Demand Driven Deployment Capabilities in Cyclus, a Fuel Cycle Simulator

Gwendolyn J. Chee,[*,a] Roberto E. Fairhurst Agosta,[a] Jin Whan Bae,[b] Robert R. Flanagan,[c] Anthony M. Scopatz,[d] and Kathryn D. Huff[a]

[a] *Dept. of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801*

[b] *Oak Ridge National Laboratory, Oak Ridge, TN, United States*

[c] *Nuclear Engineering Program, University of South Carolina*

[d] *Quansight, LLC*

[*] Email: gchee2@illinois.edu

Number of pages: 39
Number of tables: 7
Number of figures: 12

Abstract

The present United States' nuclear fuel cycle faces challenges that hinder the expansion of nuclear energy technology. The U.S. Department of Energy identified four nuclear fuel cycle options, which make nuclear energy technology more desirable. Successfully analyzing the transitions from the current fuel cycle to these promising fuel cycles requires a nuclear fuel cycle simulator that can predictively and automatically deploy fuel cycle facilities to meet user-defined power demand. This work introduces and demonstrates demand-driven deployment capabilities in Cyclus, an open-source nuclear fuel cycle simulator framework. User-controlled capabilities such as time series forecasting algorithms, supply buffers, and facility preferences were introduced to give users tools to minimize power undersupply in a transition scenario simulation. The demand-driven deployment capabilities are referred to as `d3ploy`. We demonstrate `d3ploy`'s capability to predict future commodities' supply and demand, and automatically deploy fuel cycle facilities to meet the predicted demand in four transition scenarios. Using `d3ploy` to set up transition scenarios saves the user simulation set-up time compared to previous efforts that required a user to manually calculate and use trial and error to set up the deployment scheme for the supporting fuel cycle facilities.

Keywords — nuclear engineering, nuclear fuel cycle, nuclear fuel cycle simulator, time series forecasting, automated deployment

I. INTRODUCTION

The nuclear fuel cycle represents the nuclear fuel life cycle from initial extraction through processing, use in reactors, and, eventually, final disposal. This complex system of facilities and mass flows collectively provide nuclear energy in the form of electricity [1]. Nuclear fuel cycle simulator tools were introduced to investigate nuclear fuel cycle dynamics at a local and global level. These simulators track the flow of materials through the nuclear fuel cycle, from enrichment to final disposal of the fuel, while also accounting for decay and transmutation of isotopes. The impacts are evaluated in the form of 'metrics', quantitative measures of performance [2]. These metrics are calculated from mass balances and facility operation histories calculated by a fuel cycle simulator [2]. By evaluating performance metrics of different fuel cycles, we gain an understanding of how each facility's parameters and technology choices impact the system's performance. Therefore, these results can be used to guide research efforts, advise future design choices, and provide decision-makers with a transparent tool for evaluating fuel cycle options to inform policy decisions [1].

Many fuel cycle simulators automatically deploy reactor facilities to meet a user-defined power demand. However, the user must define a deployment scheme of supporting facilities to avoid gaps in the supply chain resulting in idle reactor capacity. Current simulators require the user to set infinite capacity for supporting facilities but this inaccurately represents reality and obfuscates required capacities. Manually determining a deployment scheme for a once-through fuel cycle is straightforward, however, for complex fuel cycle scenarios, it is not. To ease setting up realistic nuclear fuel cycle simulations, a nuclear fuel cycle simulator must bring dynamic demand-responsive deployment decisions into the simulation logic [3]. This means the nuclear fuel cycle simulator decides how many mines, mills, enrichment facilities, reprocessing facilities, etc are deployed to support dynamically changing power demand and reactor types. Thus, a next-generation nuclear fuel cycle simulator must predictively and automatically deploy fuel cycle facilities to meet a user-defined power demand.

I.A. Context of Work

The impact of climate change on natural and human systems is increasingly apparent [4]. The production and use of energy contribute to two-thirds of the total greenhouse gas (GHG)

Fuel Cycle	Open or Closed	Fuel Type	Reactor Type
EG01 (current)	Open	Enriched-U	Thermal
EG23	Closed	Recycled U/Pu + Natural-U	Fast
EG24	Closed	Recycled U/TRU + Natural-U	Fast
EG29	Closed	Recycled U/Pu + Natural-U	Fast & Thermal
EG30	Closed	Recycled U/TRU + Natural-U	Fast & Thermal

TABLE I
Descriptions of the current and other high performing nuclear fuel cycle evaluation groups described in the evaluation and screening study [6].

emissions [4]. Furthermore, as the human population increases and previously under-developed nations rapidly industrialize, global energy demand is forecasted to increase. Energy generation technology selection profoundly impacts climate change via growing energy demand. Large scale deployment of emissions free nuclear power plants could significantly reduce GHG production [4].

However, large scale nuclear power deployment faces challenges of safety, cost, and used nuclear fuel [5]. The nuclear power industry must overcome these challenges to ensure continued global use and expansion of nuclear energy technology.

The challenges described above are associated with the present once-through fuel cycle in the United States (US), in which fabricated nuclear fuel is used once and placed into storage to await disposal. An evaluation and screening study of a comprehensive set of nuclear fuel cycle options [6] was conducted to assess for promising evaluation groups (EGs) with performance improvements compared with the existing once-through fuel cycle (EG01) in the US across a wide range of criteria. Fuel cycles that involved continuous recycling of co-extracted U/Pu or U/TRU in fast spectrum critical reactors consistently scored high on overall performance based on the nine DOE-specified evaluation criteria: nuclear waste management, financial risk and economics, proliferation risk, nuclear material security risk, safety, environmental impact, resource utilization, development and deployment risk, and institutional issues [6]. Table I describes these fuel cycles: EG23, EG24, EG29, and EG30. Recent statements from Rita Baranwal [7], the Nuclear Energy Innovation Capabilities Act [8], and the Advanced Nuclear Technology Development Act [9] show that there continues to be national interest in pursuing spent fuel recycling and advanced nuclear power technology.

The evaluation and screening study assumed the nuclear energy systems were at equilibrium to understand the end-state benefits of each evaluation group [10]. In the current work, our goal is to model the transition from the initial EG01 state to these promising future end-states without assuming equilibrium fuel cycles. To successfully analyze time-dependent transition scenarios, the nuclear fuel cycle simulator tool must automate the transition scenario simulation setup. Therefore, the Demand-Driven Cycamore Archetypes project (NEUP-FY16-10512) was initiated to develop demand-driven deployment capabilities in Cyclus, a nuclear fuel cycle simulator. This capability, `d3ploy`, is a Cyclus `Institution` agent that deploys facilities to meet user-defined power demand.

Cyclus is an agent-based nuclear fuel cycle simulation framework [2], each entity (i.e. `Region`, `Institution`, or `Facility`) in the fuel cycle is an agent. An agent-based model enables model development to take place at an agent level rather than a system level [2]. For example, an analyst can design a reactor agent that is entirely independent from an fuel fabrication agent. Each agent's behavior is designed according to the application interface contract, giving them the capability to interact with each other in the simulation [2]. `Region` agents represent geographical or political areas in which `Institution` and `Facility` agents reside. `Institution` agents represent legal operating organizations such as utilities, governments, and control the deployment and decommissioning of `Facility` agents [2]. `Facility` agents represent nuclear fuel cycle facilities such as mines, conversion facilities, reactors, reprocessing facilities, etc. Cycamore [11] provides basic `Region`, `Institution`, and `Facility` archetypes compatible with Cyclus. A complete introduction to Cyclus can be found in [2].

I.B. Novelty

We utilized time series forecasting methods to effectively predict future commodities' supply and demand in `d3ploy`. Solar and wind power generation commonly use these methods to make future predictions based on past time series data [12, 13, 14, 15]. Industrial supply chain management also uses sophisticated time series forecasting techniques to predict demand for quantities of goods in the supply chain [16]. This is a novel approach that has never been applied to nuclear fuel cycle simulators.

I.C. Objectives

The main objectives of this paper are: (1) to describe the demand-driven deployment capabilities in Cyclus, (2) to describe the prediction methods available in `d3ploy`, and (3) to demonstrate the use of `d3ploy` in setting up EG01-23, EG01-24, EG01-29, and EG01-30 transition scenarios with various power demand curves.

II. METHODOLOGY

In Cyclus, developers have the option to design agents using C++ or Python. The `d3ploy` `Institution` agent was implemented in Python to enable the use of well-developed time series forecasting Python packages.

During a Cyclus simulation, at every time step, `d3ploy` predicts the supply and demand of each commodity for the next time step. It is assumed that facility deployment occurs within one time step (month). Commodities refer to materials in the nuclear fuel cycle such as reactor fuel. Upon undersupply for any commodity, `d3ploy` deploys facilities to meet its predicted demand. Therefore, if the simulation begins with user-defined power demand, `d3ploy` deploys reactors to meet power demand, followed by enrichment facilities to meet fuel demand, and so on, to create the supply chain. Based on the demand and supply trends of each commodity, `d3ploy` predicts their future demand and supply, and deploys facilities accordingly to meet the future demand to prevent demand from surpassing supply. Figure 1 shows the logical flow of `d3ploy` at every time step. In subsequent subsections, we describe how to set up a transition scenario using `d3ploy` and the input parameters `d3ploy` accepts.

`d3ploy` aims to minimize the undersupply of power:

$$\text{obj} = \min \sum_{t=1}^{t_f} |D_{t,p} - S_{t,p}|. \tag{1}$$

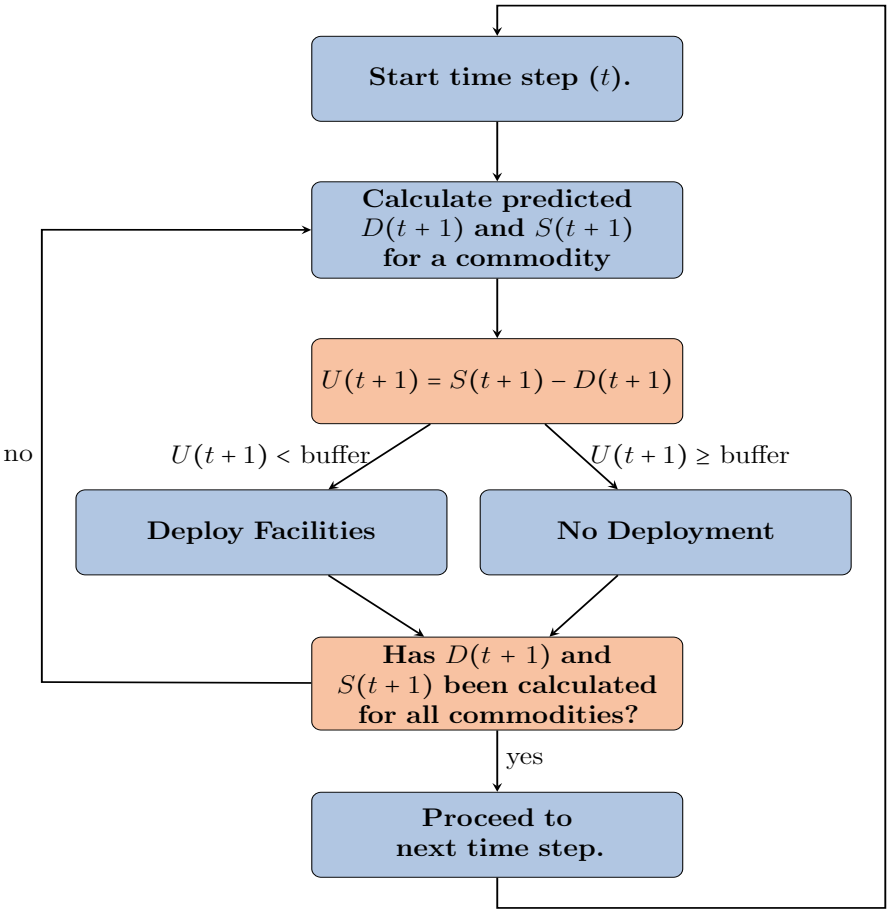


Fig. 1. `d3ploy` logic flow at every time step in Cyclus [17].

