# Transition Scenario Demonstrations of CYCAMORE Demand Driven Deployment Capabilities

*Authors*
Gwendolyn CHEE
Roberto FAIRHURST

*Supervisor*
Kathryn D. HUFF

**UIUC-ARFC-2019-03**

June 29, 2019

ADVANCED REACTORS AND FUEL CYCLES
DEPT. OF NUCLEAR, PLASMA, & RADIOLOGICAL ENGINEERING
UNIVERSITY OF ILLIOIS AT URBANA-CHAMPAIGN

*This research was performed using funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Program under award number 16-10512.*

# 1 Introduction

In many fuel cycle simulators, the user must define a deployment scheme for all supporting facilities to avoid supply chain gaps. To ease setting up nuclear fuel cycle simulations, Nuclear Fuel Cycle (NFC) simulators should bring demand responsive deployment decisions into the dynamics of the simulation logic [3]. Thus, a next generation NFC simulator should predictively and automatically deploy fuel cycle facilities to meet a user defined power demand.

CYCLUS is an agent-based nuclear fuel cycle simulation framework [4]. In CYCLUS, each entity (i.e. Region, Institution, or Facility) in the fuel cycle is an agent. Region agents represent geographical or political areas that institution and facility agents can be grouped into. Institution agents control the deployment and decommission of facility agents and represents legal operating organizations such as a utility, government, etc. [4]. Facility agents represent nuclear fuel cycle facilities. CYCAMORE [1] provides agents to represent process physics of various components in the nuclear fuel cycle (e.g. mine, fuel enrichment facility, reactor).

The Demand-Driven CYCAMORE Archetypes project (NEUP-FY16-10512) aims to develop CYCLUS' demand-driven deployment capabilities. This capability is added as a CYCLUS Institution agent that deploys facilities to meet the front-end and back-end fuel cycle demands based on a user-defined commodity demand. This demand-driven deployment capability is called `d3ploy`.

In this paper, we explain the capabilities of `d3ploy` and demonstrate how `d3ploy` minimizes undersupply of all commodities in a few simulations while meeting key simulation constraints. Constant, linearly increasing, and sinusoidal power demand transition scenarios are demonstrated. Insights are discussed to inform parameter input decisions for future work in setting up larger transition scenarios that include many facilities. And finally, the more complex transition scenarios are demonstrated.

# 2 D3ploy capabilities

## 2.1 Core Capability of `d3ploy`

At each time step, `d3ploy` predicts demand and supply of each commodity for the next time step. Then, `d3ploy` deploys facilities to meet predicted demand. `D3ploy`'s primary objective is minimizing the number of time steps of undersupply of any commodity.

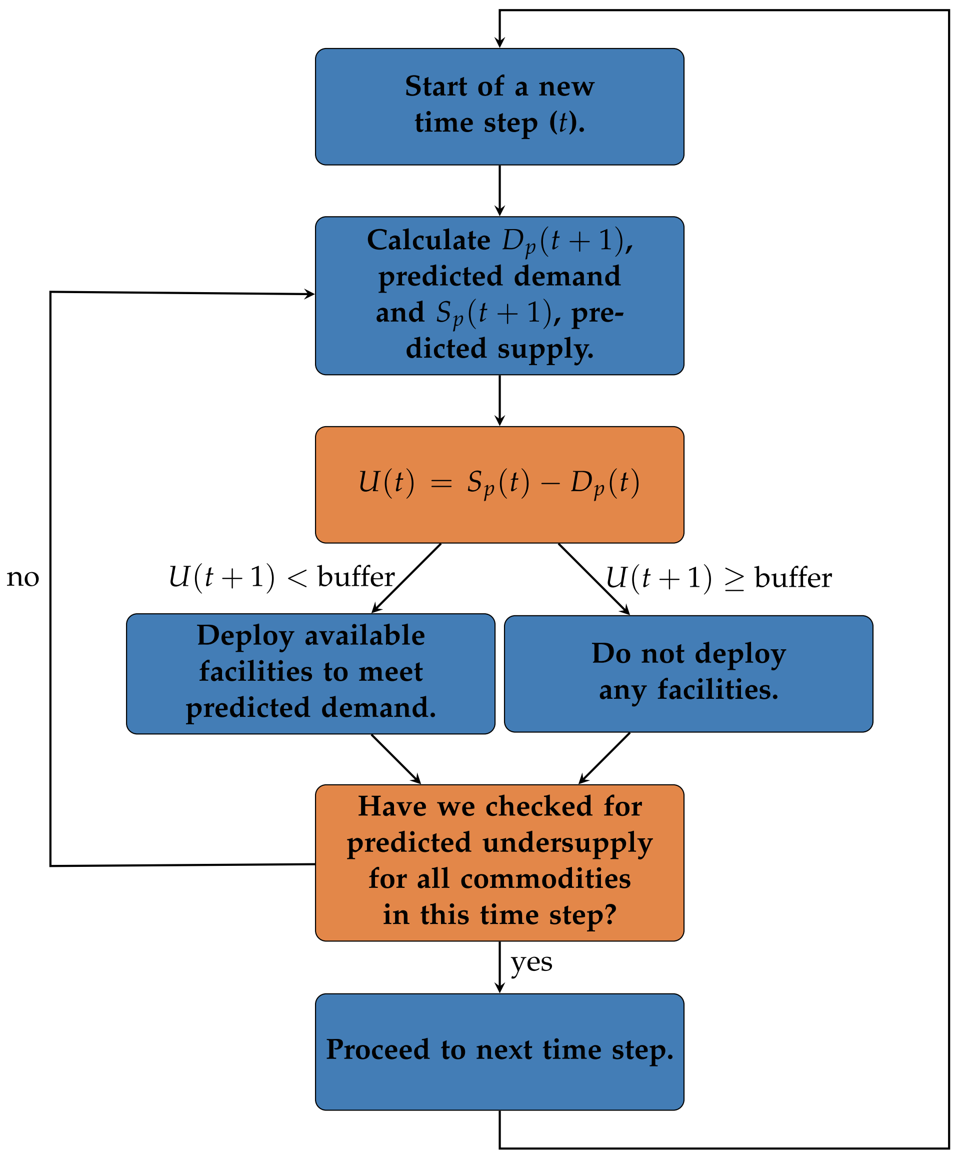



Figure 1: `D3ploy` logic flow at each time step in CYCLUS.

When `d3ploy` predicts an undersupply, it responds by deploying the fewest number of available facilities to meet demand with minimal oversupply. This logic is available in `solver.py`.

## 2.2 Basic User-Defined Input Variables

The user provides specific variables to customize their simulation. Descriptions of each input variable can be found in the README of the `d3ploy` github repository.

