

¹ Demand Driven Deployment Capabilities in Cyclus, a Fuel
² Cycle Simulator

³

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

⁴

Number of pages: 39

Number of tables: 7

Number of figures: 12

Abstract

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

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

23 **I. INTRODUCTION**

24 The nuclear fuel cycle represents the nuclear fuel life cycle from initial extraction through
25 processing, use in reactors, and, eventually, final disposal. This complex system of facilities and
26 mass flows collectively provide nuclear energy in the form of electricity [1]. Nuclear fuel cycle
27 simulator tools were introduced to investigate nuclear fuel cycle dynamics at a local and global level.
28 These simulators track the flow of materials through the nuclear fuel cycle, from enrichment to final
29 disposal of the fuel, while also accounting for decay and transmutation of isotopes. The impacts
30 are evaluated in the form of ‘metrics’, quantitative measures of performance [2]. These metrics are
31 calculated from mass balances and facility operation histories calculated by a fuel cycle simulator
32 [2]. By evaluating performance metrics of different fuel cycles, we gain an understanding of how
33 each facility’s parameters and technology choices impact the system’s performance. Therefore,
34 these results can be used to guide research efforts, advise future design choices, and provide
35 decision-makers with a transparent tool for evaluating fuel cycle options to inform policy decisions
36 [1].

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

49 **I.A. Context of Work**

50 The impact of climate change on natural and human systems is increasingly apparent [4].
51 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].

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

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

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

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

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

92 I.B. Novelty

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

99 **I.C. Objectives**

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

104 **II. METHODOLOGY**

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

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

`d3ploy` aims to minimize the undersupply of power:

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

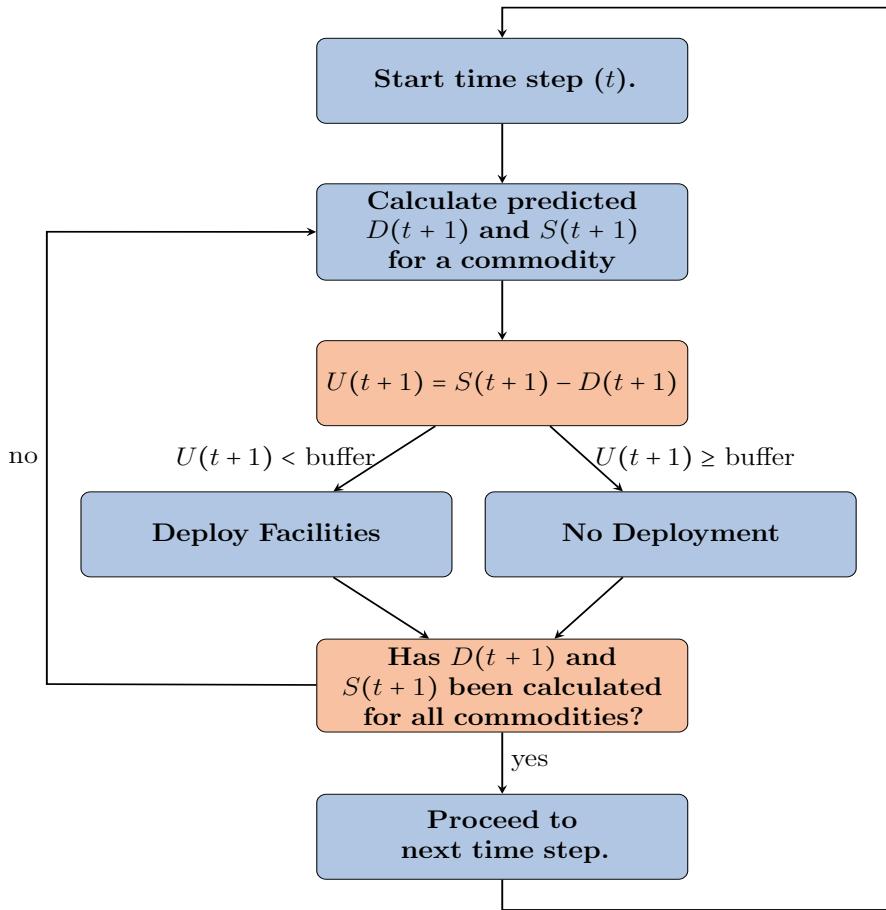


Fig. 1. d3ploy logic flow at every time step in CYCLUS [17].

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:

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

and to minimize excessive oversupply of all commodities:

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

where:

c = commodity type

M = Number of commodities

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

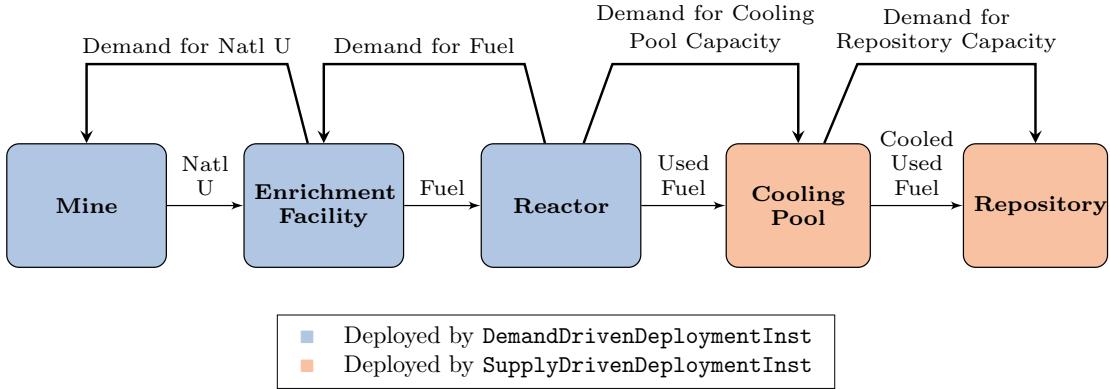


Fig. 2. Simple once-through fuel cycle depicting which facilities are deployed by `DemandDrivenDeploymentInst` and `SupplyDrivenDeploymentInst`.

124 **II.A. Structure**

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

134 *II.A.1. Deployment-Driving Method*

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

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

150 **II.B. Input Variables**

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

- 153 **1. The available facilities for `d3ploy` to deploy in the simulation and their respective
154 capacities.** Users must define the facilities they want `d3ploy` to deploy. It is the user's
155 responsibility to ensure the defined facilities create a supply chain to produce the demand
156 driving commodity.
- 157 **2. The demand driving commodity and its demand equation.** For most simulations,
158 the demand driving commodity is power. The demand equation is defined by a mathematical
159 equation with units of MW. For example, a constant power demand equation is 10000, while
160 a linearly increasing power demand equation is $100t$.
- 161 **3. The deployment driving method.** This input variable is described in Section II.A.1.
- 162 **4. The prediction method.** This input variable is described in Section II.D. There are also
163 optional input variables:
- 164 **5. Supply/capacity buffers for individual commodities.** This input variable is described
165 in section II.B.1.
- 166 **6. Facility preferences.** This input variable is described in section II.C.
- 167 **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
		Power: 3000 MW
	Supply/Capacity Buffer size	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.

168 *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:

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

or an absolute value:

$$S_{pwb} = S_p + b \quad (5)$$

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

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

172 II.C. Facility Preference and Fleet Share

173 The user can define time-dependent preference equations to facilities' that supply the same
174 commodity. If there are two reactor types, LWRs and Sodium-Cooled Fast Reactors (SFRs), in a
175 simulation, the user can make use of time-dependent preferences to make the simulation deploy
176 LWRs at earlier times in the simulation, and deploy SFRs at later times in the simulation when
177 there is a power demand. In Table II, the user defined that the LWR has a preference of $100 - t$,
178 while the SFR has a preference of $t - 99$. Figure 3 depicts how the preference for each reactor
179 changes with time. When there is a power undersupply, d3ploy will deploy the reactor that has a
180 larger preference at that time step. At time step 100, LWR preference is 0, while SFR preference is
181 1; therefore an SFR is deployed if there is a power shortage. Thus, the transition occurs at the
182 100th time step.