where:

t_f = Number of time steps [months]

t = time [month]

D = Demand

S = Supply

p = power [MW]

The sub-objectives are to minimize the number of time steps of undersupply or under-capacity of any commodity:

$$obj = min \sum_{c=1}^{M} \sum_{t=1}^{t_f} |D_{t,c} - S_{t,c}|, \tag{2}$$

and to minimize excessive oversupply of all commodities:

$$obj = min \sum_{c=1}^{M} \sum_{t=1}^{t_f} |S_{t,c} - D_{t,c}|. \tag{3}$$

where:

c = commodity type

M = Number of commmodities

Minimizing excessive oversupply reflects reality, in which utilities ensure grid availability by ensuring power plants are never short of fuel while avoiding expensive storage of excess fuel. Nuclear fuel cycle simulations often face power shortages due to lack of viable fuel, despite having sufficient installed reactor capacity. Using `d3ploy` to automate the deployment of supporting facilities prevents this.

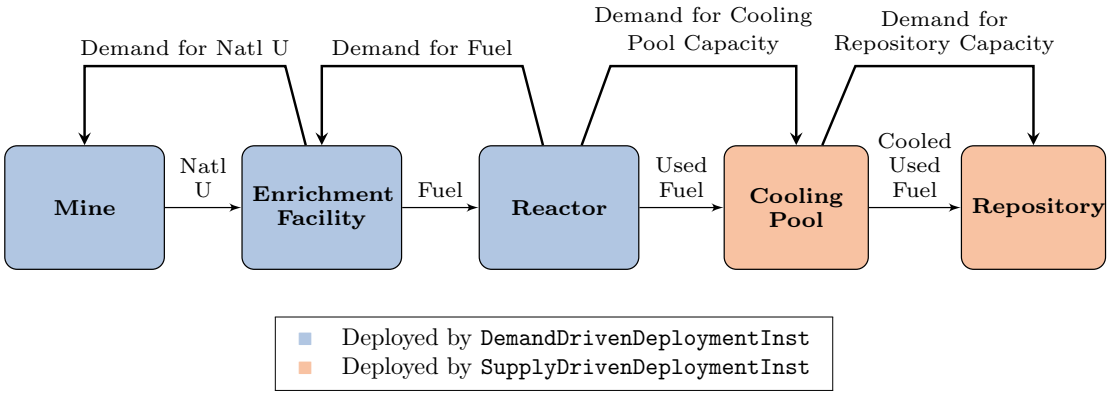


Fig. 2. Simple once-through fuel cycle depicting which facilities are deployed by `DemandDrivenDeploymentInst` and `SupplyDrivenDeploymentInst`.

II.A. Structure

Front-end facilities meet the demand for commodities they produce, whereas back-end facilities meet supply for the commodities they demand. Therefore, in `d3ploy` two distinct institutions control front-end and back-end fuel cycle facilities: `DemandDrivenDeploymentInst` and `SupplyDrivenDeploymentInst`, respectively. For example, when a reactor facility demands fuel, `DemandDriven-DeploymentInst` deploys fuel fabrication facilities to create fuel supply. For back-end facilities, the reactor generates spent fuel, and `SupplyDrivenDeploymentInst` deploys used fuel storage facilities to create capacity to store the spent fuel. Figure 2 depicts a simple once-through fuel cycle and the `Institution` type governing each facility's deployment.

II.A.1. Deployment-Driving Method

To prevent over-deployment of facilities with an intermittent supply such as reactors that require refueling, and to prevent infinite deployment of a facility that demands a commodity no longer available in the simulation, we introduced the capability to deploy facilities based on the difference between predicted demand and installed capacity. The user may deploy facilities based on the difference between predicted demand and predicted supply, *or* predicted demand and installed capacity. For example, a reprocessing plant that fabricates Sodium-Cooled Fast Reactor (SFR) fuel demands for Pu after depletion of the existing Pu inventory and decommissioning of the Light Water Reactors (LWRs) that produce it. If we used the deployment-driving method driven by

the difference in predicted demand and predicted supply, this results in infinite deployment of reprocessing facilities in a futile attempt to produce SFR fuel, crashing the simulation. Instead, if we use the deployment-driving method driven by the difference in predicted demand and installed capacity, only one reprocessing facility will be deployed, the simulation will finish, and the user will see that a large Pu inventory must be accumulated. Therefore, using the deployment-driving method that deploys facilities based on the difference between predicted demand and installed capacity is ideal for most transition scenarios.

II.B. Input Variables

Table II lists and gives examples of the input variables `d3ploy` accepts. The user must define the following input variables:

1. **The available facilities for `d3ploy` to deploy in the simulation and their respective capacities.** Users must define the facilities they want `d3ploy` to deploy. It is the user's responsibility to ensure the defined facilities create a supply chain to produce the demand driving commodity.

2. **The demand driving commodity and its demand equation.** For most simulations, the demand driving commodity is power. The demand equation is defined by a mathematical equation with units of MW. For example, a constant power demand equation is 10000, while a linearly increasing power demand equation is $100t$.

3. **The deployment driving method**. This input variable is described in Section II.A.1.

4. **The prediction method**. This input variable is described in Section II.D. There are also optional input variables:

5. **Supply/capacity buffers for individual commodities.** This input variable is described in section II.B.1.

6. **Facility preferences.** This input variable is described in section II.C.

7. **Facility fleet shares.** This input variable is described in section II.C.

	Input Parameter	Examples
Required	Demand driving commodity	Power
	Demand equation [MW]	P(t) = $10000, sin(t), 10000t$
	Available Facilities	Mine, LWR, Repository, etc.
	Capacities of the facilities	3000 kg, 1000 MW, 50000 kg
	Prediction method	Power: Fast Fourier Transform Fuel: Moving Average Spent fuel: Moving Average
	Deployment driven by	Installed Capacity
Optional	Supply/Capacity Buffer type	Absolute
	Supply/Capacity Buffer size	Power: 3000 MW Fuel: 0 kg Spent fuel: 0 kg
	Facility preferences [month]	LWR = 100-t SFR = t-99
	Fleet share percentage [%]	MOX LWR = 85% SFR = 15%

TABLE II
`d3ploy`'s required and optional input parameters with examples.

II.B.1. Supply/Capacity Buffer

The user has the option to specify a supply buffer for each commodity; `d3ploy` accounts for the buffer when calculating predicted demand and deploys facilities accordingly. The buffer is defined as a percentage:

$$S_{pwb} = S_p(1 + d) \tag{4}$$

or an absolute value:

$$S_{pwb} = S_p + b \tag{5}$$

where:

S_{pwb} = predicted supply/capacity with buffer

S_p = predicted supply/capacity

d = buffer's percentage value in decimal form

b = buffer's absolute value

Using the buffer capability and installed capacity to drive facility deployment in a transition scenario simulation will effectively minimize undersupply of a commodity while avoiding excessive oversupply. This is demonstrated in Section III.A.

II.C. Facility Preference and Fleet Share

The user can define time-dependent preference equations to facilities' that supply the same commodity. If there are two reactor types, LWRs and Sodium-Cooled Fast Reactors (SFRs), in a simulation, the user can make use of time-dependent preferences to make the simulation deploy LWRs at earlier times in the simulation, and deploy SFRs at later times in the simulation when there is a power demand. In Table II, the user defined that the LWR has a preference of $100 - t$, while the SFR has a preference of $t - 99$. Figure 3 depicts how the preference for each reactor changes with time. When there is a power undersupply, `d3ploy` will deploy the reactor that has a larger preference at that time step. At time step 100, LWR preference is 0, while SFR preference is 1; therefore an SFR is deployed if there is a power shortage. Thus, the transition occurs at the 100^{th} time step.

The user also has the option to specify percentage-share for facilities that provide the same commodity. For example, if there are two reactor types, mixed oxide (MOX) LWRs and SFRs, in a simulation, the user can make use of percentage-share specifications to determine the percentage of power supplied by each reactor. When MOX LWR has a share of $s\%$ and SFR has a share of $(100 - s)\%$, MOX LWR deployment constrains to $s\%$ of total power demand and SFR deployment constrains to $(100 - s)\%$ of total power demand.

The transition year is selected by customizing facility preferences to prefer advanced reactors at that year. The fleet-share percentage determines the share of each type of reactor to transition to.

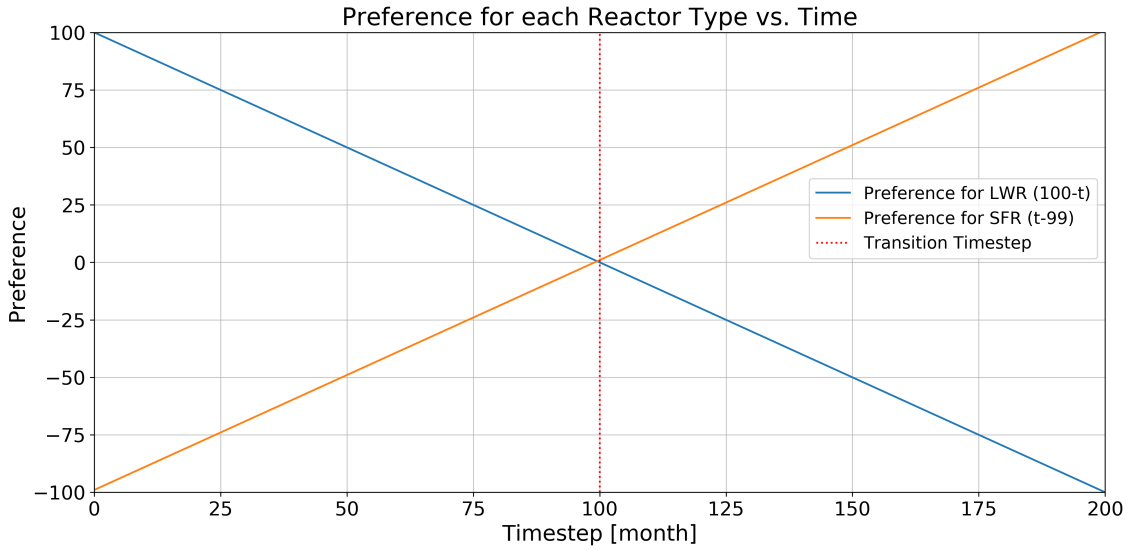


Fig. 3. `d3ploy` has a $100 - t$ preference for LWRs and a $t - 99$ preference for SFRs. When there is a power undersupply, `d3ploy` will deploy the reactor that has a larger preference at that time step.