Essentially, the user must define the facilities the `d3ploy` institution controls and can deploy. The user must also define the driving commodity, all facility capacities for producing that commodity, its demand equation, and which method predicts supply and demand. For example, the user can define a demand equation for power of $D_p(t) = 1000t$ MW and `d3ploy` will deploy available reactor and supporting facilities to meet the defined power demand at each timestep, $t$.

The user can also provide a time-dependent equation that governs preference for a particular facility compared to other facilities which provide the same commodity. For example, the user can define a Light Water Reactor (LWR) and a Sodium-Cooled Fast Reactor (SFR) to have preferences of $p_{LWR}(t) = 101 - t$ and $p_{SFR}(t) = t$ respectively. The institution will prefer deployment of LWR facilities over SFR before time step 51, when $p_{SFR}(t)$ becomes greater than $p_{SFR}(t)$.

The user can constrain facility deployment until a facility accumulates enough inventory of a specific commodity. The user can also define an initial facility list of facilities to be present in the institution at the beginning of the simulation.

## 2.3 Prediction Algorithms

Three interchangeable algorithm types govern demand and supply predictions: non-optimizing (NO), deterministic optimizing (DO), and stochastic optimizing (SO).

There are three methods implemented for the non-optimizing model: Moving Average (MA), autoregressive moving average (ARMA), and autoregressive conditional heteroskedasticity (ARCH). There are four methods implemented for the deterministic optimizing model: Polynomial fit regression (POLY), simple exponential smoothing (EXP_SMOOTHING), triple exponential smoothing (HOLT_WINTERS) and fast fourier transform (FFT). There is one method implemented for stochastic optimizing model: stepwise seasonal (SW_SEASONAL).

The user can choose which prediction algorithm governs each specific `d3ploy` commodity. The effectiveness of a prediction algorithm depends on the type of power demand in a scenario and the type of commodity (demand driving commodity vs non-driving commodity, demand driven deployment vs supply driven deployment etc.). For example, the most effective method for predicting demand and supply for the power commodity in a scenario with a sinusoidal power demand is the triple exponential smoothing method. However, for the non-driving commodities in the same scenario, the fast fourier transform method is more effective than triple exponential smoothing. This paper will comment on these categories of problems and their suitable algorithms.

### 2.4 Demand-driven vs. Supply-driven Institutions

`D3ploy` defines two institutions: `DemandDrivenDeploymentInst` and `SupplyDrivenDeploymentInst`. The prior exists for the front-end of the fuel cycle and the latter for the back-end. For example, for front end facilities, the reactor demands fuel and `DemandDrivenDeploymentInst` triggers the deployment of fuel fabrication facilities to create supply meeting the demand for fuel. For back end facilities, the reactor generates spent fuel and `SupplyDrivenDeploymentInst` triggers the deployment of waste repository facilities to create capacity for storage of the supply of spent fuel.

### 2.5 Installed Capacity

The user can choose between deploying facilities based on the difference between predicted demand and predicted supply or predicted demand and installed capacity. Two advantages make preferable to use installed capacity over predicted supply. The first is for facilities that provide intermittent supply, such as a reactor facility with a designated refueling time. During time steps in which a reactor is refueling, the user might not want `d3ploy` to deploy more facilities to make up for the lack of supply caused by this one time step gap in supply. The second is for situations in which the input commodity for a facility has run out and the facility that produces the input commodity is no longer commissionable. Therefore, with the demand for the output commodity of that facility, `d3ploy` would deploy that facility to meet the demand, however due to the lack of the input commodity, even if there are infinite numbers of that facility, it will not produce the output commodity. For example, in a transition scenario from LWRs to fast reactors, the fast reactor demand for Pu may exceed the inventory provided by LWRs before they were decommissioned. This will result in the deployment of mixer facilities that generate the fast reactor fuel despite the lack of plutonium to generate the fuel. This can be avoided by constraining fast reactor facility deployment until a sizable inventory of Pu is accumulated.

### 2.6 Supply/Capacity Buffer

In `DemandDrivenDeploymentInst`, the user can choose to provide a buffer for predicted supply. `D3ploy` will deploy facilities to meet the predicted demand with the additional buffer.

In `SupplyDrivenDeploymentInst`, the user can choose to provide a buffer for predicted capacity. `D3ploy` will deploy facilities to meet the predicted supply with the additional buffer. The buffer can be defined as a percentage value (equation 1) or an absolute value (equation 2):

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

$$S_{pwb} = S_p + a \tag{2}$$

where $S_{pwb}$ is predicted supply/capacity with buffer, $S_p$ is the predicted supply/capacity without buffer, $d$ is the percentage value in decimal form, and $a$ is the absolute value of the buffer.

## 3 Demonstration of d3ploy capabilities

Constant, linearly increasing, and sinusoidal power demand simulations are shown to demonstrate `d3ploy`'s capabilities. A balance between the various system parameters must be met for each type of simulation to meet the goal of minimizing undersupply and under capacity for the various commodities. The input files and scripts to produce the plots in this paper can be reproduced using [2].

These simulations are basic transition scenarios that only include three types of facilities: `source`, `reactor` and `sink`. All simulations in this work begin with a ten reactor facilities, `reactor1` to `reactor10`. These reactors have staggered cycle lengths and lifetimes so that they do not all refuel and decommission at the same time steps. When the ten initial reactor facilities begin to decommission, `d3ploy` deploys reactor facilities of `newreactor` type to meet unmet demand for power. All the simulations deploy facilities based on the relationship between predicted demand and installed capacity. This capability was discussed in the previous section. Table 1 shows the simulation parameters that are consistent across all the discussed scenarios.

Table 1: Transition scenario parameters that are consisted for constant, linear increasing and sinusoidal power demand simulations

| **Parameters** | **Description** |
|---|---|
| Facilities Present | `Source` (Capacity: 3000kg)<br>`Reactor` (Capacity: 1000MW)<br>`Sink` (Capacity: 50000kg) |
| New Reactor Parameters | Cycle time: 18<br>Refuel time: 1 |
| Driving Commodity | Power |

These basic transition scenarios were set up to demonstrate `d3ploy`'s capabilities for simulating transition scenarios and to inform decisions about input parameters when setting up larger demand transition scenarios with many facilities.

### 3.1 Transition Scenario: Constant Demand

In this section, a constant power transition scenario is shown. Table 2 shows the simulation parameters used in this transition scenario. The input file used to generate this simulation can be found in constant_transition.xml and the file used to run the simulation and generate the plots can be found in algorithm_performance_tests_transitions.py.

Table 2: Constant Power Demand Transition Scenario Parameters

| **Commodity** | **Parameter** | **Description** |
|---|---|---|
| **Power** | Demand Equation | $D_p(t) = 10000MW$ |
| **Power** | Prediction Method<br>Supply Buffer | Fast Fourier Transform<br>3000 MW |
| **Fuel** | Prediction Method<br>Supply Buffer | Moving Average<br>0 kg |
| **Spent Fuel** | Prediction Method<br>Capacity Buffer | Moving Average<br>0 kg |

Figures 2a, 2b and 2c demonstrate `d3ploy`'s capability to deploy reactor and supporting facilities to meet the user determined power demand and subsequently demanded secondary commodities with minimal undersupply. Table 3 shows the number of undersupplied time steps.