183 The user also has the option to specify percentage-share for facilities that provide the same
184 commodity. For example, if there are two reactor types, mixed oxide (MOX) LWRs and SFRs, in a
185 simulation, the user can make use of percentage-share specifications to determine the percentage
186 of power supplied by each reactor. When MOX LWR has a share of $s\%$ and SFR has a share of
187 $(100 - s)\%$, MOX LWR deployment constrains to $s\%$ of total power demand and SFR deployment
188 constrains to $(100 - s)\%$ of total power demand.

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

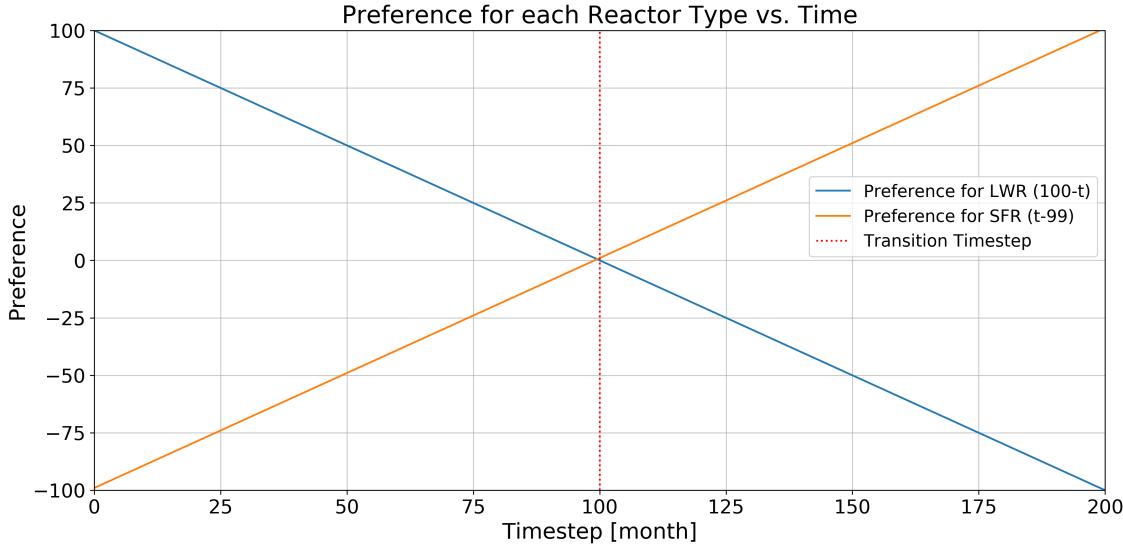


Fig. 3. `d3ploy` has a $100 - t$ preference for LWRs and a $t - 99$ preference for SFRs. When there is a power undersupply, `d3ploy` will deploy the reactor that has a larger preference at that time step.

¹⁹¹ Figure 4 shows the logical flow of which facility `d3ploy` deploys when there are multiple facilities
¹⁹² offering the same commodity.

¹⁹³ **II.D. Prediction Methods**

¹⁹⁴ `d3ploy` records supply and demand at each time step for all commodities. Time-series data
¹⁹⁵ informs `d3ploy`'s time series forecasting methods which predict future supply and demand for each
¹⁹⁶ commodity. The time series forecasting methods investigated include non-optimizing, deterministic-
¹⁹⁷ optimizing, and stochastic-optimizing methods. Non-optimizing methods are techniques that
¹⁹⁸ harness simple moving average and autoregression concepts which use historical data to infer
¹⁹⁹ future supply and demand values. Deterministic-optimizing and stochastic-optimizing methods are
²⁰⁰ techniques that use an assortment of more sophisticated time series forecasting concepts to predict
²⁰¹ future supply and demand values. Deterministic-optimizing methods give deterministic solutions,
²⁰² while stochastic-optimizing methods give stochastic solutions.

²⁰³ Depending on the scenario in question, each forecasting method offers distinct benefits and
²⁰⁴ disadvantages. The various methods are compared for each type of simulation to determine the
²⁰⁵ most effective prediction method for a given scenario. The following sections describe the prediction
²⁰⁶ methods.

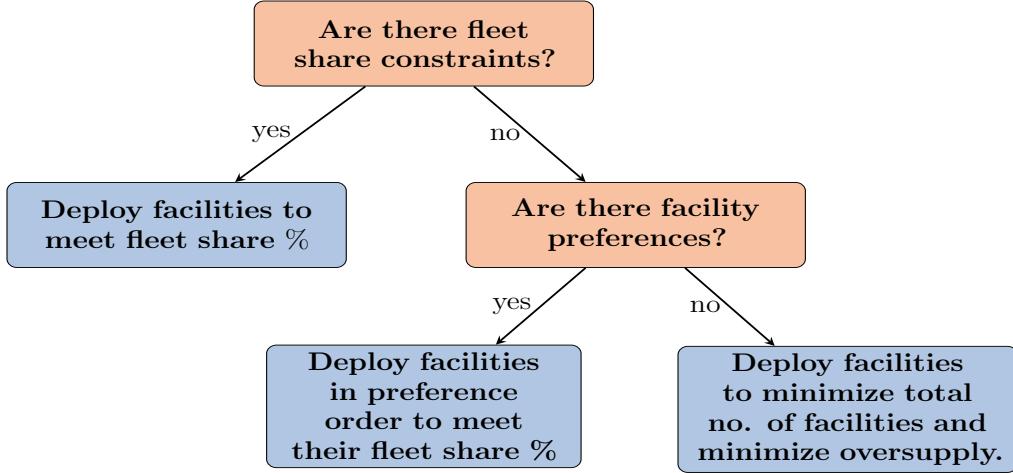


Fig. 4. Logical flow of how `d3ploy` selects which facility to deploy when there are multiple facilities offering the same commodity.

207 *II.D.1. Non-Optimizing Methods*

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

$$PV = \frac{\sum_{n=1}^N V_n}{n} \quad (6)$$

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:

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

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

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

217 *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]:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}. \quad (8)$$

where:

X = sequence of complex numbers

$k = 0, \dots, N - 1$

N = No. of complex numbers

x = sequence of complex numbers

$n = 0, \dots, N - 1$

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

219 The `POLY` method fits the time series data with a user-defined n^{th} degree polynomial and uses

220 the fitted trend-line to determine future demand and supply values:

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

where:

t = time index

n = polynomial order

β = fitted parameters

ε = unobserved random error

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

222 The `EXP-SMOOTHING` and `HOLT-WINTERS` methods use a weighted average of time-series data

223 with exponentially decaying weights for older time series values [24] to create a model to determine

224 future demand and supply values. The `EXP-SMOOTHING` method excels in modeling univariate time

225 series data without trend or seasonality [24]:

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

where:

y = timeseries value

α = smoothing factor ($0 < \alpha < 1$)

²²⁶ The **HOLT-WINTERS** method applies triple exponential smoothing, resulting in higher accuracy when
²²⁷ modeling seasonal time series data [25]:

$$F_{t+m} = (S_t + mb_t)I_{t-L+m} \quad (11)$$

$$S_t = \alpha \frac{y_t}{I_{t-L}} + (1 - \alpha)(S_{t-1} + b_{t-1})$$

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1}$$

$$I_t = \beta \frac{y_t}{S_t} + (1 - \beta)I_{t-L}$$

where:

F = forecast at m periods ahead

t = time period index

L = periods in a season

S = smoothed observation

y = the observation

b = trend factor

I = seasonal index

α, β, γ = constants

²²⁸ The StatsModels [20] Python package was used to implement the **EXP-SMOOTHING** and **HOLT-WINTERS**
²²⁹ methods in **d3ploy**.

230 **II.E. Stochastic-Optimizing Methods**

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

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

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

236 **III. RESULTS**

237 This section aims to demonstrate **d3ploy**'s capability to completely automate the setup of
238 transition scenarios and meet the objectives described in section [I.C](#). This section is split into two
239 subsections. The first subsection (section [III.A](#)) will demonstrate **d3ploy**'s capabilities to set up
240 a simple transition scenario with only three facility types. The second subsection (section [III.B](#))
241 will demonstrate **d3ploy**'s capabilities to set up complex EG01-23, EG01-24, EG01-29, EG01-30
242 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.