Figure 4 shows the logical flow of which facility `d3ploy` deploys when there are multiple facilities offering the same commodity.

II.D. Prediction Methods

`d3ploy` records supply and demand at each time step for all commodities. Time-series data informs `d3ploy`'s time series forecasting methods which predict future supply and demand for each commodity. The time series forecasting methods investigated include non-optimizing, deterministic-optimizing, and stochastic-optimizing methods. Non-optimizing methods are techniques that harness simple moving average and autoregression concepts which use historical data to infer future supply and demand values. Deterministic-optimizing and stochastic-optimizing methods are techniques that use an assortment of more sophisticated time series forecasting concepts to predict future supply and demand values. Deterministic-optimizing methods give deterministic solutions, while stochastic-optimizing methods give stochastic solutions.

Depending on the scenario in question, each forecasting method offers distinct benefits and disadvantages. The various methods are compared for each type of simulation to determine the most effective prediction method for a given scenario. The following sections describe the prediction methods.

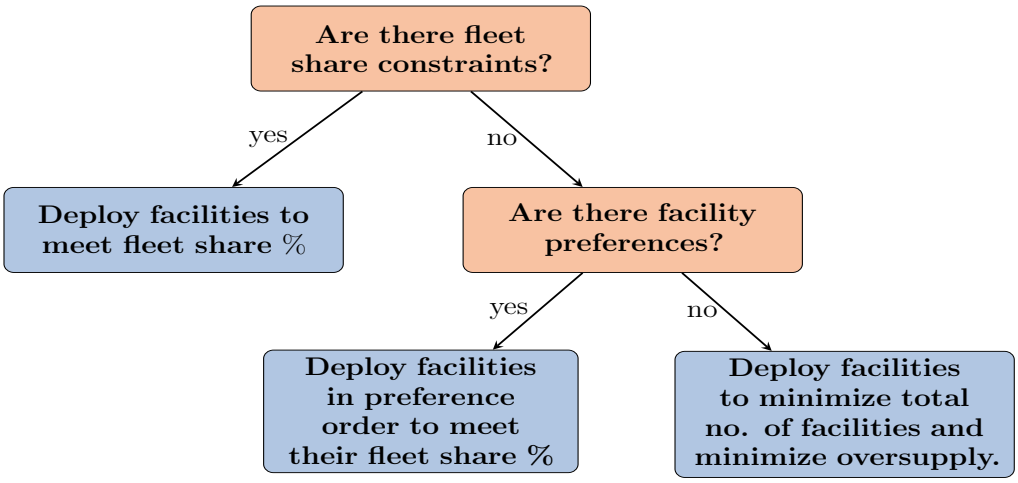


Fig. 4. Logical flow of how `d3ploy` selects which facility to deploy when there are multiple facilities offering the same commodity.

II.D.1. Non-Optimizing Methods

Non-optimizing methods include: Moving Average (`MA`), Autoregressive Moving Average (`ARMA`), and Autoregressive Heteroskedasticity (`ARCH`). The `MA` method calculates the average of a user-defined number of previous entries in a commodity's time series and returns it as the predicted value (equation 6).

$$PV = \frac{\sum_{n=1}^{N} V_n}{n} \tag{6}$$

where:

PV = predicted value

V_n = time series value

N = length of time series

The `ARMA` method combines moving average and autoregressive models (equation 7). The first term is a constant, the second term is white noise, the third term is the autoregressive model, and the fourth term is the moving average model. The `ARMA` method is more accurate than the `MA`

method because of the inclusion of the autoregressive term:

$$X_t = c + \epsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}. \tag{7}$$

where:

c = a constant

ϵ_t = error terms (white noise)

φ = the autoregressive models parameters

θ = the moving average models parameters

p = order of the autoregressive polynomial

q = order of the moving average polynomial

The `ARCH` method models time series data by describing the variance of the current error term as a function of the sizes of the previous time periods' error terms [18]. This allows the method to support changes in the time dependent volatility, such as increasing and decreasing volatility in the same series [18]. The `ARCH` method is better than the `ARMA` method for volatile time-series data [19]. The StatsModels [20] Python package is used to implement `ARMA` and `ARCH` methods in `d3ploy`.

II.D.2. Deterministic-Optimizing Methods

Deterministic methods include Fast Fourier Transform (`FFT`), Polynomial Fit (`POLY`), Exponential Smoothing (`EXP-SMOOTHING`), and Triple Exponential Smoothing (`HOLT-WINTERS`). The `FFT` method uses the fast Fourier transform algorithm to map a time series into the frequency domain. The algorithm returns complex numbers from which frequency, amplitude, and phase is extracted. Future demand and supply values are predicted by summing the significant components, then using the inverse Fourier transform method to return it into a usable form. The discrete Fourier transform (DFT) transforms a sequence of N complex numbers (X_k) into another sequence of complex numbers (x_n) [21]:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi k n / N}. \tag{8}$$

where:

X = sequence of complex numbers

$k = 0, ..., N-1$

N = No. of complex numbers

x = sequence of complex numbers

$n = 0, ..., N-1$

This method is implemented in `d3ploy` using the SciPy [22] Python package.

The `POLY` method fits the time series data with a user-defined n^{th} degree polynomial and uses the fitted trend-line to determine future demand and supply values:

$$Y_t = \beta_0 + \sum_{n=1}^{N} \beta_n t^n + \varepsilon \tag{9}$$

where:

t = time index

n = polynomial order

β = fitted parameters

ε = unobserved random error

This method was implemented in `d3ploy` using the NumPy [23] Python package.

The `EXP-SMOOTHING` and `HOLT-WINTERS` methods use a weighted average of time-series data with exponentially decaying weights for older time series values [24] to create a model to determine future demand and supply values. The `EXP-SMOOTHING` method excels in modeling univariate time series data without trend or seasonality [24]:

$$y_{t+1} = \alpha y_i + (1-\alpha) y_t. \tag{10}$$

where:

$$y = \text{timeseries value}$$

$$\alpha = \text{smoothing factor } (0 < \alpha < 1)$$

The `HOLT-WINTERS` method applies triple exponential smoothing, resulting in higher accuracy when modeling seasonal time series data [25]:

$$\begin{aligned} F_{t+m} &= (S_t + mb_t)I_{t-L+m} \\ S_t &= \alpha\frac{y_t}{I_{t-L}} + (1-\alpha)(S_{t-1} + b_{t-1}) \\ b_t &= \gamma(S_t - S_{t-1})+)(1-\gamma)b_{t-1} \\ I_t &= \beta\frac{y_t}{S_t} + (1-\beta)I_{t-L} \end{aligned} \tag{11}$$

where:

$$\begin{aligned} F &= \text{forecast at m periods ahead} \\ t &= \text{time period index} \\ L &= \text{periods in a season} \\ S &= \text{smoothed observation} \\ y &= \text{the observation} \\ b &= \text{trend factor} \\ I &= \text{seasonal index} \\ \alpha, \beta, \gamma &= \text{constants} \end{aligned}$$

The StatsModels [20] Python package was used to implement the `EXP-SMOOTHING` and `HOLT-WINTERS` methods in `d3ploy`.

II.E. Stochastic-Optimizing Methods

We implemented one stochastic-optimizing method: step-wise seasonal method (`SW-SEASONAL`). The method was implemented in `d3ploy` by the Auto-Regressive Integrated Moving Averages (ARIMA) method in the pmdarima [26] Python package. The ARIMA model is a dependent time series that is modeled as a linear combination of its own past values and past values of an error series [27]:

$$(1-B)^d Y_t = \mu + \frac{\theta(B)}{\phi(B)} a_t \tag{12}$$

where:

t = time index

μ = mean term

B = backshift operator, such that $BX_t = X_{t-1}$

d = no. of roots

Y = timeseries data

$\phi(B)$ = autoregressive operator

$\theta(B)$ = moving average operator

a_t = random error

III. RESULTS

This section aims to demonstrate `d3ploy`'s capability to completely automate the setup of transition scenarios and meet the objectives described in section I.C. This section is split into two subsections. The first subsection (section III.A) will demonstrate `d3ploy`'s capabilities to set up a simple transition scenario with only three facility types. The second subsection (section III.B) will demonstrate `d3ploy`'s capabilities to set up complex EG01-23, EG01-24, EG01-29, EG01-30 transition scenarios and is further subdivided into:

	Input Parameters	Simple Transition Scenario
Required	Demand driving commodity	Power
	Demand equation [MW]	$t < 40 = 1000, t \geq 40 = 1000 + 250t$
	Available facilities	`Source`, `Reactor`, `Sink`
	Prediction method	`FFT`
	Deployment driving method	Installed Capacity
Optional	Buffer type	Absolute
	Buffer size	Power: 2000MW, Fuel: 1000kg

TABLE III
`d3ploy`'s input parameters for the simple transition scenario with linearly increasing power demand.

1. Section III.B.1: compare the use of different `d3ploy` prediction methods in EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios,

2. Section III.B.2: compare the use of varied power buffer sizes in EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios, and

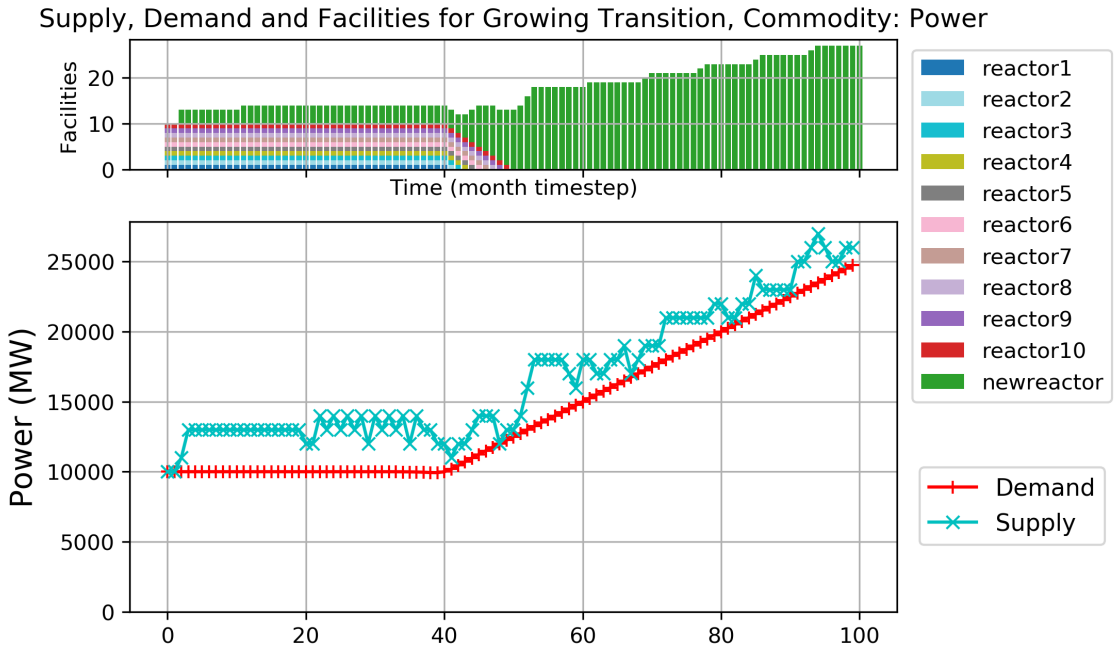
3. Section III.B.3: demonstrate successful `d3ploy` setup of EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios using the prediction method and power buffer size that proved to best minimize power undersupply in the Sections III.B.1 and III.B.2. These will be referred to as 'best performance models'.