In figure 2a, the supply of power never falls under demand. By using

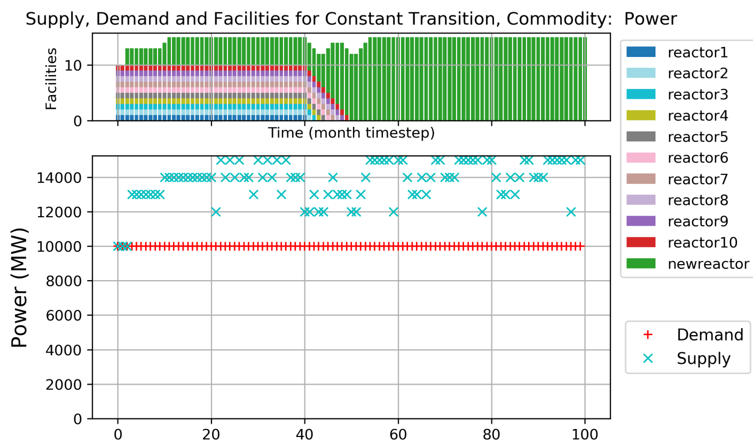
Table 3: Undersupply results for each commodity in each scenario

| Power Demand Equation | Commodity | Undersupplied Time Steps |
|---|---|---|
| **Constant** | Fuel | 1 |
| | Power | 0 |
| | Spent Fuel | 0 |
| **Linearly Increasing** | Fuel | 1 |
| | Power | 0 |
| | Spent Fuel | 0 |
| **Sinusoidal** | Fuel | 1 |
| | Power | 1 |
| | Spent Fuel | 0 |

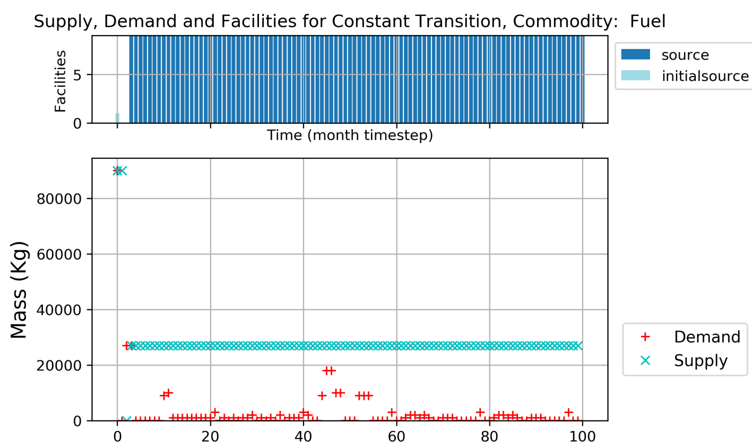
a combination of the fast fourier transform method for predicting demand and setting the supply buffer to 3000MW (the capacity of 3 reactors), the user minimizes the number of undersupplied time steps of every commodity. To avoid an undersupply, it is helpful to perform a sensitivity analysis of the size of buffer to use for each commodity.

In figure 2b, a facility with a large fuel throughput is initially deployed to meet the large initial fuel demand for the starting up of ten reactors. `D3ploy` is prevented from deploying many supporting facilities that end up being redundant at the later parts of the simulation, by having an initial facility with a large throughput exist for the first few time steps in the simulation. This is a reflection of reality in which reactor manufacturers will accumulate an appropriate amount of fuel inventory before starting up reactors. After the decommissioning of the large initial fuel production facility we notice an undersypply in one time step. This is unavoidable since the prediction methods in `d3ploy` are unable to predict this sudden drop in demand.

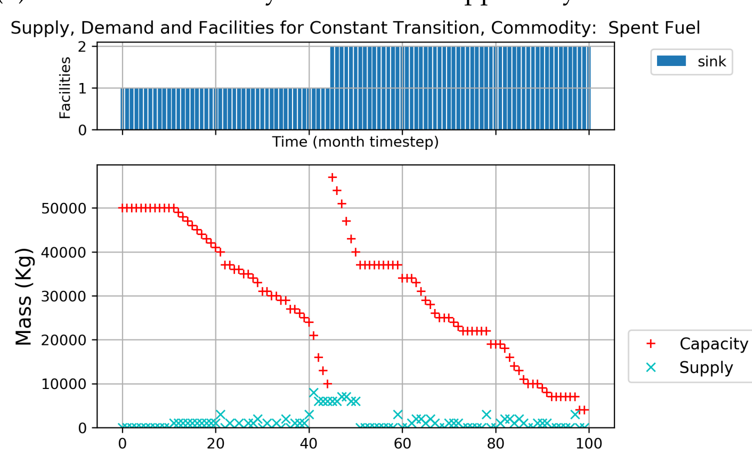
It is helpful to perform a small sensitivity analysis of the size of buffer used for each commodity to ensure that no undersupply occurs based on the nuances of any given facility type: refueling in a reactor, etc.. For these kind of simulations, those in which a facility requires a large initial amount of some commodity, the user should add an initial facility with a large production capacity that exists for only the first few time steps in the simulation; this prevents `d3ploy` from deploying a large number of supporting facilities that end up being redundant later in the simulation. Alternatively, this could be circumvented by introducing decommissioning capability into `d3ploy`.




(a) The power demand is a user-defined equation and power is supplied by the reactors.




(b) Fuel is demanded by reactors and supplied by source facilities.




(c) Spent Fuel is supplied by reactors and the capacity is provided by sink facilities.

Figure 2: Transition Scenario: Constant Power Demand of 10000MW

### 3.2 Transition Scenario: Linearly Increasing Demand

In this section, a transition scenario with a linearly increasing power demand is shown. Table 4 shows the simulation parameters used in this transition scenario.

Table 4: Linearly Increasing Power Demand Transition Scenario Parameters

| Commodity | Parameter | Description |
|---|---|---|
| **Power** | Demand Equation | $D_p(t < 40) = 10000$ MW<br>$D_p(t > 40) = 250t$ MW |
| **Power** | Prediction Method<br>Supply Buffer | Fast Fourier Transform<br>2000 MW |
| **Fuel** | Prediction Method<br>Supply Buffer | Moving Average<br>1000 kg |
| **Spent Fuel** | Prediction Method<br>Capacity Buffer | Fast Fourier Transform<br>0 kg |

Figures 3a, 3b and 3c demonstrate the capability of `d3ploy` to deploy reactor and supporting facilities to meet the power demand and subsequently demanded secondary commodities for a linearly increasing power demand.