243 1. Section [III.B.1](#): compare the use of different `d3ploy` prediction methods in EG01-EG23,
 244 EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios,

245 2. Section [III.B.2](#): compare the use of varied power buffer sizes in EG01-EG23, EG01-EG24,
 246 EG01-EG29, and EG01-EG30 transition scenarios, and

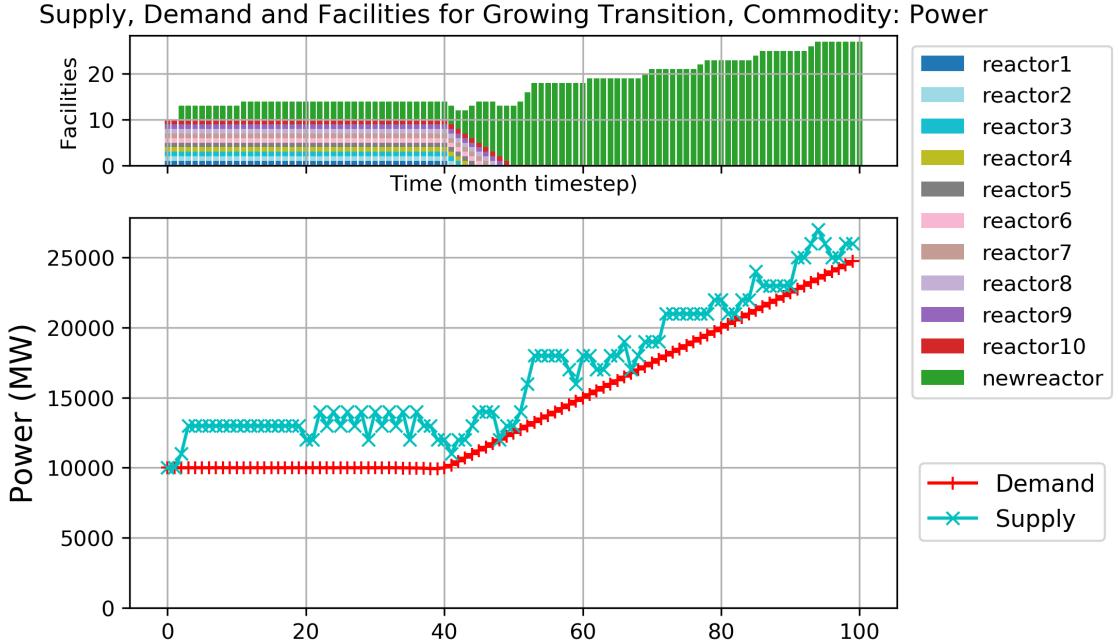
247 3. Section [III.B.3](#): demonstrate successful `d3ploy` setup of EG01-EG23, EG01-EG24, EG01-
 248 EG29, and EG01-EG30 transition scenarios using the prediction method and power buffer size
 249 that proved to best minimize power undersupply in the Sections [III.B.1](#) and [III.B.2](#). These
 250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

251 The input files and scripts to reproduce the results and plots in this paper are found in [\[28\]](#) and
 252 [\[29\]](#).

253 **III.A. Simple Transition Scenario**

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

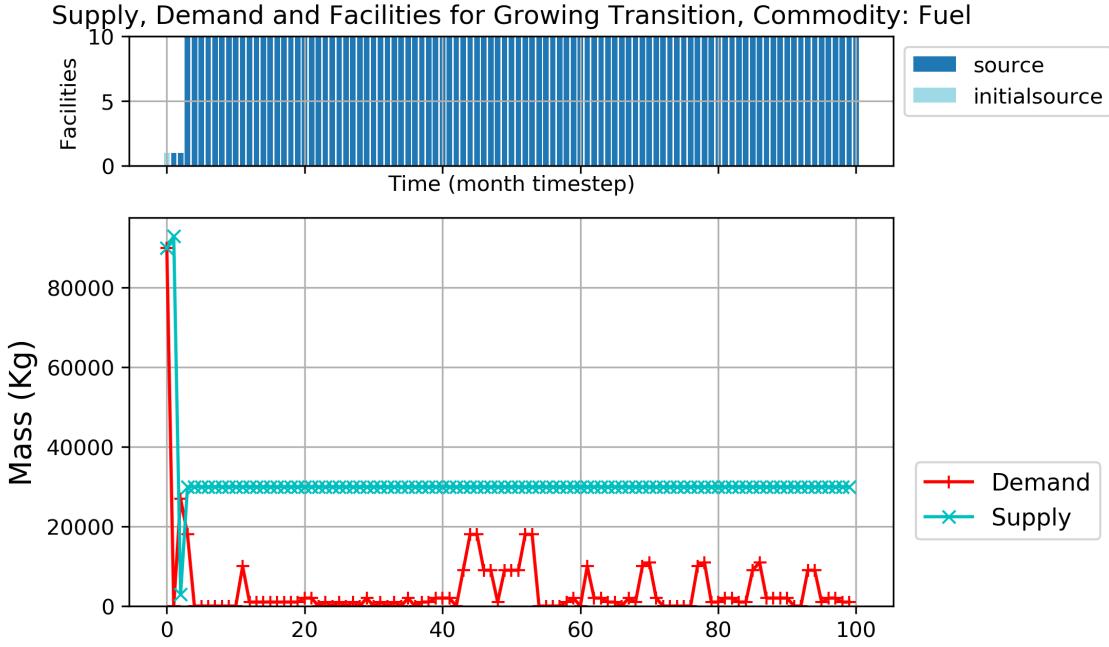
262 Figures [5\(a\)](#), [6\(a\)](#), and [6\(b\)](#) demonstrate `d3ploy`'s capability to deploy reactors and supporting
 263 facilities to minimize undersupply when meeting linearly increasing power demand and subsequent



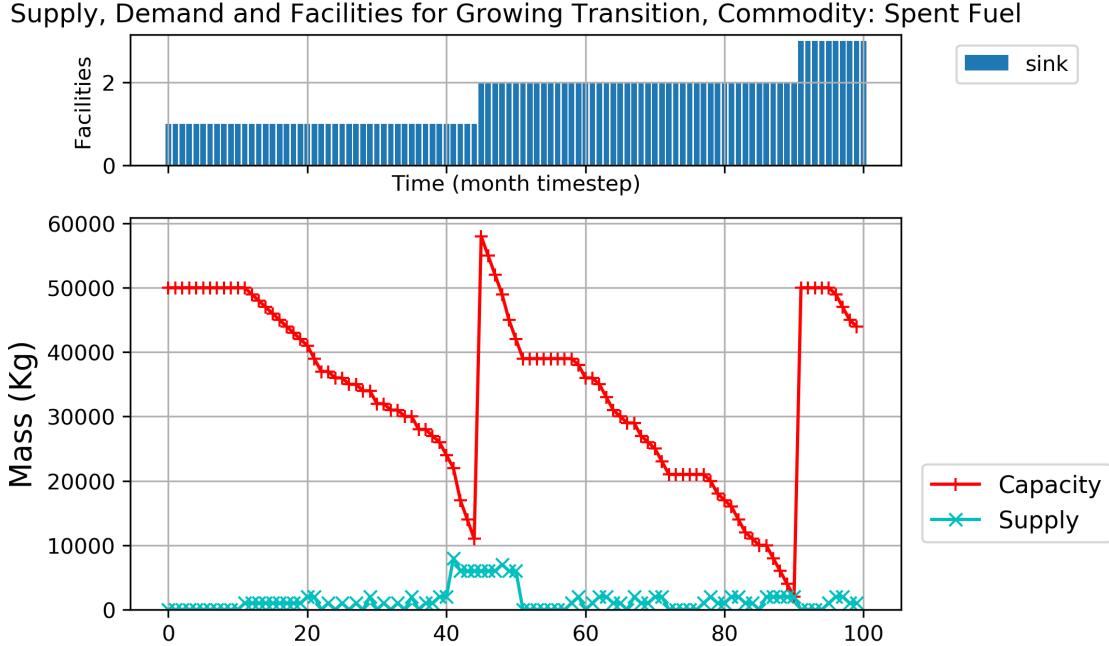
(a) Power demand and supply, and reactor facility deployment plot for a simple linearly increasing power demand transition scenario with three facility types: `source`, `reactor`, and `sink`. The simulation begins with `reactor1` to `reactor10` and `d3ploy` deploys `newreactors` to meet increasing power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

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

268 In figure 6(a), a large-throughput source facility is initially deployed to meet the large initial
 269 fuel demand for the commissioning of ten reactors. Deployment of a large-throughput source
 270 facility for the first few time steps ensures `d3ploy` does not deploy supporting facilities that become
 271 redundant at later times in the simulation. This reflects reality in which reactor manufacturers
 272 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.