The input files and scripts to reproduce the results and plots in this paper are found in [28] and [29].

III.A. Simple Transition Scenario

We conducted a simple transition scenario simulation with linearly increasing power demand to demonstrate `d3ploy`'s capabilities and inform input parameter choices when setting up complex many-facility transition scenarios. This simulation is defined as *simple* since it only includes three facility types: `source`, `reactor`, and `sink`. The simulation begins with ten `reactor` facilities (`reactor1` to `reactor10`). These reactors have staggered cycle lengths and lifetimes to prevent simultaneous refueling and set up gradual decommissioning. `d3ploy` is configured to deploy `new reactor` facilities to meet the loss of power supply created by the decommissioning of the initial `reactor` facilities. Table III shows the `d3ploy` input parameters for this simulation.

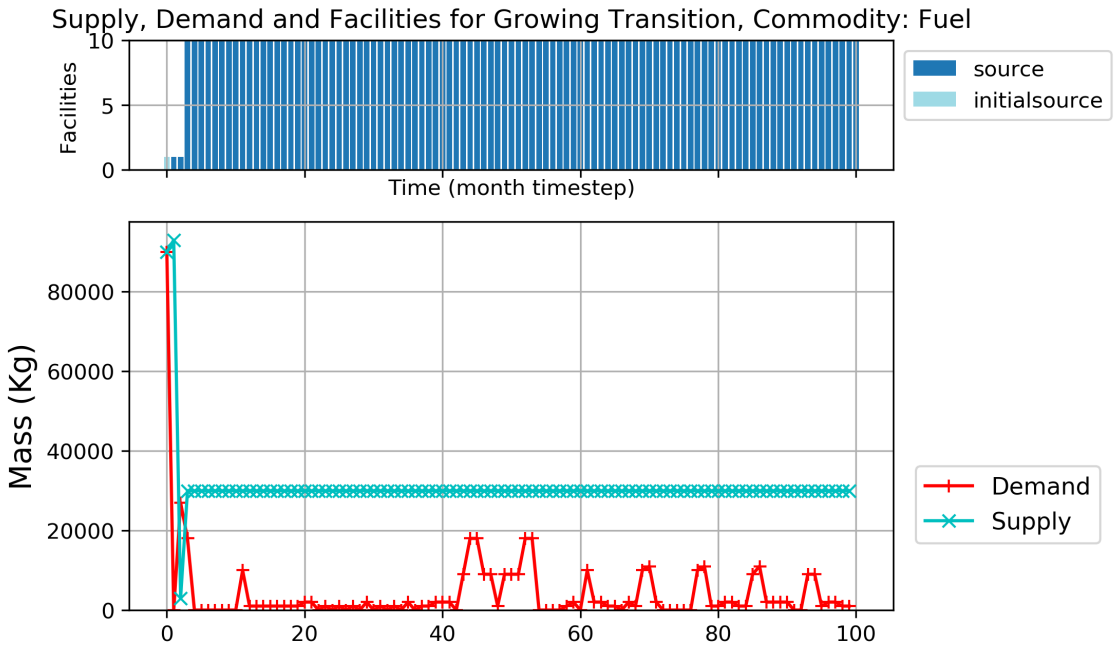
Figures 5(a), 6(a), and 6(b) demonstrate `d3ploy`'s capability to deploy reactors and supporting facilities to minimize undersupply when meeting linearly increasing power demand and subsequent



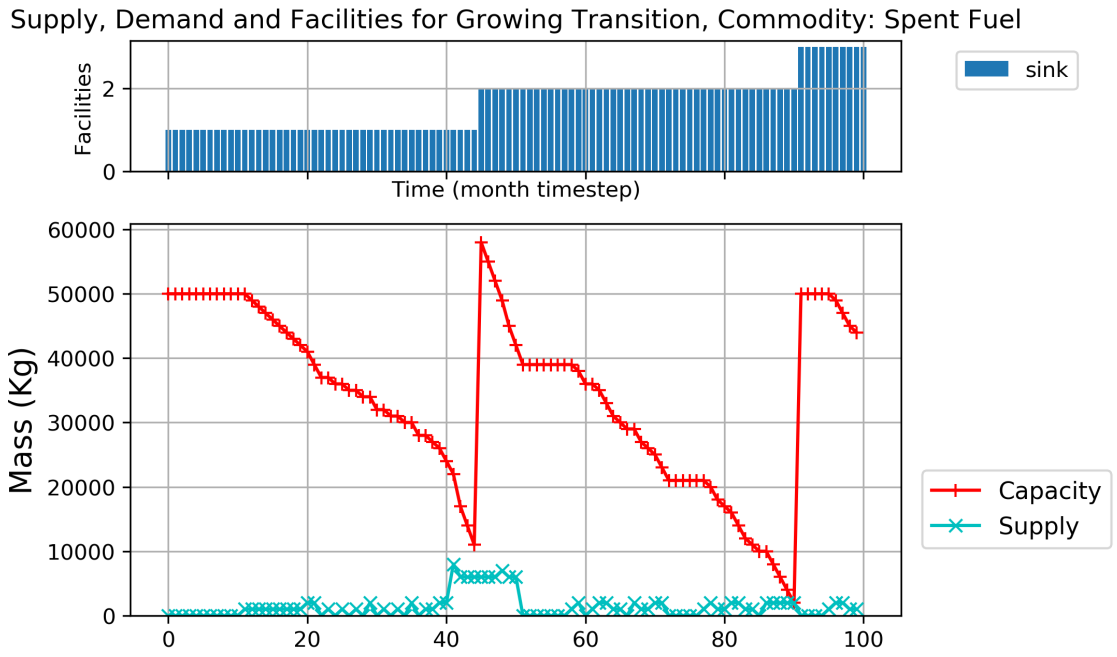
(a) Power demand and supply, and reactor facility deployment plot for a simple linearly increasing power demand transition scenario with three facility types: `source`, `reactor`, and `sink`. The simulation begins with `reactor1` to `reactor10` and `d3ploy` deploys `newreactors` to meet increasing power demand.

secondary commodities demand. In Figure 5(a) there exists no time steps in which the supply of power falls under demand, meeting the main objective of `d3ploy`. By using a combination of the `FFT` method for predicting demand and a power supply buffer of 2000MW (the capacity of 2 reactors), we minimized the number of undersupplied time steps for every commodity.

In figure 6(a), a large-throughput source facility is initially deployed to meet the large initial fuel demand for the commissioning of ten reactors. Deployment of a large-throughput source facility for the first few time steps ensures `d3ploy` does not deploy supporting facilities that become redundant at later times in the simulation. This reflects reality in which reactor manufacturers accumulate an appropriate amount of fuel inventory before starting up reactors.



(a) Fuel demand and supply, and source facility deployment plot. Fuel is demanded by reactors and supplied by source facilities. There is only one time step with undersupply of fuel.



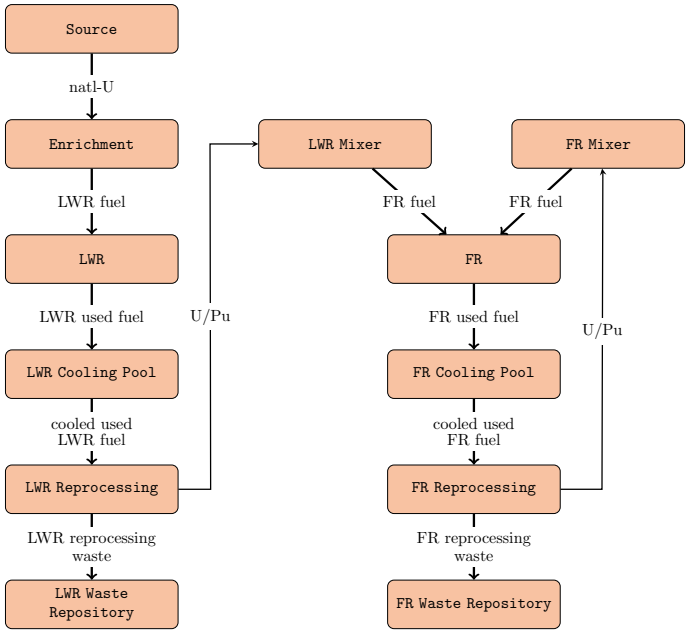
(b) Spent fuel capacity and supply, and sink facility deployment plot. Spent fuel is supplied by reactors and the capacity to store them is provided by sink facilities. There are no time steps with under-capacity of sink space.

Fig. 6. Simple linearly increasing power demand transition scenario with three facility types: `source`, `reactor`, and `sink`.

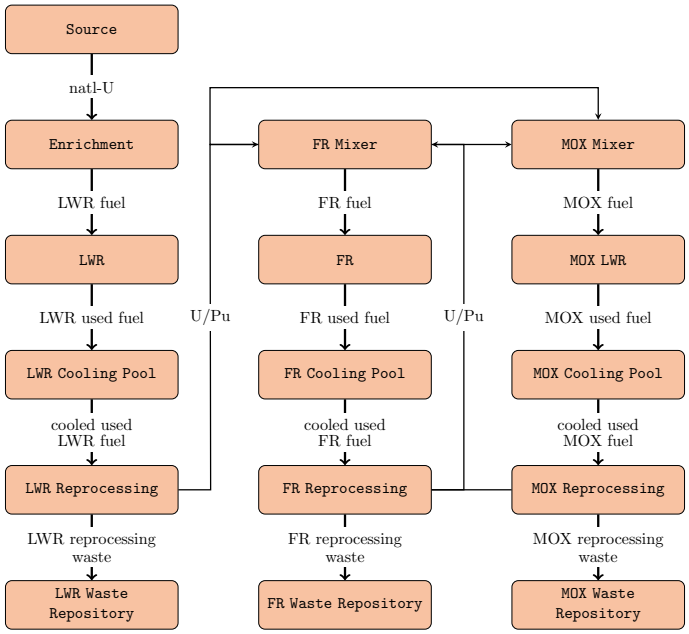
III.B. Complex Transition Scenarios

EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios are automatically set up in CYCLUS using `d3ploy`. We defined the EG01-EG23 and EG01-EG29 transition scenario simulations to have a constant power demand, while EG01-EG24 and EG01-EG30 have a linearly increasing power demand. Similar to the simple transition scenario, these transition scenario simulations begin with an initial fleet of LWRs that start progressively decommissioning at the 80-year mark, after which `d3ploy` deploys SFRs and MOX LWRs to meet the power demand. Figure 7 shows the setup of facilities and mass flows for EG01-EG23 and EG01-EG29 in CYCLUS. In EG01-EG23 and EG01-EG29, recycled plutonium from LWR spent fuel produces SFR fuel. EG01-EG24 and EG01-EG30 are similar to EG01-EG23 and EG01-EG29, respectively, with the exception that all transuranic elements are recycled.