The fast fourier transform method for predicting power demand is used for this scenario which is identical to what was used for the constant power demand transition scenario. A smaller power buffer of 2000MW could be used while still minimizing under supply.

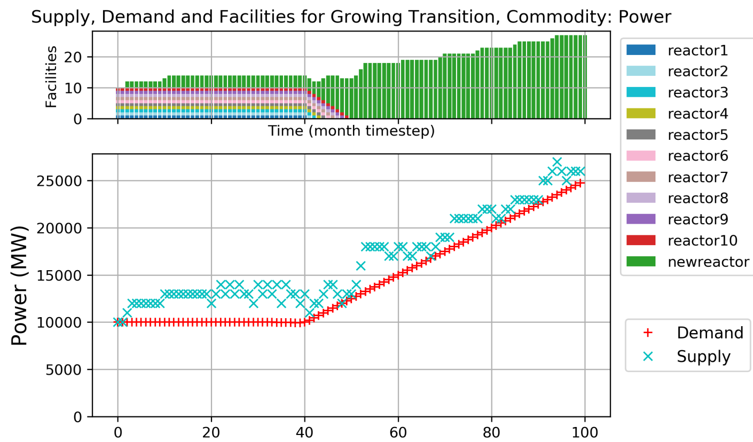
The input file used to generate this simulation can be found in growing_transition.xml and the file used to run the simulation and generate the plots can be found in algorithm_performance_tests_transitions.py.

### 3.3 Transition Scenario: Sinusoidal Demand

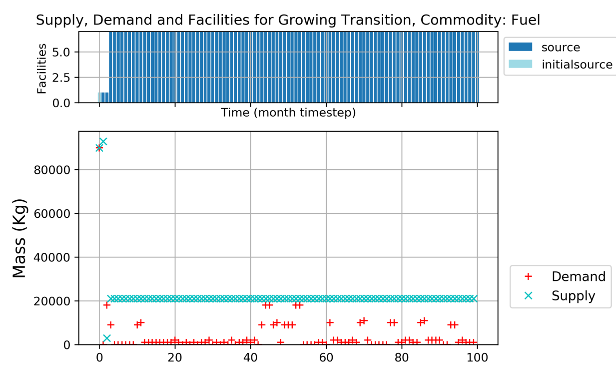
In this section, a transition scenario with sinusoidal power demand is shown. A sinusoidal power demand is the reflection of power demand in the real world such that power usage is higher in the winter and summer

and lower in the spring and fall. Table 5 shows the simulation parameters used in this transition scenario.

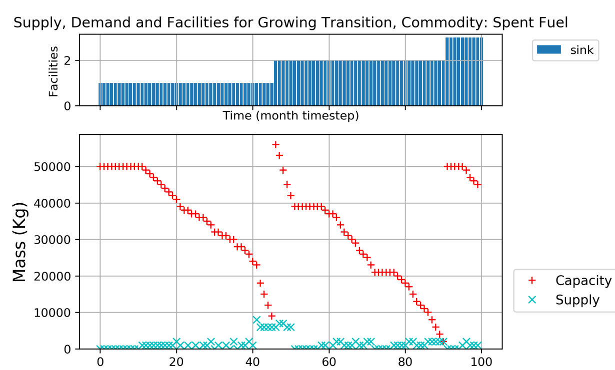
Figures 4a, 4b and 4c demonstrate the capability of `d3ploy` to deploy reactor and supporting facilities to meet the power demand and subsequently demanded secondary commodities for a sinusoidal power demand.




(a) The power demand is a user-defined equation and power is supplied by the reactors.




(b) Fuel is demanded by reactors and supplied by source facilities.




(c) Spent Fuel is supplied by reactors and the capacity is provided by sink facilities.

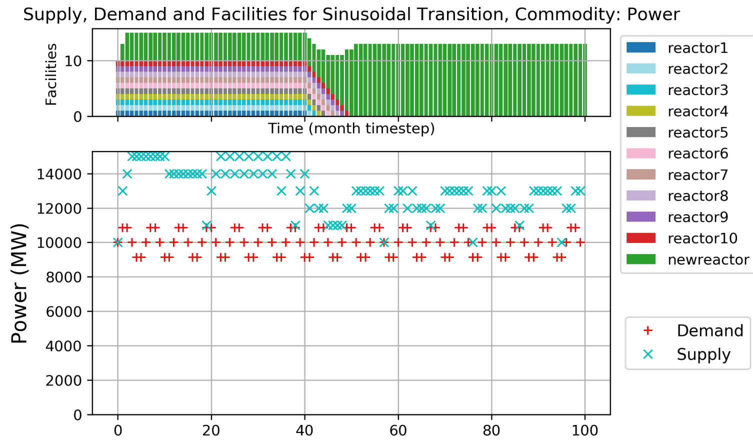
Figure 3: Transition Scenario: Linearly increasing power demand.

For a sinusoidal power demand, the use of the triple exponential method for predicting demand is more effective than the fast fourier transform method which was used for the constant and linearly increasing power demand transition scenarios. This is because the triple exponential smoothing method excels in forecasting data points for repetitive seasonal series of data.

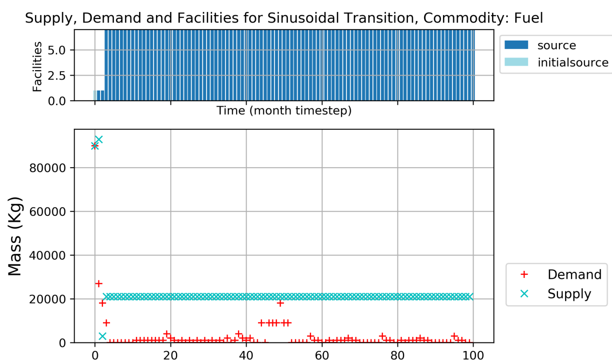
Table 5: Sinusoidal Power Demand Transition Scenario Parameters

| Commodity | Parameter | Description |
|---|---|---|
| **Power** | Demand Equation | $D_p(t) = 1000\sin\left(\frac{\pi t}{3}\right) + 10000$ |
| **Power** | Prediction Method | Triple Exponential Smoothing |
| | Supply Buffer | 2000 MW (2 reactor capacities) |
| **Fuel** | Prediction Method | Moving Average |
| | Supply Buffer | 1000 kg |
| **Spent Fuel** | Prediction Method | Fast Fourier Transform |
| | Capacity Buffer | 0 kg |

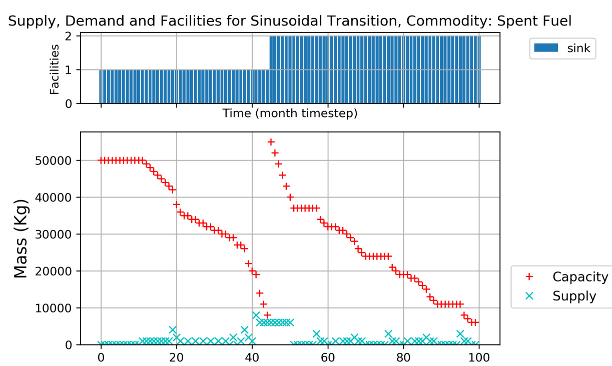
The input file used to generate this simulation can be found in: sine_transition.xml and the file used to run the simulation and generate the plots can be found in: algorithm_performance_tests_transitions.py.