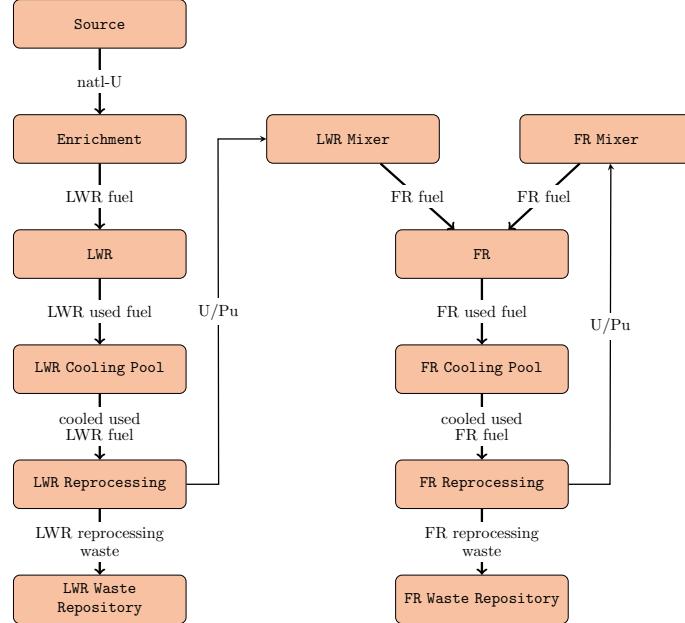
(b) Spent fuel capacity and supply, and sink facility deployment plot. Spent fuel is supplied by reactors and the capacity to store them is provided by sink facilities. There are no time steps with under-capacity of sink space.

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

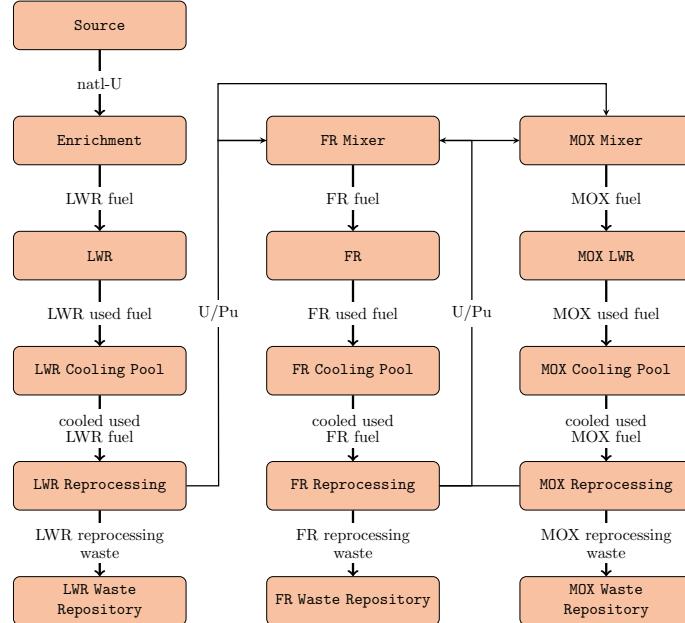
273 **III.B. Complex Transition Scenarios**

274 EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios are automatically
275 set up in CYCLUS using `d3ploy`. We defined the EG01-EG23 and EG01-EG29 transition scenario
276 simulations to have a constant power demand, while EG01-EG24 and EG01-EG30 have a linearly
277 increasing power demand. Similar to the simple transition scenario, these transition scenario
278 simulations begin with an initial fleet of LWRs that start progressively decommissioning at the
279 80-year mark, after which `d3ploy` deploys SFRs and MOX LWRs to meet the power demand.
280 Figure 7 shows the setup of facilities and mass flows for EG01-EG23 and EG01-EG29 in CYCLUS.
281 In EG01-EG23 and EG01-EG29, recycled plutonium from LWR spent fuel produces SFR fuel.
282 EG01-EG24 and EG01-EG30 are similar to EG01-EG23 and EG01-EG29, respectively, with the
283 exception that all transuranic elements are recycled.

284 The facilities used in the transition scenario simulations are described below. The source
285 facility has a throughput of 1e8kg of natural uranium, and the enrichment facility has a SWU
286 capacity of 1e100. The LWRs have an assembly size of 29863.3kg with 3 assemblies per core, and a
287 power capacity of 1000 MW. The FRs have an assembly size of 3950kg and power capacity of 333.34
288 MW. The MOX LWRs have an assembly size of 33130kg and power capacity of 1000 MW. The
289 `reactor` facility used in the CYCLUS simulation is a recipe reactor; it accepts a fresh fuel recipe
290 and outputs a spent fuel recipe. The recipes used for the LWR, MOX LWR, and SFR are based on
291 recipes generated by VISION [29] that closely match EG30 scenario specifications in Appendix B
292 of the Department of Energy (DOE) Evaluation and Screening Study (E&S study) [6]. The LWR,
293 FR, and MOX LWR cooling pools have a residence time of 36 months, and a max inventory size of
294 1e8kg of fuel. The reprocessing segment for each reactor type has a reprocessing and mixer facility.
295 Each reprocessing facility has a throughput of 1e8kg and separates U/Pu or U/TRU from other
296 isotope in spent fuel. Each mixer facility mixes the U/Pu or U/TRU to fabricate new reprocessed
297 fuel. `d3ploy` will deploy reprocessing facilities based on the demand of reprocessed fuel from the
298 MOX LWRs and FRs to ensure that sufficient fissile material feeds the reprocessing facilities to
299 make sufficient reprocessed fuel for each reactor type. Each waste repository is assumed to have
300 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.

301 *III.B.1. Comparison of Prediction Methods*

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

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

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

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

EG1-23: Time steps with an undersupply or under capacity of each commodity for different prediction methods

