), sub-task 2.6b was due. While in the initial plan for the work, the lead institution for this subtask was to be USC, changes in the leadership meant that UIUC would be leading all remaining sub-tasks. So, UIUC lead this task, which included adding additional capabilities to the predictive methods. While supply and demand of materials are the most recognizable types of curves to optimize against in a fuel cycle, it is possible to optimize against other fuel cycle metrics as well; nonproliferation, facility down time, economics, etc.

Additionally, realistic models of national decision-making and policies were investigated for implementation as parameters and preferences in the algorithm configurations.

1.16.1 Effort Performed

In this effort, a journal article on synergistic political collaboration in the European Union was published which required significant additions such as capacity buffers, facility technology preferences, and inter-regional trade constraints. These resulted in deploy configuration parameters as reported in Table 1.

1.16.2 Accomplishments and Deliverables

Such realistic modeling capabilities for national policy and corporate utility behavior were incorporated through software development associated with the capacity gaps identified in the early stages of the deploy package and were demonstrated in the paper [10].

In this work, the following assumptions were made during the simulation.

- Collaborative spent fuel management is materially feasible among nuclear nations in the European Union.
- Collaborative European Union (EU) spent fuel management could expedite a fast reactor technology transition in France.
- By using spent fuel from other EU nations, France can avoid building new light water reactors to support a transition to fast reactors.
**Table 1:** `deploy`'s required and optional input parameters with examples.

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required</strong></td>
<td></td>
</tr>
<tr>
<td>Demand driving commodity</td>
<td>Power, Fuel, Plutonium, etc.</td>
</tr>
<tr>
<td>Demand equation</td>
<td>$P(t) = 10000, \sin(t), 10000 \cdot t$</td>
</tr>
<tr>
<td>Facilities it controls</td>
<td>Fuel Fab, LWR reactor, SFR reactor, Waste repository, etc.</td>
</tr>
<tr>
<td>Capacities of the facilities</td>
<td>3000 kg, 1000 MW, 50000 kg</td>
</tr>
<tr>
<td>Prediction method</td>
<td>Power: fast fourier transform</td>
</tr>
<tr>
<td></td>
<td>Fuel: moving average</td>
</tr>
<tr>
<td></td>
<td>Spent fuel: moving average</td>
</tr>
<tr>
<td>Deployment driven by</td>
<td>Installed Capacity/Supply</td>
</tr>
<tr>
<td><strong>Optional</strong></td>
<td></td>
</tr>
<tr>
<td>Supply/Capacity Buffer type</td>
<td>Absolute</td>
</tr>
<tr>
<td>Supply/Capacity Buffer size</td>
<td>Power: 3000 MW</td>
</tr>
<tr>
<td></td>
<td>Fuel: 0 kg</td>
</tr>
<tr>
<td></td>
<td>Spent fuel: 0 kg</td>
</tr>
<tr>
<td>Facility preferences</td>
<td>LWR reactor = 100-t</td>
</tr>
<tr>
<td></td>
<td>SFR reactor = t-100</td>
</tr>
<tr>
<td>Facility constraint</td>
<td>SFR reactor constraint = 5000kg of Pu</td>
</tr>
</tbody>
</table>

Figure 8 and 7 show the deployment of just SFRs and total installed reactor capacity. Meanwhile, Figure 9 includes the deployments for all nations.

This work tangentially assisted in other DOE-funded work, which resulted in the collaborative paper regarding molten salt reactor fuel cycles [37].

### 1.17 Demonstrate Capabilities in Transition Scenarios

In quarter 11 (month 33), subtask 3.7, the final phase was focused on the performance of the demand driven models in the context of the eval-
Figure 9: The simulated nuclear power deployment scheme relies on used nuclear fuel collaboration among nations. The historical operation of EU reactors is followed by the French transition to SFRs. The steep transition from 2040 to 2060 reflects the scheduled decommissioning of reactors built in the 1975-2000 era of aggressive nuclear growth in France.
ulation groups. This was to include a comparison to the “Evaluation and Screening” study.

1.17.1 Effort Performed

A benchmark evaluation was performed against the transition scenario benchmark developed by the Fuel Cycle Options campaign. Transition scenarios were run, using the algorithms in d3ploy, and compared to the benchmark. CYCLUS performed very well.

1.17.2 Accomplishments and Deliverables

A benchmarking effort was published in [9].

1.18 Publish Comparison Among Algorithms

At the end of the project (month 36) the milestone required a peer reviewed journal article on the results of the comparison study between the several modules detailed in the proposal.

1.18.1 Effort Performed

A sensitivity based analysis comparing the various algorithms with respect to performance (number of timesteps with insufficient material, smoothness of transition, etc.) were performed on the four evaluation groups. That comparison covered the following methods:

- Non-Optimizing Methods
  - Moving Average (ma)
  - Autoregressive Moving Average (arma)
  - Autoregressive Heteroskedasticity (arch)

- Deterministic-Optimizing Methods
  - Fast Fourier Transform (fft)
  - Polynomial Fit (poly)
  - Exponential Smoothing
  - Triple Exponential Smoothing (holt-winters)
Stochastic-Optimizing Methods

- Auto-Regressive Integrated Moving Averages (ARIMA)

Understanding and insight, regarding the appropriate algorithms for various tasks were identified. Some of these results appear in Figures 10 and 11.

Figure 10: Time dependent undersupply of commodities for different prediction methods for the EG01-23 Transition Scenario with Constant Power Demand. The size of each cross is based on the size of the undersupply. Fewer crosses on plot indicates the method is more successful at preventing undersupply of each commodity.

Figure 11: Time dependent undersupply of commodities for different prediction methods for the EG01-24 Transition Scenario with Linearly Increasing Power Demand. The size of each cross is based on the size of the under capacity. Fewer crosses on plot indicates the method is more successful at preventing under capacity of each commodity.
1.18.2 Accomplishments and Deliverables

UIUC led this effort and a journal article has been submitted for review. The best performing prediction method for each transition scenario was found to be:

1. EG01-23 Constant Power Demand: poly
2. EG01-24 Linearly Increasing Power Demand: fft
3. EG01-29 Constant Power Demand: poly
4. EG01-30 Linearly Increasing Power Demand: fft

2 List of Publications

The following list summarizes the publications associated with this work. Key publications and reports follow as appendices.

2.1 Journal Articles


2.2 Technical Reports


2.3 Theses


2.4 Conference Papers


2.5 Software Packages

Software released as a part of this work has included many packages. Most are listed below. Other minor packages can be found at github.com/arfc, github.com/ergs, or github.com/cyclus.


• G. Chee. gwenchee/ddwrapper : Gwen’s MS Thesis Release, 2019

• G. Chee, G. Westphal, and K. Huff. arfc/dcwrapper : Gwen’s MS Thesis Release, 2019


2.6 Other


**References**