(a) The power demand is a user-defined equation and power is supplied by the reactors.




(b) Fuel is demanded by reactors and supplied by source facilities.




(c) Spent Fuel is supplied by reactors and the capacity is provided by sink facilities.

Figure 4: Transition Scenario: Sinusoidal Power Demand

# 4 Transition Scenarios

The objective of this section was to carry out various simulations to validate `D3ploy`'s current capabilities for simulating complex cycles. The Idaho National Laboratory Nuclear Fuel Cycle Evaluation and Screening Report [5] established several fuel cycle scenarios. As part of the project NEUP-FY16-10512, the simulations focused on the cases EG01, EG23, EG24. The scenarios started at EG01 – representing the current U.S. fuel cycle – and transitioned to advanced fuel cycles. The simulations utilized `d3ploy`'s NO, DO, and SO algorithms.

All the analyzed scenarios started at EG01. In EG01 all reactors were LWRs running a once-through cycle burning enriched-U. In EG23 fast reactors (FRs) produced all the power, relying on the continuous recycle of U/Pu supplemented by the addition of new natural-U to the cycle. EG24 was similar to EG23, but its cycle utilized continuous recycling of U/TRU with the addition of new natural-U.

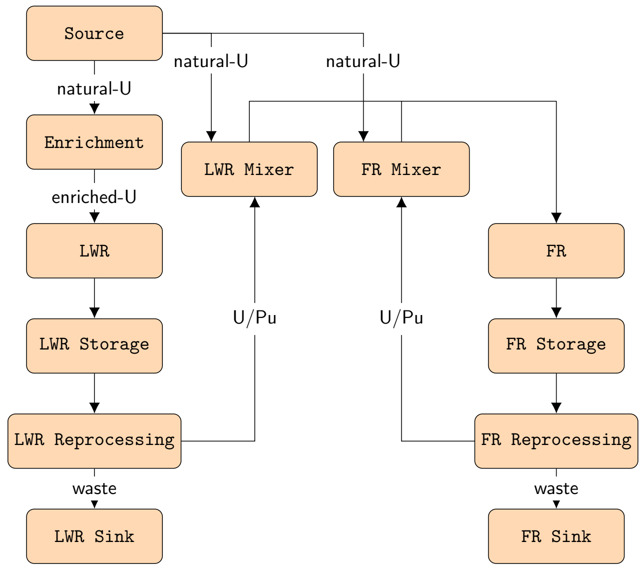
The present work focused on two transition scenarios: EG01-EG23 and EG01-EG24, as shown in Figure 5. The simulations started with a fleet of LWRs. After 80 years, the simulation progressively decommissioned the LWRs while transitioning to FRs. By the end of the cycle, all power was produced by FRs. Initial fueling of the FRs relied on reprocessed Pu from the LWR fleet. Following the transition, the FRs were able to produce their own Pu to sustain the cycle.

The following section presents the results for EG01-EG23 and EG01-EG24. The power demand was set at a constant 60 GW at all times. The transition scenarios used the capability of deploying facilities based on the difference between predicted demand and predicted supply, using a power supply buffer of 2000 MW.

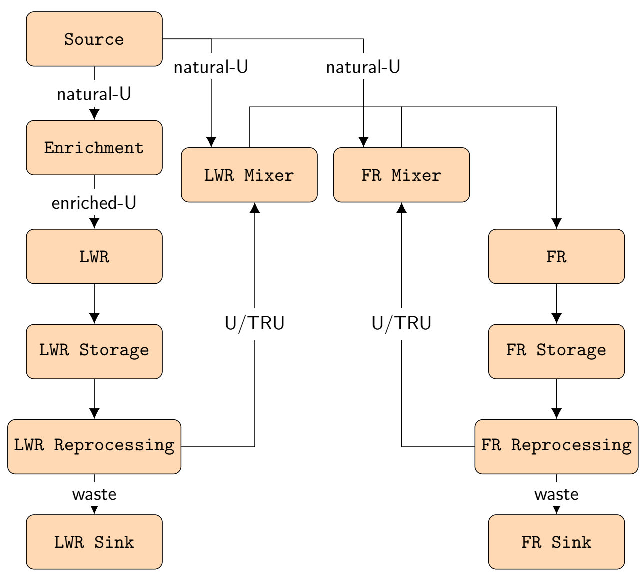
This section also includes a sensitivity analysis of the buffer size. A separate sensitivity analysis shows the dependency of the undersupply on the number of look-ahead time steps used to calculate the predicted demand and supply.

## 4.1 EG01-EG23

Figure 6 shows the power demand and supply obtained using different prediction methods. Following it, Table 6 displays a comparison of the different algorithms. Table 6 presents the Cumulative Undersupply and the Cumulative Oversupply magnitudes. These values represent the summation of the difference between the power supplied and the power demanded for




(a) EG01-EG23.




(b) EG01-EG24.

Figure 5: Diagrams with facilities and mass flow of the scenarios EG01-EG23 and EG01-EG24.

all the time steps in the simulation. This magnitude could best be thought of as energy. For undersupply conditions, the magnitude represents lack of energy provided during the time steps in which the supply did not meet the demand. Likewise, the oversupply would be the magnitude of excess energy produced.

Table 6: Undersupply and oversupply of Power for the different algorithms used to calculate EG01-EG23.

| Algorithm | Undersupplied Timesteps | Cumulative Undersupply [GW] | Cumulative Oversupply [GW] |
|---|---|---|---|
| MA | 20 | 20.0 | 920.5 |
| ARMA | 18 | 7.7 | 1036.5 |
| ARCH | 0 | 0 | 1320.1 |
| POLY | 1 | 0.3 | 1783.5 |
| EXP_SMOOTHING | 20 | 11.0 | 1473.5 |
| HOLT-WINTERS | 20 | 11.0 | 1473.5 |
| FFT | 2 | 60.3 | 1751.9 |
| SW_SEASONAL | 20 | 18.6 | 1119.9 |

Table 7: Number of time steps with undersupply and under capacity of various commodities for the different algorithms used to calculate EG01-EG23.

| | Undersupply | | | Undercapacity | |
|---|---|---|---|---|---|
| Algorithm | Natural U | Enriched U | FR fuel | LWR PU | FR PU |
| MA | 0 | 0 | 0 | 1 | 1 |
| ARMA | 0 | 0 | 0 | 1 | 1 |
| ARCH | 0 | 0 | 0 | 1 | 1 |
| POLY | 0 | 0 | 0 | 1 | 1 |
| EXP_SMOOTHING | 0 | 0 | 0 | 1 | 1 |
| HOLT_WINTERS | 0 | 0 | 0 | 1 | 1 |
| FFT | 0 | 1 | 0 | 1 | 1 |
| SW_SEASONAL | 0 | 0 | 0 | 1 | 1 |

Table 7 presents the number of time steps with undersupply of natural-U (sourceout), enriched-U (enrichmentout), and FR fuel. The table also displays the number of time steps in which the capacity of the LWR Mixer to process LWR Pu and the capacity of the FR Mixer to process FR Pu are not enough (undercapacity). In this table we notice that the supply of Pu and the deployment of the respective mixer has one time step of delay.