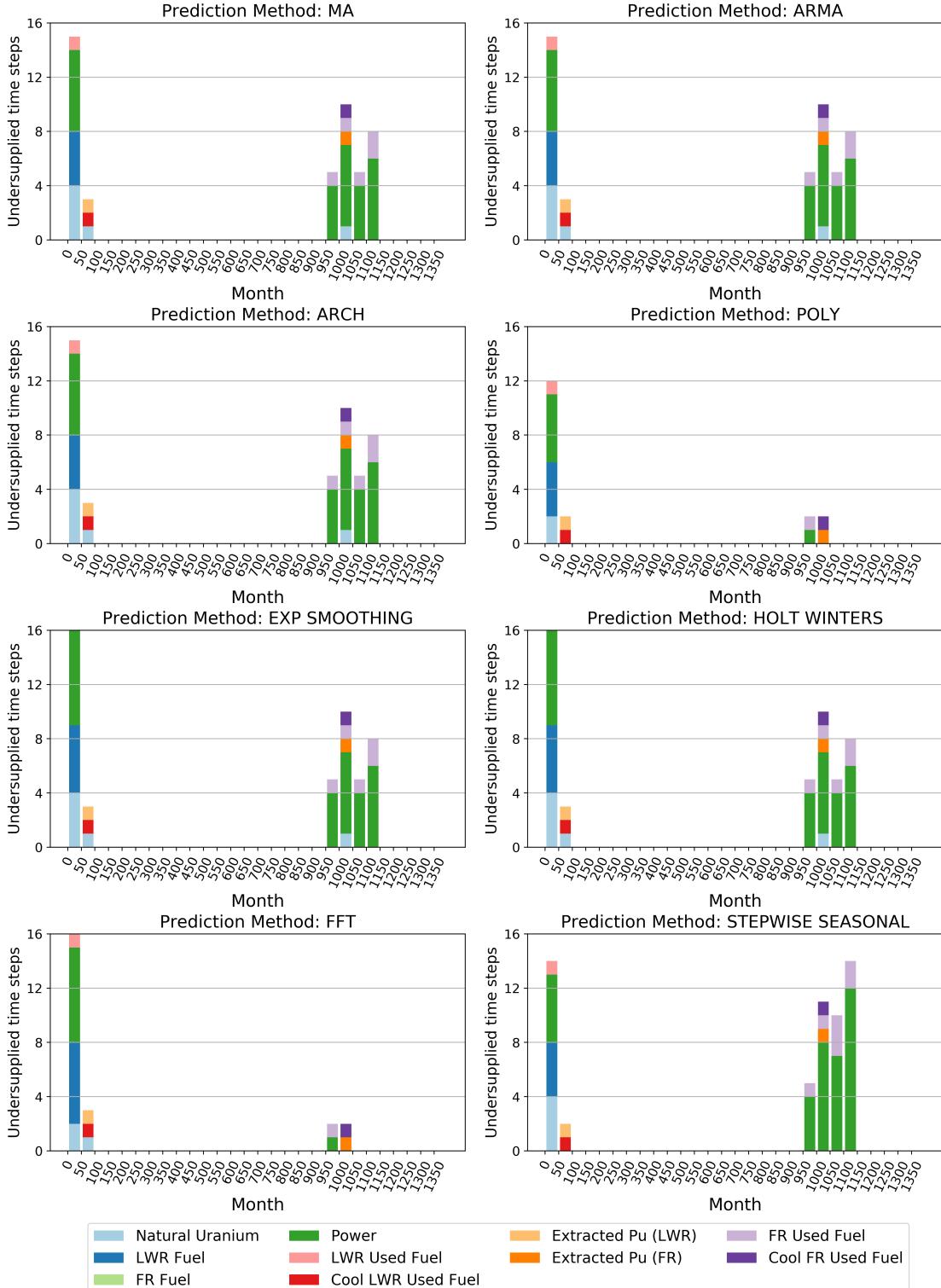


Fig. 8. EG01-EG23 transition scenario with constant power demand. Each subplot shows the total number of time steps in which there exists undersupply and under capacity of commodities for each prediction method. The different colors represent different commodities and each vertical bar refers to 50 time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

EG1-24: Time steps with an undersupply or under capacity of each commodity for different prediction methods

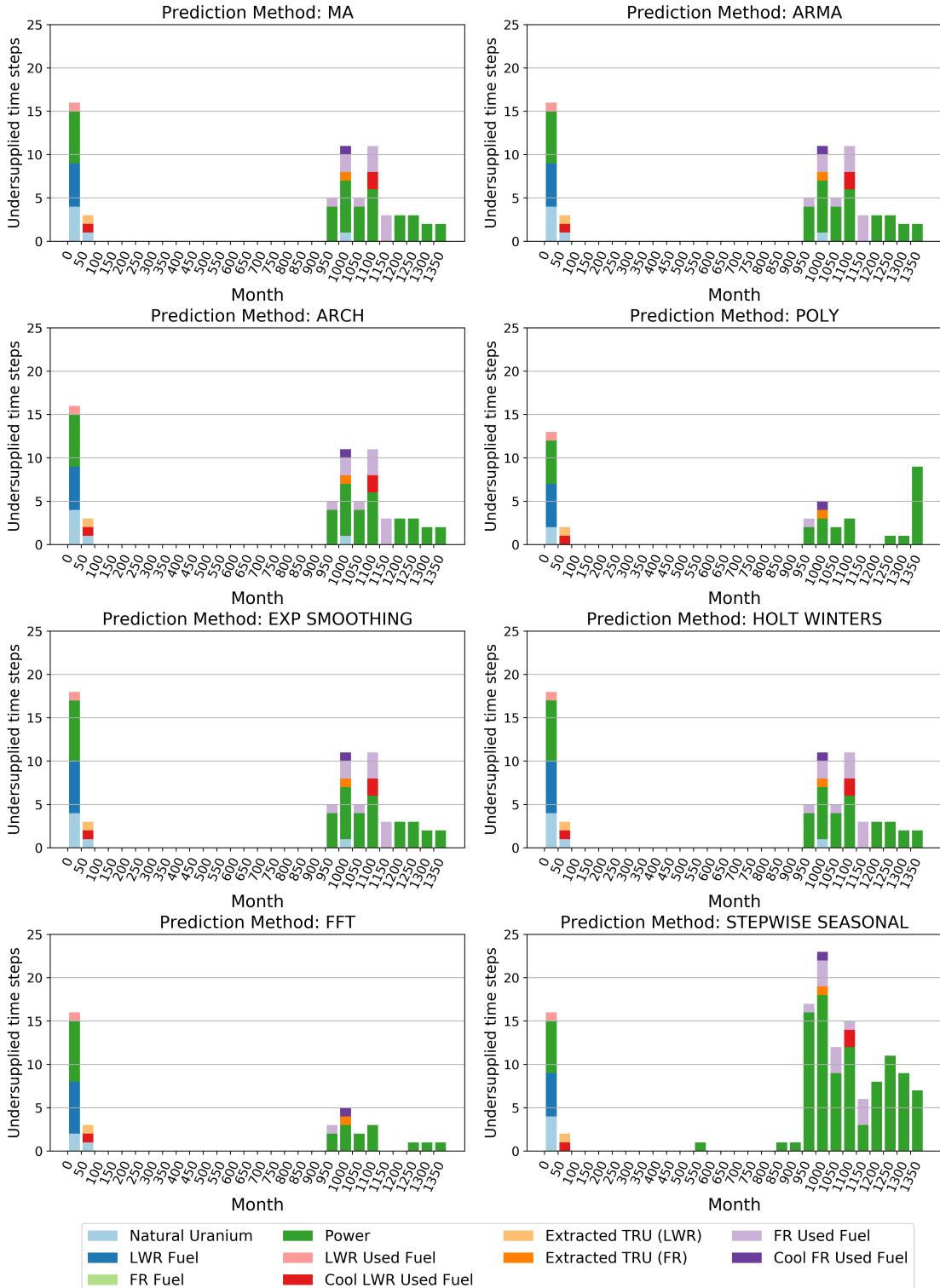


Fig. 9. EG01-EG24 transition scenario with linearly increasing power demand. Each subplot shows the total number of time steps in which there exists undersupply and under capacity of commodities for each prediction method. The different colors represent different commodities and each vertical bar refers to 50 time steps in the simulation. ²⁶

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

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

339 *III.B.2. Comparison of Power Buffer Sizes*

340 For the EG01-EG23, EG01-EG24, EG01-EG29, and EG01-EG30 transition scenarios, the
341 power buffer size is varied for the prediction method which minimizes commodity undersupply,
342 which is the **POLY** method for EG01-EG23 and EG01-EG29, and the **FFT** method for EG01-EG24
343 and EG01-EG30. Varying the power buffer size does not impact the number of undersupplied
344 time steps for the EG01-EG23 and EG01-EG29 constant power demand transition scenarios. In
345 Table V, there are 6 and 4 time steps in which there is power undersupply for the EG01-EG23
346 and EG01-EG29 transition scenarios, respectively. As seen in figure 8, these undersupply time
347 steps occur at the beginning of the simulation and for one time step when the transition begins.
348 We expected this since without time-series data at the beginning of the simulation, **d3ploy** takes
349 a few time steps to collect time-series data about power demand to predict and start deploying
350 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

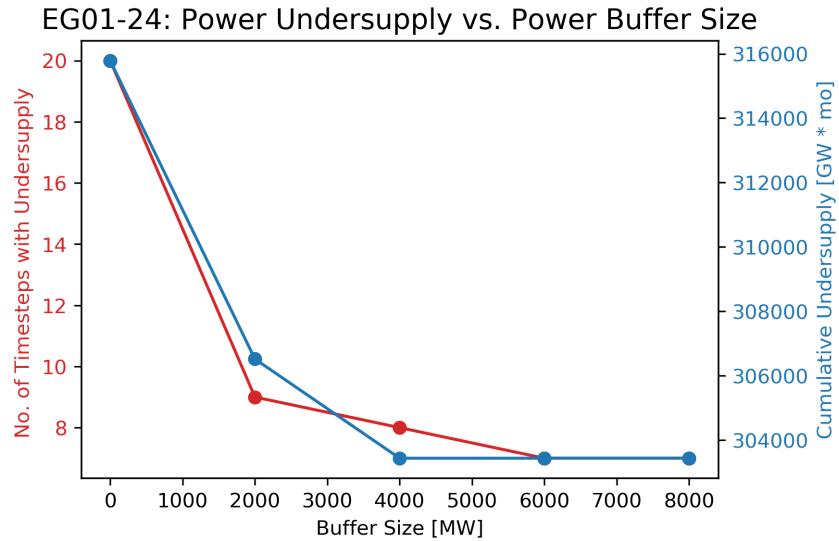
Undersupply/capacity of key commodities for the best performing EG01-EG23, EG24, EG29, EG30 transition scenarios.