The facilities used in the transition scenario simulations are described below. The source facility has a throughput of 1e8kg of natural uranium, and the enrichment facility has a SWU capacity of 1e100. The LWRs have an assembly size of 29863.3kg with 3 assemblies per core, and a power capacity of 1000 MW. The FRs have an assembly size of 3950kg and power capacity of 333.34 MW. The MOX LWRs have an assembly size of 33130kg and power capacity of 1000 MW. The `reactor` facility used in the CYCLUS simulation is a recipe reactor; it accepts a fresh fuel recipe and outputs a spent fuel recipe. The recipes used for the LWR, MOX LWR, and SFR are based on recipes generated by VISION [29] that closely match EG30 scenario specifications in Appendix B of the Department of Energy (DOE) Evaluation and Screening Study (E&S study) [6]. The LWR, FR, and MOX LWR cooling pools have a residence time of 36 months, and a max inventory size of 1e8kg of fuel. The reprocessing segment for each reactor type has a reprocessing and mixer facility. Each reprocessing facility has a throughput of 1e8kg and separates U/Pu or U/TRU from other isotope in spent fuel. Each mixer facility mixes the U/Pu or U/TRU to fabricate new reprocessed fuel. `d3ploy` will deploy reprocessing facilities based on the demand of reprocessed fuel from the MOX LWRs and FRs to ensure that sufficient fissile material feeds the reprocessing facilities to make sufficient reprocessed fuel for each reactor type. Each waste repository is assumed to have infinite capacity. For more details about each simulation, the input files can be found at [29].



(a) EG01-EG23.



(b) EG01-EG29.

Fig. 7. Facility and mass flow of the transition scenarios EG01-EG23 and EG01-EG29 in Cyclus. EG23 and EG29 are closed fuel cycles with continuous recycling of U/Pu. EG23 consists of fast reactors, while EG29 consists of both fast and thermal reactors.

No. of Time Steps with Power Undersupply for Each Transition Scenario				
Algorithm	**EG01-EG23**	**EG01-EG24**	**EG01-EG29**	**EG01-EG30**
MA	26	36	15	24
ARMA	26	36	15	24
ARCH	26	36	15	21
POLY	6	65	4	9
EXP-SMOOTHING	27	37	16	25
HOLT-WINTERS	27	37	16	25
FFT	8	20	5	9
SW-SEASONAL	36	107	14	51

TABLE IV
Total number of time steps with undersupply of power for the EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios for different prediction methods.

III.B.1. Comparison of Prediction Methods

EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios are set up in Cyclus using `d3ploy`. We ran each transition scenario with different prediction methods to determine the prediction method that best minimizes power undersupply for that scenario.

In Figure 8, each histogram represents the number of time steps with undersupply or under capacity for all commodities for each prediction method. Table IV shows the total number of time steps with power undersupply for constant power EG01-EG23 and EG01-EG29 transition scenarios and linearly increasing power EG01-EG24 and EG01-EG30 transition scenarios for each prediction method. Figure 8 demonstrates that the `POLY` method minimized undersupply for all commodities for the EG01-EG23 transition scenario, with the smallest bars on the plot, indicating that they have the fewest number of time steps with undersupply and under capacity of commodities. We conducted a similar analysis for the constant power EG01-EG29 scenario, and as seen in Table IV, the `POLY` prediction method also minimized undersupply for all commodities for minimizing undersupply of power.

In Figure 9, each histogram represents the number of time steps with undersupply or under capacity for all commodities for each prediction method. Figure 9 demonstrates that the `FFT` method minimized undersupply for all commodities for the EG01-EG24 transition scenario. We conducted a similar analysis for the linearly increasing power EG01-EG30 scenario, and as seen in Table IV, the `FFT` prediction method also minimized undersupply for all commodities.

Figures 8, 9, and Table IV show that the `POLY` method minimizes power undersupply for constant power transition scenarios, and the `FFT` method minimizes power undersupply for linearly

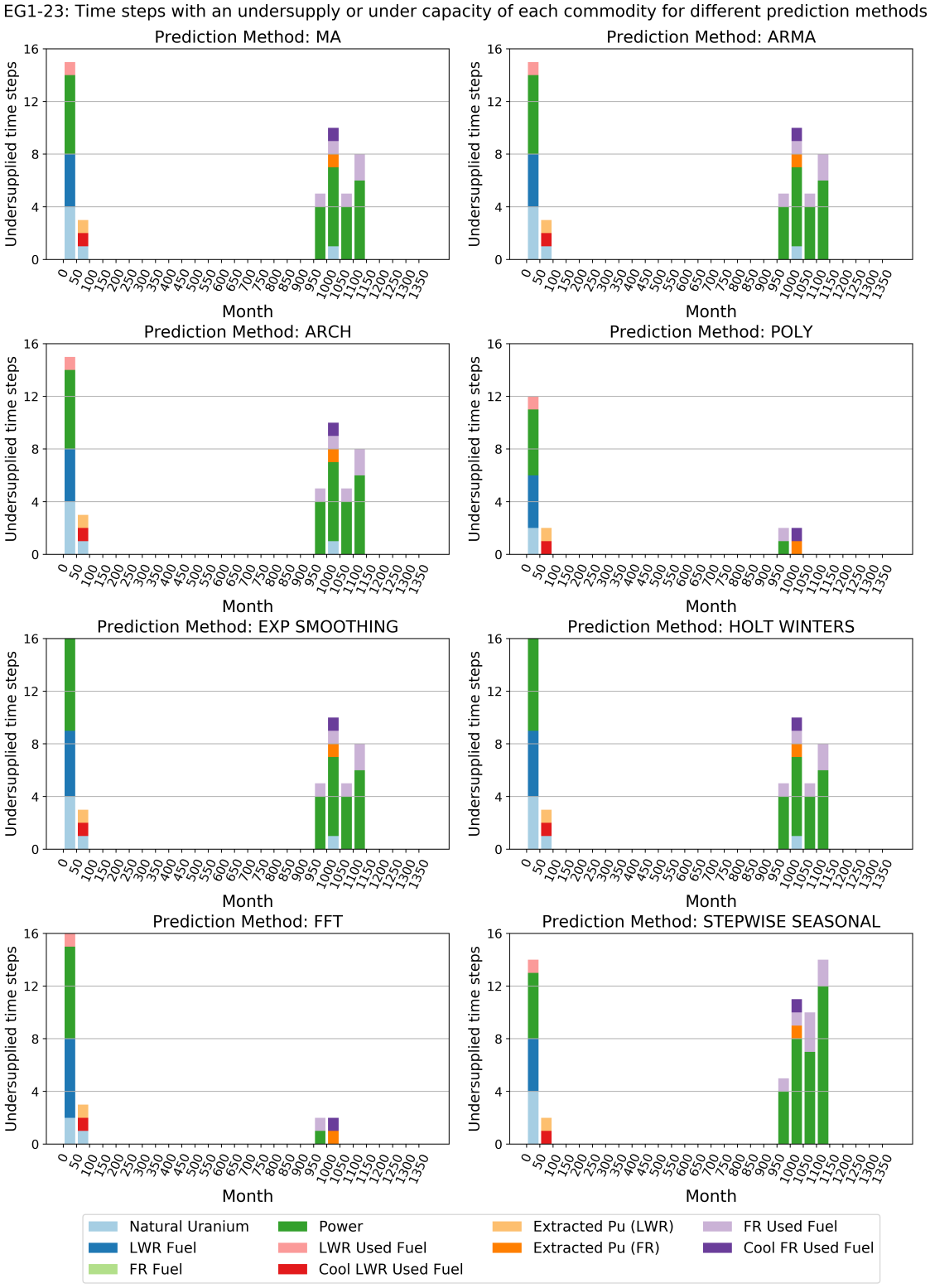


Fig. 8. EG01-EG23 transition scenario with constant power demand. Each subplot shows the total number of time steps in which there exists undersupply and under capacity of commodities for each prediction method. The different colors represent different commodities and each vertical bar refers to 50 time steps in the simulation.