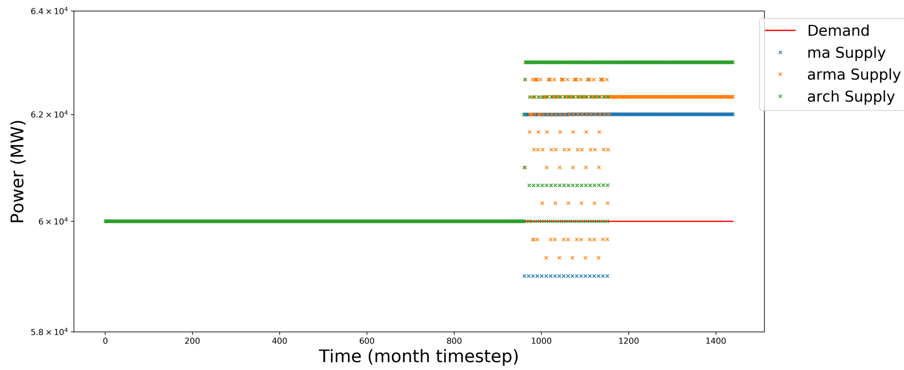
One of the methods that performs the better is ARCH. For this scenario and said method, Figure 7 presents some of the different supply and demand time series plots for various commodities. Figure 7a presents the number of Source facilities deployed, and the resultant demand and supply of natural-U. For this case, the capacity of natural-U supply is higher than the demand. We notice that the demand in the beginning of the simulation is higher than in the end. The LWRs use enriched-U produced by the enrichment of natural-U, while the FRs require a smaller quantity of U for their fuel. Figure 7b displays the number of LWR Mixers deployed, and the supply and the capacity of LWR Pu (Pu produced by the LWRs). Logically, the supply of Pu decreases as the LWRs stop operating. Figure 7c shows the FR Mixers, and the supply and capacity of FR Pu. The supply of Pu increases as `d3ploy` deploys new FRs.

### 4.2 EG01-EG24

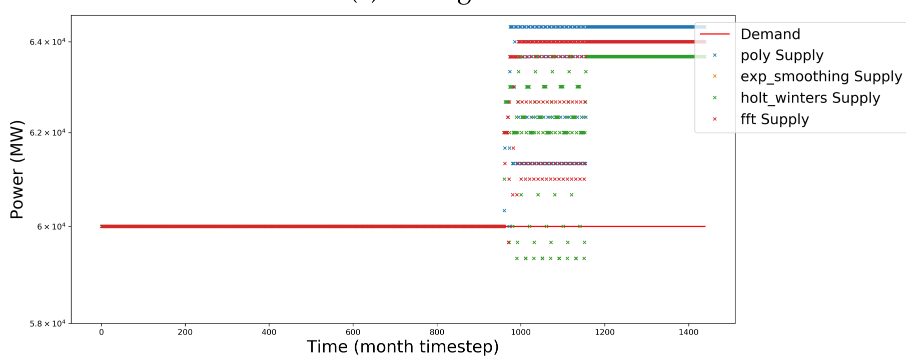
Figure 8 shows the power demand and supply obtained using different prediction methods. Following it, Tables 8 and 9 display a comparison of the different algorithms.

Table 8: Undersupply and oversupply of power with the different algorithms used to drive EG01-EG24.

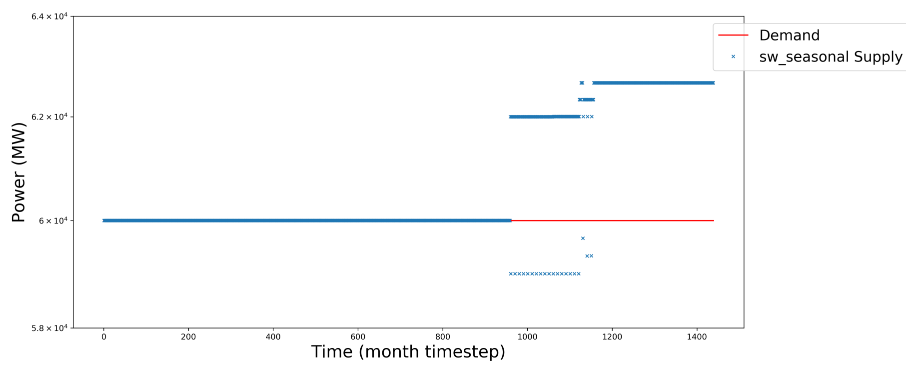
| | Power | | |
|---|---|---|---|
| Algorithm | Undersupplied Time Steps | Cumulative Undersupply [GW] | Cumulative Oversupply [GW] |
| MA | 20.0 | 20.0 | 920.5 |
| ARMA | 18.0 | 7.7 | 1036.5 |
| ARCH | 0 | 0 | 1320.1 |
| POLY | 1.0 | 0.3 | 1783.5 |
| EXP_SMOOTHING | 20.0 | 11.0 | 1473.5 |
| HOLT-WINTERS | 20.0 | 11.0 | 1473.5 |
| FFT | 2.0 | 60.3 | 1751.9 |
| SW_SEASONAL | 20.0 | 18.6 | 1119.9 |




(a) NO algorithms.

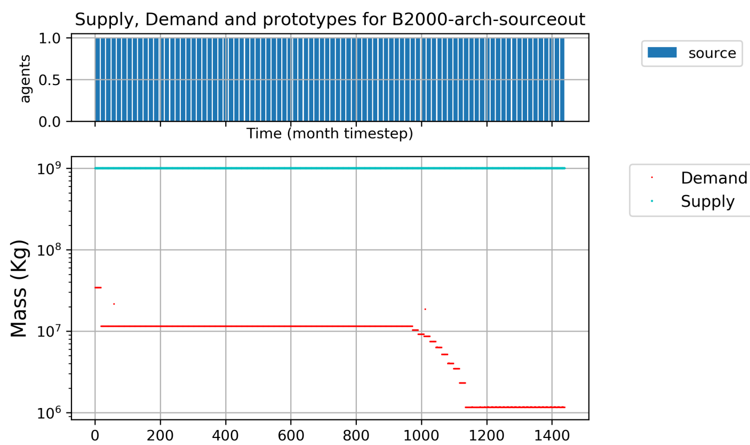



(b) DO algorithms.