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

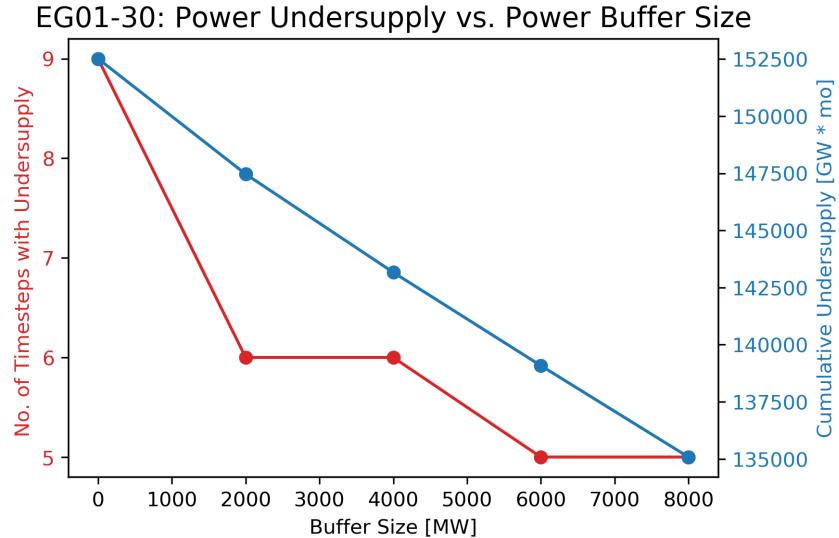
354 We varied the power buffer size for the EG01-EG24 and EG01-EG30 linearly increasing power
 355 demand transition scenarios. Figures 10(a), 10(b), and Table VI show that increasing the buffer
 356 size increases the robustness of the supply chain by minimizing power undersupply. The cumulative
 357 undersupply is minimized with a 6000MW and 8000MW buffer for EG01-EG24 and EG01-EG30
 358 respectively. In Figure 10(a), a 4000MW buffer size has 8 time steps with undersupply, while a
 359 6000MW buffer size has 7 time steps with undersupply. In Figure 10(b), a 2000MW buffer size has
 360 6 time steps with undersupply, while a 8000MW buffer size has 5 time steps with undersupply. We
 361 determined that extra commissioning of multiple reactors does not justify a single time step with
 362 no undersupply. This type of logic is difficult to program into a NFC simulator, therefore, even
 363 though NFC simulators can help inform policy decisions, decision-makers must still evaluate NFC
 364 simulator results to determine if they are valid and logical. Therefore, a buffer of 4000MW and
 365 2000MW minimizes the power undersupply for EG01-EG24 and EG01-EG30 transition scenarios,
 366 respectively.

367 *III.B.3. Best Performance Models*

368 The ‘best performance models’ for each transition scenario use the prediction method and
 369 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.

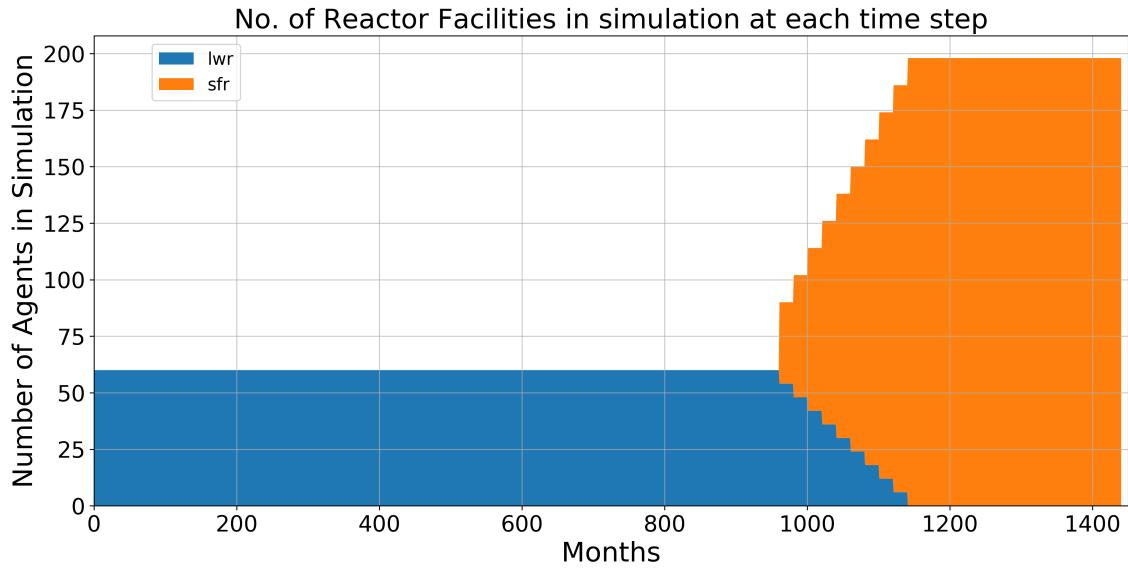
Buffer [MW]	Undersupply	EG01-EG24	EG01-EG30
0	Time steps [#] Energy [$GW \cdot mo$]	20 315791	9 152517
2000	Undersupplied [#] Energy [$GW \cdot mo$]	9 306520	6 147166
4000	Time steps [#] Energy [$GW \cdot mo$]	8 303438	6 143166
6000	Time steps [#] Cumulative [GW]	7 303438	5 139083
8000	Time steps [#]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

TABLE VI

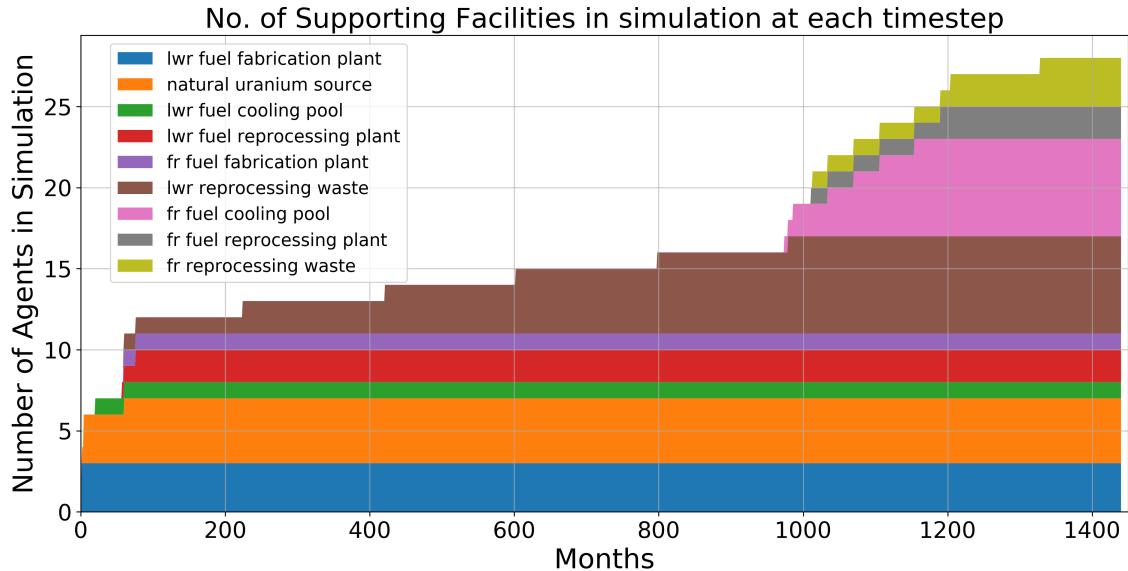
The effect of sensitivity analysis of power buffer size on cumulative undersupply of power for EG01-EG24 and EG01-EG30 transition scenarios with linearly increasing power demand using the FFT prediction method.