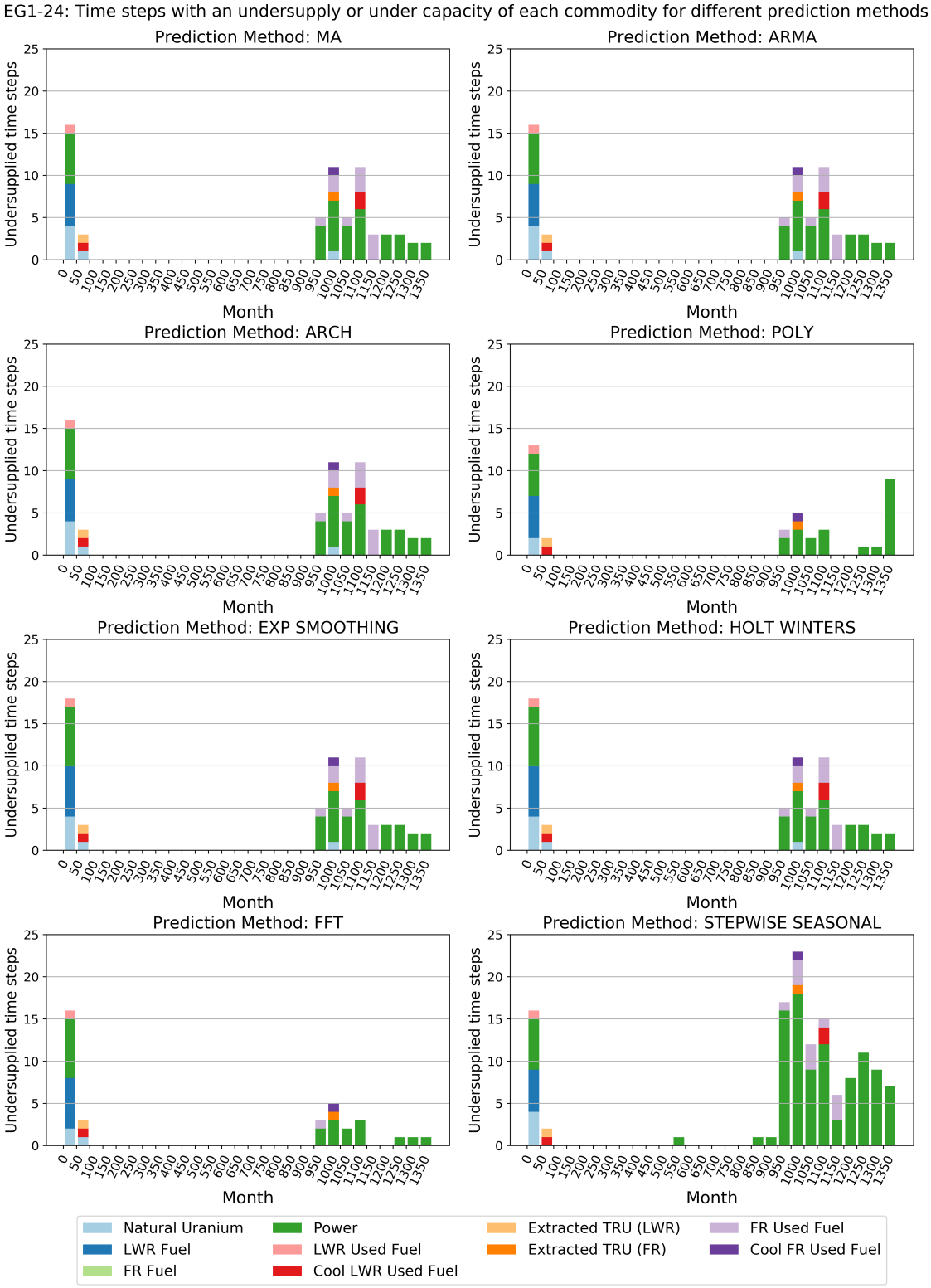


Fig. 9. EG01-EG24 transition scenario with linearly increasing power demand. Each subplot shows the total number of time steps in which there exists undersupply and under capacity of commodities for each prediction method. The different colors represent different commodities and each vertical bar refers to 50 time steps in the simulation.

increasing power transition scenarios. Undersupply and under-capacity of commodities occur during two main time periods: initial demand for the commodity and during the transition period (month 950 onwards). The `POLY` method minimizes commodity undersupply during the transition period, and does especially well during the start of the simulation in Figure 8. We hypothesize that it is because a first order polynomial was used, and thus, `POLY` could best predict the future demand of each commodity. The `FFT` method struggled with predicting the demand at the start of the simulation in both Figures 8 and 9, but did very well during the transition period for both simulations. The reason why it is so effective is that is able to capture the significant features of the time series data and uses it to predict future demand values. It is weaker at the beginning of the simulation because there is a lack of time series data. Different methods perform well for different power demand curves. In [17], we demonstrate that the `HOLT-WINTERS` method minimizes undersupply of all commodities for a sinusoidal power demand curve. This is because the triple exponential smoothing method excels in forecasting data points for repetitive seasonal series of data [17].

To further `d3ploy`'s primary objective of minimizing the power undersupply, sensitivity analysis of the power supply buffer is conducted with the prediction method that best minimizes commodity undersupply for each transition scenario.

III.B.2. Comparison of Power Buffer Sizes

For the EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios, the power buffer size is varied for the prediction method which minimizes commodity undersupply, which is the `POLY` method for EG01-EG23 and EG01-EG29, and the `FFT` method for EG01-EG24 and EG01-EG30. Varying the power buffer size does not impact the number of undersupplied time steps for the EG01-EG23 and EG01-EG29 constant power demand transition scenarios. In Table V, there are 6 and 4 time steps in which there is power undersupply for the EG01-EG23 and EG01-EG29 transition scenarios, respectively. As seen in figure 8, these undersupply time steps occur at the beginning of the simulation and for one time step when the transition begins. We expected this since without time-series data at the beginning of the simulation, `d3ploy` takes a few time steps to collect time-series data about power demand to predict and start deploying reactors and supporting fuel cycle facilities. When the transition begins, power is undersupplied for

	No. of Time Steps with Undersupply			
Transition Scenario	EG01-EG23	EG01-EG24	EG01-EG29	EG01-EG30
Commodities				
Natural Uranium	2	3	1	1
LWR Fuel	4	6	1	2
SFR Fuel	0	0	2	2
MOX LWR Fuel	-	-	2	2
Power	6	7	4	5
LWR Spent Fuel	1	1	1	1
SFR Spent Fuel	1	1	1	1
MOX LWR Spent Fuel	-	-	1	1

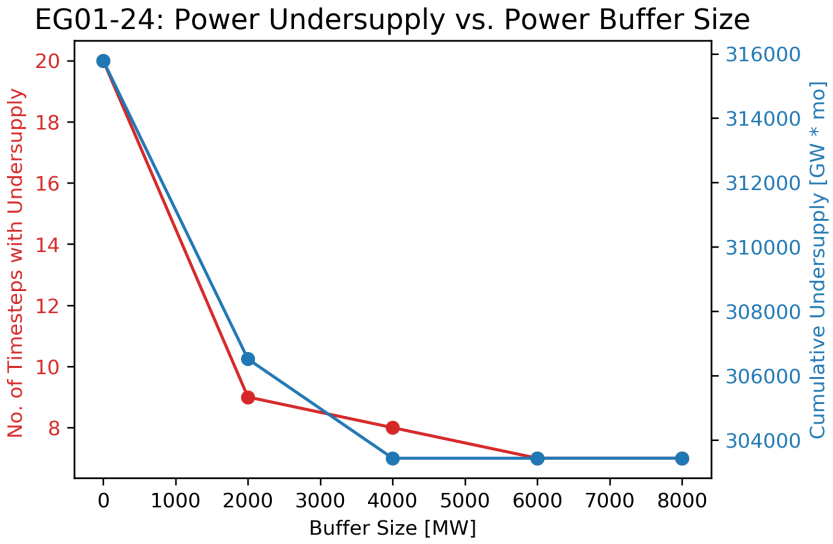
TABLE V
Undersupply/capacity of key commodities for the best performing EG01-EG23, EG24, EG29, EG30 transition scenarios.

one time step, following this, `d3ploy` accounts for the undersupply by deploying facilities to meet power demand. Therefore, we minimized the power undersupply for constant power EG01-EG23 and EG01-EG29 transition scenarios with a 0MW power supply buffer.

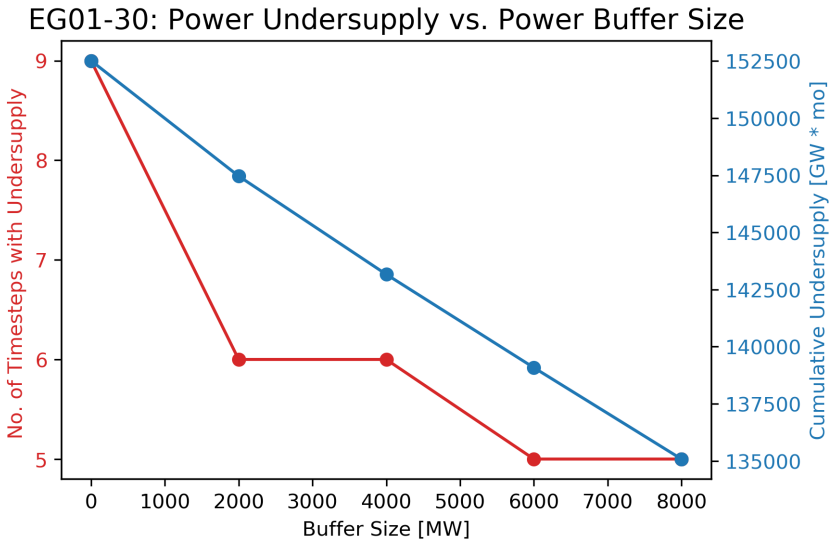
We varied the power buffer size for the EG01-EG24 and EG01-EG30 linearly increasing power demand transition scenarios. Figures 10(a), 10(b), and Table VI show that increasing the buffer size increases the robustness of the supply chain by minimizing power undersupply. The cumulative undersupply is minimized with a 6000MW and 8000MW buffer for EG01-EG24 and EG01-EG30 respectively. In Figure 10(a), a 4000MW buffer size has 8 time steps with undersupply, while a 6000MW buffer size has 7 time steps with undersupply. In Figure 10(b), a 2000MW buffer size has 6 time steps with undersupply, while a 8000MW buffer size has 5 time steps with undersupply. We determined that extra commissioning of multiple reactors does not justify a single time step with no undersupply. This type of logic is difficult to program into a NFC simulator, therefore, even though NFC simulators can help inform policy decisions, decision-makers must still evaluate NFC simulator results to determine if they are valid and logical. Therefore, a buffer of 4000MW and 2000MW minimizes the power undersupply for EG01-EG24 and EG01-EG30 transition scenarios, respectively.

III.B.3. Best Performance Models

The 'best performance models' for each transition scenario use the prediction method and power buffer size determined in the previous subsections that best minimize the undersupply of



(a) EG01-EG24: Power buffer size vs. cumulative undersupply



(b) EG01-EG30: Power buffer size vs. cumulative undersupply

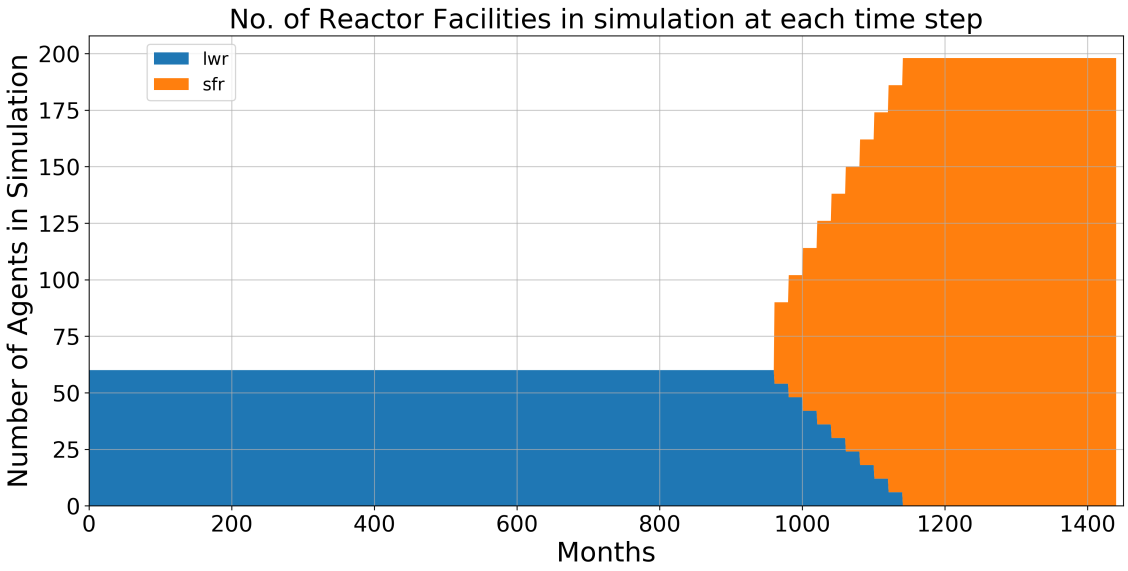
Fig. 10. The effect of sensitivity analysis of power buffer size on cumulative undersupply of power for EG01-EG24 and EG01-EG30 transition scenarios with linearly increasing power demand using the `FFT` prediction method.