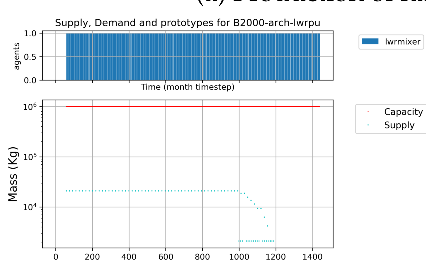



(c) SO algorithms.

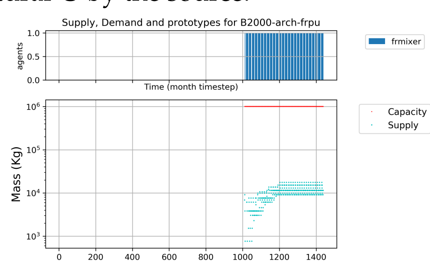
Figure 6: Plot of the power demand and supply of EG01-EG23 for a constant power demand of 60GW for different prediction algorithms.




(a) Production of natural-U by the source.

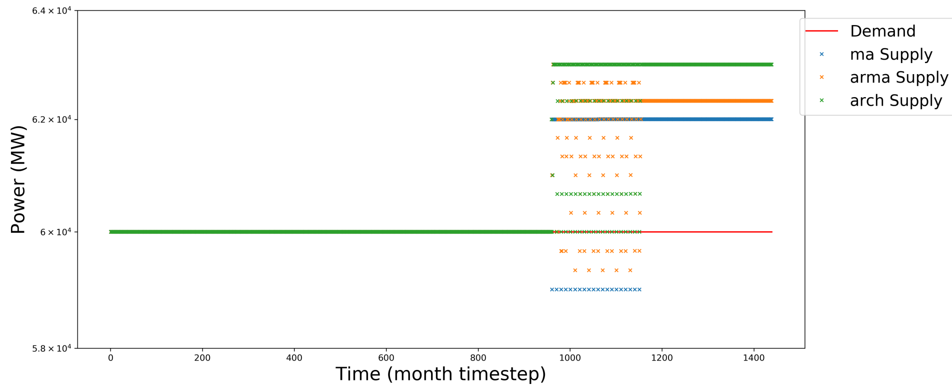



(b) Pu produced by the LWRs and exchanged to the LWR Mixer.

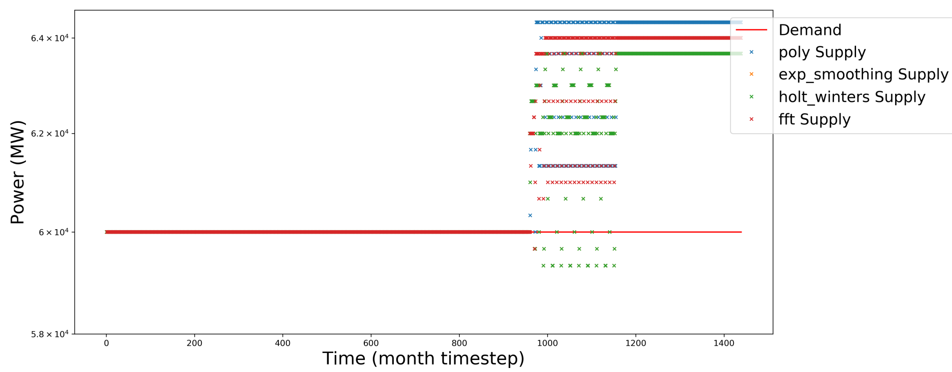



(c) Pu produced by the FRs and exchanged to the FR Mixer.

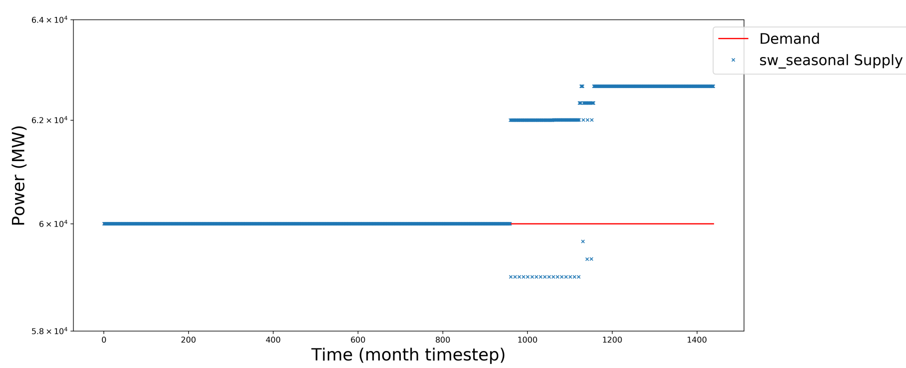
Figure 7: Plot for different commodities EG01-EG23.




(a) NO algorithms.




(b) DO algorithms.




(c) SO algorithms.

Figure 8: Plot of the power demand and supply of EG01-EG24 for a constant power demand of 60GW for different prediction algorithms.

Table 9: Number of time steps with undersupply and under capacity of various commodities for the different algorithms used to calculate EG01-EG24.

| | Undersupply | | | Undercapacity | |
|---|---|---|---|---|---|
| Algorithm | Natural U | Enriched U | FR fuel | LWR PU | FR PU |
| MA | 0 | 0 | 0 | 1 | 1 |
| ARMA | 0 | 0 | 0 | 1 | 1 |
| ARCH | 0 | 0 | 0 | 1 | 1 |
| POLY | 0 | 0 | 0 | 1 | 1 |
| EXP_SMOOTHING | 0 | 0 | 0 | 1 | 1 |
| HOLT_WINTERS | 0 | 0 | 0 | 1 | 1 |
| FFT | 0 | 1 | 0 | 1 | 1 |
| SW_SEASONAL | 0 | 0 | 0 | 1 | 1 |

## 4.3 Buffer Size

This section focuses on the analysis of undersupply dependency on buffer size in the EG01-EG23 transition scenario. Table 10 shows the number of time steps with undersupply and the cumulative undersupply for different buffer sizes and various prediction methods. Figure 9 displays the cumulative undersupply as a function of buffer size.