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

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

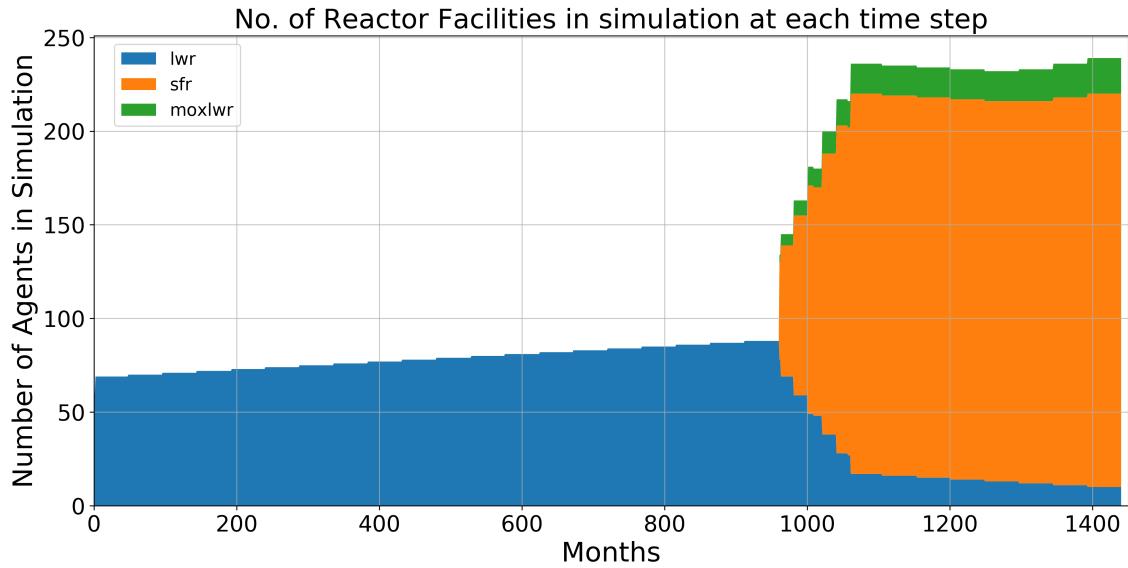


(a) EG01-EG23: Reactor Deployment

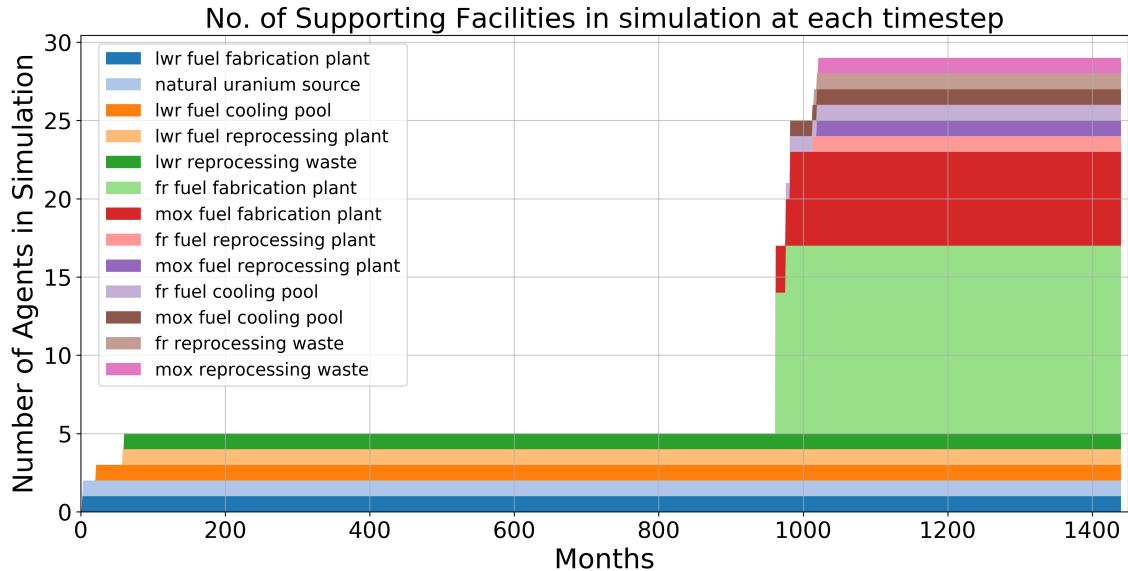


(b) EG01-EG23: Supporting Facility Deployment

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 +250t/12	60000 +250t/12	60000 +250t/12	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.

³⁸² **IV. CONCLUSION**

³⁸³ The present nuclear fuel cycle in the United States is a once-through fuel cycle of LWRs
³⁸⁴ with no used fuel reprocessing. This nuclear fuel cycle faces cost, safety, proliferation, and spent
³⁸⁵ nuclear fuel challenges that hinder large-scale nuclear power deployment. The U.S Department of
³⁸⁶ Energy identified future nuclear fuel cycles, involving continuous recycling of co-extracted U/Pu
³⁸⁷ or U/TRU in fast and thermal spectrum reactors, that may overcome these challenges. These
³⁸⁸ transition scenarios have been modeled previously in the following nuclear fuel cycle simulators
³⁸⁹ [10, 31]: ORION, DYMOND, VISION, MARKAL, and CYCLUS. However, for many nuclear fuel
³⁹⁰ cycle simulators, the user is required to define a deployment scheme for all supporting facilities to
³⁹¹ avoid any supply chain gaps or resulting idle reactor capacity. Manually determining a deployment
³⁹² scheme for a once-through fuel cycle is straightforward; however, for complex fuel cycle scenarios, it
³⁹³ is not. In this paper, we introduce the capability, d3ploy, in CYCLUS that automatically deploys fuel
³⁹⁴ cycle facilities to meet user-defined power demand. In this paper, we demonstrate that with careful
³⁹⁵ selection of d3ploy parameters, we can completely automate the setup of constant and linearly
³⁹⁶ increasing power demand transition scenarios for EG01-23, EG01-24, EG01-29, and EG01-30 with
³⁹⁷ minimal power undersupply. Using d3ploy to set up transition scenarios saves the user simulation

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

405 **V. FUTURE WORK**

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

420 **VI. ACKNOWLEDGMENTS**

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

426 collaborative CYCLUS development.

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

437 **REFERENCES**

438 [1] A. M. YACOUT, J. J. JACOBSON, G. E. MATTHERN, S. J. PIET, and A. MOISSEYTSEV,
439 “Modeling the Nuclear Fuel Cycle,” *The 23rd International Conference of the System Dynam-
440 ics Society*, Boston, Citeseer (2005)URL <http://www.inl.gov/technicalpublications/Documents/3169906.pdf>.

442 [2] K. D. HUFF, M. J. GIDDEN, R. W. CARLSEN, R. R. FLANAGAN, M. B. MCGARRY, A. C.
443 OPOTOWSKY, E. A. SCHNEIDER, A. M. SCOPATZ, and P. P. H. WILSON, “Fundamental
444 concepts in the Cyclus nuclear fuel cycle simulation framework,” *Advances in Engineering
445 Software*, **94**, 46 (2016); 10.1016/j.advengsoft.2016.01.014., URL <http://www.sciencedirect.com/science/article/pii/S0965997816300229>, arXiv: 1509.03604.