Buffer [MW]	Undersupply	EG01-EG24	EG01-EG30
0	Time steps $[\#]$	20	9
	Energy $[GW \cdot mo]$	315791	152517
2000	Undersupplied $[\#]$	9	6
	Energy $[GW \cdot mo]$	306520	147166
4000	Time steps $[\#]$	8	6
	Energy $[GW \cdot mo]$	303438	143166
6000	Time steps $[\#]$	7	5
	Cumulative $[GW]$	303438	139083
8000	Time steps $[\#]$	7	5
	Energy $[GW \cdot mo]$	303438	135083

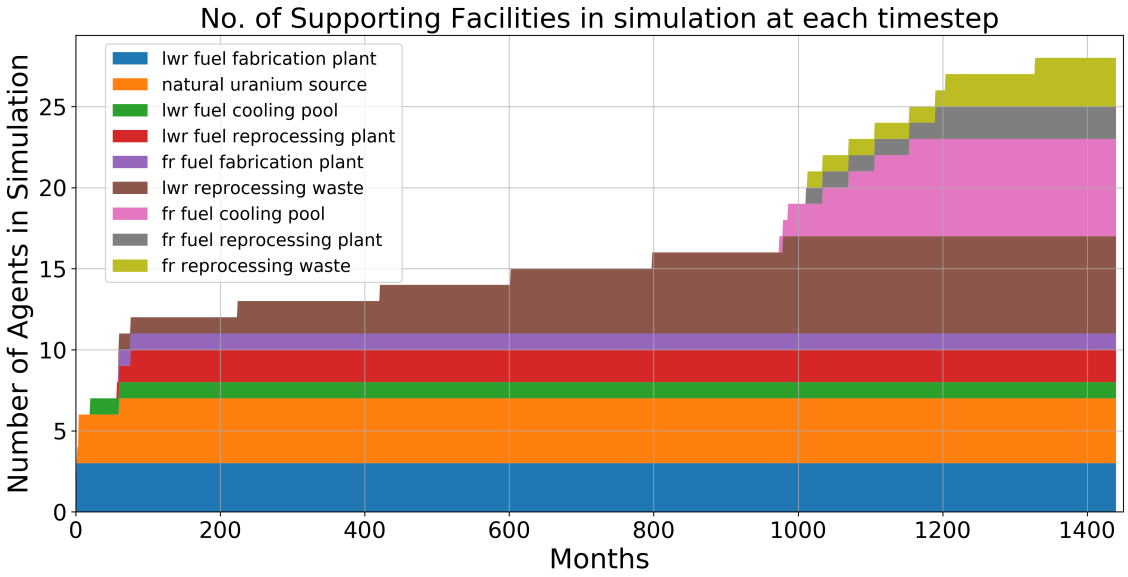
TABLE VI
The effect of sensitivity analysis of power buffer size on cumulative undersupply of power for EG01-EG24 and EG01-EG30 transition scenarios with linearly increasing power demand using the `FFT` prediction method.

power as well as the undersupply and under-capacity of the other commodities in the simulation. Table VII shows the `d3ploy` input parameters for these EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios. The need for commodity supply buffers is a reflection of reality in which a supply buffer is usually maintained to ensure continuity in the event of an unexpected failure in the supply chain [30].

Figures 11 and 12 show time-dependent deployment of reactor and supporting facilities for the EG01-EG23 constant power demand and EG01-EG30 linearly increasing power demand transition scenarios, respectively. `d3ploy` automatically deploys reactor and supporting facilities to set up a supply chain to meet power demand during a transition from LWRs to SFRs for EG01-EG23, and from LWRs to MOX LWRs and SFRs for EG01-EG30. EG01-EG24 and EG01-EG29 facility deployment plots are similar to EG01-EG23 and EG01-EG30, respectively, therefore they are not shown.

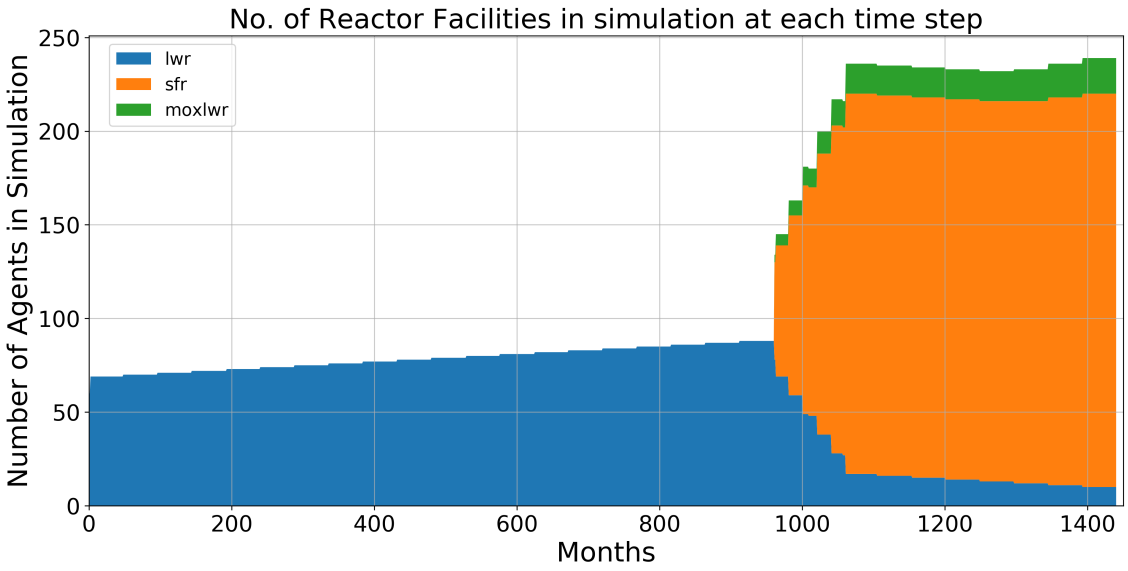


(a) EG01-EG23: Reactor Deployment

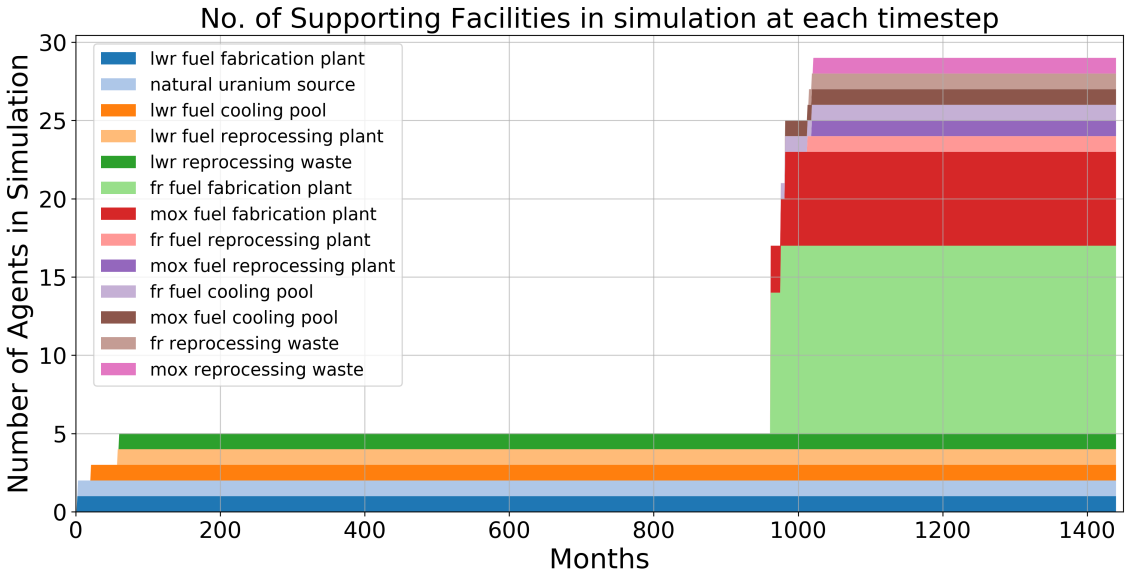


(b) EG01-EG23: Supporting Facility Deployment

Fig. 11. Time dependent deployment of reactor and supporting facilities in the EG01-EG23 constant power demand transition scenario. `d3ploy` automatically deploys reactor and supporting facilities to setup a supply chain to meet constant power demand of 60000 MW during a transition from LWRs to SFRs.



(a) EG01-EG30: Reactor Deployment



(b) EG01-EG30: Supporting Facility Deployment

Fig. 12. Time dependent deployment of reactor and supporting facilities in the EG01-EG30 linearly increasing power demand transition scenario. `d3ploy` automatically deploys reactor and supporting facilities to setup a supply chain to meet linearly increasing power demand of $60000 + 250t/12$ MW during a transition from LWRs to MOX LWRs and SFRs.

Input Parameter	Simulation Description			
	EG01-EG23	**EG01-EG24**	**EG01-EG29**	**EG01-EG30**
Demand Driving Commodity	Power	Power	Power	Power
Demand Equation [MW]	60000	60000 $+250t/12$	60000	60000 $+250t/12$
Prediction Method	`POLY`	`FFT`	`POLY`	`FFT`
Deployment Driving Method	Installed Capacity	Installed Capacity	Installed Capacity	Installed Capacity
Fleet Share Percentage	MOX: 85% SFR: 15%	MOX: 85% SFR: 15%	MOX: 85% SFR: 15%	MOX :85% SFR: 15%
Buffer type	Absolute			
Power Buffer Size [MW]	0	4000	0	2000

TABLE VII
`d3ploy`'s input parameters for EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios that minimizes undersupply of power and minimizes the undersupply and under-capacity of the other facilities.