447 [3] K. D. HUFF, J. W. BAE, K. A. MUMMAH, R. R. FLANAGAN, and A. M. SCOPATZ, “Current
448 Status of Predictive Transition Capability in Fuel Cycle Simulation,” *Proceedings of Global
449 2017*, 11, American Nuclear Society, Seoul, South Korea (2017).

450 [4] *Climate Change and Nuclear Power 2018*, Non-serial Publications, INTERNATIONAL
451 ATOMIC ENERGY AGENCY, Vienna (2018)URL <https://www.iaea.org/publications/13395/climate-change-and-nuclear-power-2018>.

453 [5] “The Future of Nuclear Energy in a Carbon-Constrained World,” *Massachusetts Institute of
454 Technology Energy Initiative (MITEI)*, 272 (2018).

455 [6] R. WIGELAND, T. TAIWO, H. LUDEWIG, M. TODOSOW, W. HALSEY, J. GEHIN, R. JUBIN,
456 J. BUEL, S. STOCKINGER, K. JENNI, and B. OAKLEY, “Nuclear Fuel Cycle Evaluation
457 and Screening – Final Report,” INL/EXT-14-31465, U.S. Department of Energy (2014)URL
458 <https://fuelcycleevaluation.inl.gov>.

459 [7] “New DOE Nuclear Energy Chief Suggests Rethinking Spent Fuel Reprocessing
460 - 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.

463 [8] M. CRAPO, "S.97 - 115th Congress (2017-2018): Nuclear Energy Innovation Capabilities Act
464 of 2017," (2018)URL <https://www.congress.gov/bill/115th-congress/senate-bill/97>,
465 archive Location: 2017/2018 Library Catalog: www.congress.gov.

466 [9] R. E. LATTA, "H.R.590 - 115th Congress (2017-2018): Advanced Nuclear Technology De-
467 velopment Act of 2017," (2017)URL <https://www.congress.gov/bill/115th-congress/house-bill/590>, archive Location: 2017/2018 Library Catalog: www.congress.gov.

469 [10] B. FENG, B. DIXON, E. SUNNY, A. CUADRA, J. JACOBSON, N. R. BROWN, J. POWERS,
470 A. WORRALL, S. PASSERINI, and R. GREGG, "Standardized verification of fuel cycle modeling,"
471 *Annals of Nuclear Energy*, **94**, 300 (2016); 10.1016/j.anucene.2016.03.002., URL <http://www.sciencedirect.com/science/article/pii/S0306454916301098>.

473 [11] R. W. CARLSEN, M. GIDDEN, K. HUFF, A. C. OPOTOWSKY, O. RAKHI-
474 MOV, A. M. SCOPATZ, and P. WILSON, "Cycamore v1.0.0," *Figshare*
475 (2014)Http://figshare.com/articles/Cycamore_v1_0_0/1041829.

476 [12] G. REIKARD, "Predicting solar radiation at high resolutions: A comparison of time series
477 forecasts," *Solar Energy*, **83**, 3, 342 (2009).

478 [13] M. DIAGNE, M. DAVID, P. LAURET, J. BOLAND, and N. SCHMUTZ, "Review of solar irradiance
479 forecasting methods and a proposition for small-scale insular grids," *Renewable and Sustainable
480 Energy Reviews*, **27**, 65 (2013).

481 [14] S. S. SOMAN, H. ZAREPOUR, O. MALIK, and P. MANDAL, "A review of wind power and wind
482 speed forecasting methods with different time horizons," *North American Power Symposium
483 2010*, 1–8, IEEE (2010).

484 [15] J. W. TAYLOR, P. E. MCSHARRY, and R. BUIZZA, "Wind power density forecasting using
485 ensemble predictions and time series models," *IEEE Transactions on Energy Conversion*, **24**,
486 3, 775 (2009).

487 [16] G. C. SOUZA, "Supply chain analytics," *Business Horizons*, **57**, 5, 595 (2014).

488 [17] G. CHEE, J. W. BAE, K. D. HUFF, R. R. FLANAGAN, and R. FAIRHURST, "Demonstration
489 of Demand-Driven Deployment Capabilities in Cyclus," *Proceedings of Global/Top Fuel 2019*,
490 American Nuclear Society, Seattle, WA, United States (2019).

491 [18] R. F. ENGLE, “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance
 492 of United Kingdom Inflation,” *Econometrica: Journal of the Econometric Society*, **50**, 4, 987
 493 (1982); 10.2307/1912773., URL <http://www.jstor.org/stable/1912773>.

494 [19] R. R. FLANAGAN, J. W. BAE, K. D. HUFF, G. J. CHEE, and R. FAIRHURST, “Methods for
 495 Automated Fuel Cycle Facility Deployment,” *Proceedings of Global/Top Fuel 2019*, American
 496 Nuclear Society, Seattle, WA, United States (2019).

497 [20] S. SEABOLD and J. PERKTOLD, “Statsmodels: Econometric and statistical modeling with
 498 python,” *Proceedings of the 9th Python in Science Conference*, vol. 57, 61, Scipy (2010).

499 [21] K. R. RAO, D. N. KIM, and J. J. HWANG, *Fast Fourier transform-algorithms and applications*,
 500 Springer Science & Business Media (2011).

501 [22] E. JONES, T. OLIPHANT, and P. PETERSON, *SciPy: Open source scientific tools for Python*,
 502 2001 (2016).

503 [23] T. E. OLIPHANT, *A guide to NumPy*, vol. 1, Trelgol Publishing USA (2006).

504 [24] R. J. HYNDMAN and G. ATHANASOPOULOS, *Forecasting: principles and practice*, OTexts
 505 (2018).

506 [25] N. SEMATECH, “Engineering statistics handbook,” *NIST SEMATECH* (2006).

507 [26] T. G. SMITH, *pmdarima: Arima estimators for python* (2017).

508 [27] S. A. S. INSTITUTE, *SAS user’s guide: statistics*, vol. 2, Sas Inst (1985).

509 [28] G. J. CHEE, J. W. BAE, R. FAIRHURST, R. R. FLANAGAN, and A. M. SCOPATZ, “arfcd3ploy:
 510 A collection of Cyclus manager archetypes for demand driven deployment,” (2019)URL
 511 <https://github.com/arfcd3ploy>, 10.5281/zenodo.3464123.

512 [29] J. W. BAE, G. CHEE, ROBFAIRH, K. HUFF, T. KENNELLY, P. . SPEAKS, K. KLEIMENHAGEN,
 513 P. WILSON, G. WESTPHAL, and A. SCOPATZ, “arfcd/transition-scenarios: MS Thesis Release,”
 514 (2019); 10.5281/zenodo.3569721., URL <https://zenodo.org/record/3569721>.

515 [30] UNITED NATIONS INSTITUTE FOR DISARMAMENT RESEARCH and
 516 Y. YUDIN, *Multilateralization of the Nuclear Fuel Cycle: Helping to Ful-*
 517 *fil the NPT Grand Bargain*, UN (2013); 10.18356/827477bb-en., URL

518 [https://www.un-ilibrary.org/natural-resources-water-and-energy/
519 multilateralization-of-the-nuclear-fuel-cycle_827477bb-en.](https://www.un-ilibrary.org/natural-resources-water-and-energy/multilateralization-of-the-nuclear-fuel-cycle_827477bb-en)

520 [31] J. W. BAE, J. L. PETERSON-DROOGH, and K. D. HUFF, “Standardized verification
521 of the Cyclus fuel cycle simulator,” *Annals of Nuclear Energy*, **128**, 288 (2019);
522 10.1016/j.anucene.2019.01.014., URL <http://www.sciencedirect.com/science/article/pii/S0306454919300179>.

524 [32] J. BAE, B. BETZLER, and A. WORRALL, “Neural Network Approach to Model Mixed Oxide
525 Fuel Cycles in Cyclus, a Nuclear Fuel Cycle Simulator,” *Transactions of the American Nuclear
526 Society - Volume 121*, 378–382, AMNS (2019); 10.13182/T31269., URL http://www.ans.org/pubs/transactions/a_47577.

528 [33] J. L. PETERSON-DROOGH and R. GREGG, “Value Added When Using Cross Sections for Fuel
529 Cycle Analysis,” , Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States) (2018).