IV. CONCLUSION

The present nuclear fuel cycle in the United States is a once-through fuel cycle of LWRs with no used fuel reprocessing. This nuclear fuel cycle faces cost, safety, proliferation, and spent nuclear fuel challenges that hinder large-scale nuclear power deployment. The U.S Department of Energy identified future nuclear fuel cycles, involving continuous recycling of co-extracted U/Pu or U/TRU in fast and thermal spectrum reactors, that may overcome these challenges. These transition scenarios have been modeled previously in the following nuclear fuel cycle simulators [10, 31]: ORION, DYMOND, VISION, MARKAL, and Cyclus. However, for many nuclear fuel cycle simulators, the user is required to define a deployment scheme for all supporting facilities to avoid any supply chain gaps or resulting idle reactor capacity. Manually determining a deployment scheme for a once-through fuel cycle is straightforward; however, for complex fuel cycle scenarios, it is not. In this paper, we introduce the capability, `d3ploy`, in Cyclus that automatically deploys fuel cycle facilities to meet user-defined power demand. In this paper, we demonstrate that with careful selection of `d3ploy` parameters, we can completely automate the setup of constant and linearly increasing power demand transition scenarios for EG01-23, EG01-24, EG01-29, and EG01-30 with minimal power undersupply. Using `d3ploy` to set up transition scenarios saves the user simulation

set-up time, making it more efficient than the previous efforts that required a user to manually calculate and use trial and error to set up the deployment scheme for the supporting fuel cycle facilities. Transition scenario simulations set up manually are sensitive to changes in the input parameters resulting in an arduous setup process since a slight change in one input parameter would result in the need to recalculate the deployment scheme to ensure no undersupply of power. Therefore, by automating this process, when the user varies input parameters in the simulation, `d3ploy` automatically adjusts the deployment scheme to meet the new constraints.

V. FUTURE WORK

We simulate transition scenarios to predict the future; however, when implemented in the real world, the transition scenario tends to deviate from the optimal scenario. Therefore, nuclear fuel cycle simulators must be used to conduct sensitivity analysis studies to understand the subtleties of a transition scenario better to reliably inform policy decisions. Previously it was difficult to conduct sensitivity analysis with Cyclus as users have to manually calculate the deployment scheme for a single change in an input parameter. By using the `d3ploy` capability, sensitivity analysis studies are more efficiently conducted as facility deployment in transition scenarios are automatically set up. `d3ploy` will also be open-source and available for the forseeable future on github [28], to be used with Cyclus for conducting any transition scenario analysis. The transition scenario simulations in this work assumed recipe reactors, however, complexity introduced by the reprocessing plants cause the reactor's incoming and outgoing material composition to be dynamic. Therefore, making simple static assumptions such as recipe method is a poor approximation [32, 33]. In future transition scenario work with `d3ploy`, a Cyclus reactor archetype that uses dynamic fuel compositions could be used.

VI. ACKNOWLEDGMENTS

DOE Office of Nuclear Energy funds this research through the Nuclear Energy University Program (Project 16-10512, DE-NE0008567) 'Demand-Driven Cycamore Archetypes'. The authors want to thank members of the Advanced Reactors and Fuel Cycles (ARFC) group at the University of Illinois at Urbana-Champaign. Special thanks to Kip Kleimenhagen and Matthew Kozak for their excellent proofreading help. We also thank our colleagues from the Cyclus community for

collaborative Cyclus development.

The authors contributed to this work as described below. Gwendolyn J. Chee conceived and designed the simulations, wrote the paper, prepared figures and/or tables, performed the computation work, contributed to and validated the software product, and reviewed drafts of the paper. Roberto E. Fairhurst Agosta performed the computation work and contributed to the software product. Jin Whan Bae conceived and designed the simulations and contributed to and validated the software product. Robert R. Flanagan conceived and designed the simulations and contributed to the software product. Anthony M. Scopatz conceived and designed the simulations and acquired funding support for the project. Kathryn D. Huff directed and supervised the work, acquired funding support for the project, conceived and designed the simulations, contributed to the software product, and reviewed drafts of the paper.

REFERENCES

[1] A. M. Yacout, J. J. Jacobson, G. E. Matthern, S. J. Piet, and A. Moisseytsev, "Modeling the Nuclear Fuel Cycle," *The 23rd International Conference of the System Dynamics Society," Boston*, Citeseer (2005)URL http://www.inl.gov/technicalpublications/Documents/3169906.pdf.

[2] K. D. Huff, M. J. Gidden, R. W. Carlsen, R. R. Flanagan, M. B. McGarry, A. C. Opotowsky, E. A. Schneider, A. M. Scopatz, and P. P. H. Wilson, "Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework," *Advances in Engineering Software*, **94**, 46 (2016); 10.1016/j.advengsoft.2016.01.014., URL http://www.sciencedirect.com/science/article/pii/S0965997816300229, arXiv: 1509.03604.

[3] K. D. Huff, J. W. Bae, K. A. Mummah, R. R. Flanagan, and A. M. Scopatz, "Current Status of Predictive Transition Capability in Fuel Cycle Simulation," *Proceedings of Global 2017*, 11, American Nuclear Society, Seoul, South Korea (2017).

[4] *Climate Change and Nuclear Power 2018*, Non-serial Publications, INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna (2018)URL https://www.iaea.org/publications/13395/climate-change-and-nuclear-power-2018.

[5] "The Future of Nuclear Energy in a Carbon-Constrained World," *Massachusetts Institute of Technology Energy Initiative (MITEI)*, 272 (2018).

[6] R. Wigeland, T. Taiwo, H. Ludewig, M. Todosow, W. Halsey, J. Gehin, R. Jubin, J. Buelt, S. Stockinger, K. Jenni, and B. Oakley, "Nuclear Fuel Cycle Evaluation and Screening – Final Report," INL/EXT-14-31465, U.S. Department of Energy (2014)URL https://fuelcycleevaluation.inl.gov.

[7] "New DOE Nuclear Energy Chief Suggests Rethinking Spent Fuel Reprocessing - ExchangeMonitor | Page 1," (2019)URL https://www.exchangemonitor.com/new-doe-nuclear-energy-chief-suggests-rethinking-spent-fuel-reprocessing/?printmode=1, library Catalog: www.exchangemonitor.com Section: Department of Energy.

[8] M. Crapo, "S.97 - 115th Congress (2017-2018): Nuclear Energy Innovation Capabilities Act of 2017," (2018)URL https://www.congress.gov/bill/115th-congress/senate-bill/97, archive Location: 2017/2018 Library Catalog: www.congress.gov.

[9] R. E. Latta, "H.R.590 - 115th Congress (2017-2018): Advanced Nuclear Technology Development Act of 2017," (2017)URL https://www.congress.gov/bill/115th-congress/house-bill/590, archive Location: 2017/2018 Library Catalog: www.congress.gov.

[10] B. Feng, B. Dixon, E. Sunny, A. Cuadra, J. Jacobson, N. R. Brown, J. Powers, A. Worrall, S. Passerini, and R. Gregg, "Standardized verification of fuel cycle modeling," *Annals of Nuclear Energy*, **94**, 300 (2016); 10.1016/j.anucene.2016.03.002., URL http://www.sciencedirect.com/science/article/pii/S0306454916301098.

[11] R. W. Carlsen, M. Gidden, K. Huff, A. C. Opotowsky, O. Rakhimov, A. M. Scopatz, and P. Wilson, "Cycamore v1.0.0," *Figshare* (2014)Http://figshare.com/articles/Cycamore_v1_0_0/1041829.

[12] G. Reikard, "Predicting solar radiation at high resolutions: A comparison of time series forecasts," *Solar Energy*, **83**, *3*, 342 (2009).

[13] M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz, "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids," *Renewable and Sustainable Energy Reviews*, **27**, 65 (2013).

[14] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," *North American Power Symposium 2010*, 1–8, IEEE (2010).

[15] J. W. Taylor, P. E. McSharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and time series models," *IEEE Transactions on Energy Conversion*, **24**, *3*, 775 (2009).

[16] G. C. Souza, "Supply chain analytics," *Business Horizons*, **57**, *5*, 595 (2014).

[17] G. Chee, J. W. Bae, K. D. Huff, R. R. Flanagan, and R. Fairhurst, "Demonstration of Demand-Driven Deployment Capabilities in Cyclus," *Proceedings of Global/Top Fuel 2019*, American Nuclear Society, Seattle, WA, United States (2019).

[18] R. F. Engle, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica: Journal of the Econometric Society*, **50**, *4*, 987 (1982); 10.2307/1912773., URL `http://www.jstor.org/stable/1912773`.

[19] R. R. Flanagan, J. W. Bae, K. D. Huff, G. J. Chee, and R. Fairhurst, "Methods for Automated Fuel Cycle Facility Deployment," *Proceedings of Global/Top Fuel 2019*, American Nuclear Society, Seattle, WA, United States (2019).

[20] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," *Proceedings of the 9th Python in Science Conference*, vol. 57, 61, Scipy (2010).

[21] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier transform-algorithms and applications*, Springer Science & Business Media (2011).

[22] E. Jones, T. Oliphant, and P. Peterson, *SciPy: Open source scientific tools for Python, 2001* (2016).

[23] T. E. Oliphant, *A guide to NumPy*, vol. 1, Trelgol Publishing USA (2006).

[24] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, OTexts (2018).

[25] N. Sematech, "Engineering statistics handbook," *NIST SEMATECH* (2006).

[26] T. G. Smith, *pmdarima: Arima estimators for python* (2017).

[27] S. A. S. Institute, *SAS user's guide: statistics*, vol. 2, Sas Inst (1985).

[28] G. J. Chee, J. W. Bae, R. Fairhurst, R. R. Flanagan, and A. M. Scopatz, "arfc/d3ploy: A collection of Cyclus manager archetypes for demand driven deployment," (2019)URL `https://github.com/arfc/d3ploy`, 10.5281/zenodo.3464123.

[29] J. W. Bae, G. Chee, robfairh, K. Huff, T. Kennelly, P. . Speaks, K. Kleimenhagen, P. Wilson, G. Westphal, and A. Scopatz, "arfc/transition-scenarios: MS Thesis Release," (2019); 10.5281/zenodo.3569721., URL `https://zenodo.org/record/3569721`.

[30] United Nations Institute for Disarmament Research and Y. Yudin, *Multilateralization of the Nuclear Fuel Cycle: Helping to Fulfil the NPT Grand Bargain*, UN (2013); 10.18356/827477bb-en., URL

https://www.un-ilibrary.org/natural-resources-water-and-energy/multilateralization-of-the-nuclear-fuel-cycle_827477bb-en.

[31] J. W. Bae, J. L. Peterson-Droogh, and K. D. Huff, "Standardized verification of the Cyclus fuel cycle simulator," *Annals of Nuclear Energy*, **128**, 288 (2019); 10.1016/j.anucene.2019.01.014., URL http://www.sciencedirect.com/science/article/pii/S0306454919300179.

[32] J. Bae, B. Betzler, and A. Worrall, "Neural Network Approach to Model Mixed Oxide Fuel Cycles in Cyclus, a Nuclear Fuel Cycle Simulator," *Transactions of the American Nuclear Society - Volume 121*, 378–382, AMNS (2019); 10.13182/T31269., URL http://www.ans.org/pubs/transactions/a_47577.

[33] J. L. Peterson-Droogh and R. Gregg, "Value Added When Using Cross Sections for Fuel Cycle Analysis," , Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States) (2018).