Table 10: Dependency of the undersupply of Power on the buffer size.

| Buffer $[MW]$ | | MA | ARMA | POLY | EXP SMOOTHING | FFT |
|---|---|---|---|---|---|---|
| 0 | Undersupplied [#] | 20 | 60 | 75 | 30 | 28 |
| | Cumulative $[GW]$ | 60.0 | 87.3 | 52.9 | 68.3 | 93.3 |
| 2000 | Undersupplied [#] | 20 | 18 | 1 | 20 | 2 |
| | Cumulative $[GW]$ | 20.0 | 7.7 | 0.3 | 11.0 | 60.3 |
| 4000 | Undersupplied [#] | 0 | 0 | 0 | 0 | 1 |
| | Cumulative $[GW]$ | 0 | 0 | 0 | 0 | 60. |

## 4.4 Number of Forward Steps

This section focuses on the dependency on the number of forward steps calculated at each time step by the prediction methods in scenario EG01-

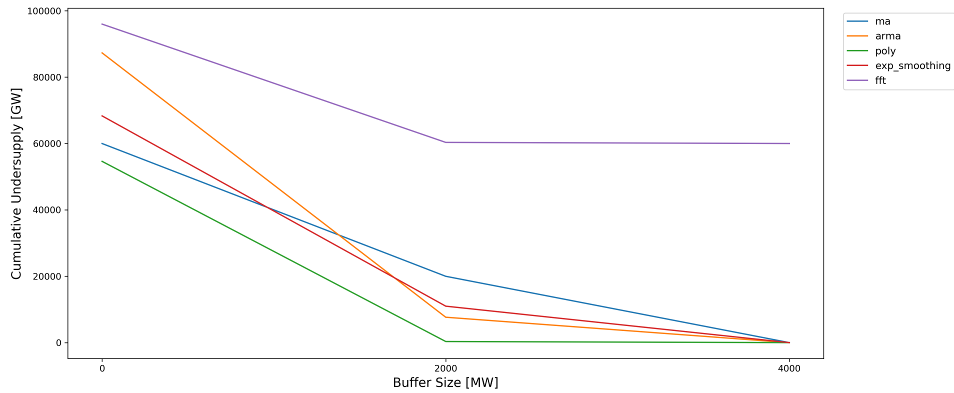



Figure 9: Plot of the dependency of the undersupply of Power on the buffer size.

EG23; the buffer size was fixed at 2000 MW. Table 11 shows number of time steps containing undersupply and the cumulative undersupply for different forward steps for some of the prediction methods. Figure 10 displays the cumulative undersupply as a function of the number of forward steps.

Table 11: Dependency of the undersupply of Power on the number of forward steps.

| Forward steps | | MA | ARMA | POLY | EXP SMOOTHING | FFT |
|---|---|---|---|---|---|---|
| 1 | # Undersupplied | 18 | 20 | 2 | 20 | 1 |
| | Cumulative $[GW]$ | 7.6 | 11.0 | 60.3 | 20.0 | 0.3 |
| 3 | # Undersupplied | 1 | 20 | 2 | 0 | 1 |
| | Cumulative $[GW]$ | 0.3 | 11.0 | 60.3 | 0 | 0.3 |
| 5 | # Undersupplied | 4 | 20 | 20 | 0 | 1 |
| | Cumulative $[GW]$ | 1.3 | 11.0 | 60.3 | 0 | 0.3 |

## 5 Conclusion and Next Steps

This paper describes the capabilities of `d3ploy` and demonstrates the use of `d3ploy` for simple transition scenarios with constant, linearly increasing, and sinusoidal power demand. The demonstration goes further with the more complex transition scenarios EG01-EG23 and EG01-EG24. This pa-

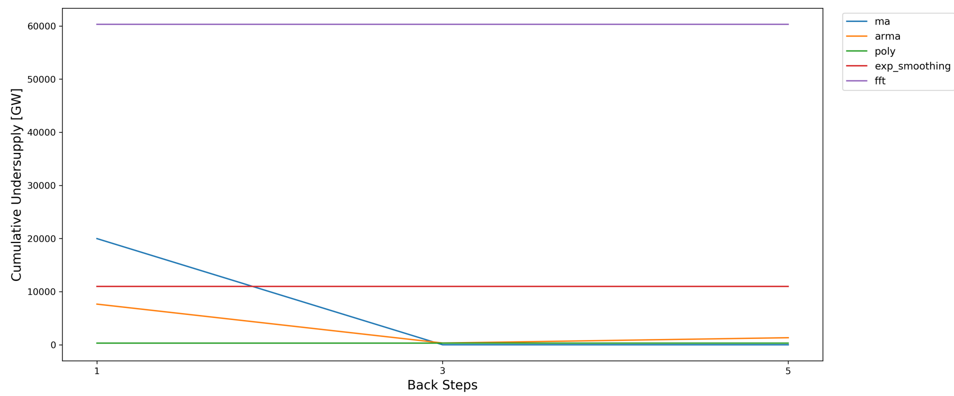



Figure 10: Plot of the dependency of the undersupply of Power on the number of forward steps.

per also provides insights on parameter inputs to ease the setup of larger transition scenarios that may include numerous facilities.

Future work includes setup of similar power demand transition scenarios for extended nuclear fuel cycles incorporating multiple reactor designs that consequently use different types of fuel. Such cases are currently under study. [5] established the transition scenarios EG01-EG29 and EG01-EG30. These scenarios are more complex than the cases presented in this report and the distribution of fuel between different reactor technologies play a main role in the transition. Additionally, as seen during the demonstration of `d3ploy` capabilities, a Decommissioning capability is highly useful for the setup of several NFCs and is currently under development.

# 6 Acknowledgements


This research is being performed using funding received from the Department of Energy (DOE) Office of Nuclear Energy's Nuclear Energy University Program (Project 16-10512, DE-NE0008567) 'Demand-Driven Cycamore Archetypes'.

The authors would like to thank members of the Advanced Reactors and Fuel Cycles (ARFC) group at the University of Illinois at Urbana-Champaign. We also thank our colleagues from the CYCLUS community, particularly those in the University of Wisconsin Computational Nuclear Engineering Research Group (CNERG) and the University of South Carolina Energy Research Group (ERGS) for collaborative CYCLUS development.

## References

[1] Robert W. Carlsen, Matthew Gidden, Kathryn Huff, Arrielle C. Opotowsky, Olzhas Rakhimov, Anthony M. Scopatz, and Paul Wilson. Cycamore v1.0.0. *Figshare*, June 2014. http://figshare.com/articles/Cycamore_v1_0_0/1041829.

[2] Gwendolyn Chee, Jin Whan Bae, Robert Flanagan, Roberto Fairhurst, and Kathryn Huff. arfc/d3ploy: Demonstration of demand driven deployment capabilities in cyclus, May 2019.

[3] Kathryn D Huff, Jin Whan Bae, Robert R Flanagan, and Anthony M Scopatz. Current Status of Predictive Transition Capability in Fuel Cycle Simulation. page 11, 2017.

[4] Kathryn D. Huff, Matthew J. Gidden, Robert W. Carlsen, Robert R. Flanagan, Meghan B. McGarry, Arrielle C. Opotowsky, Erich A. Schneider, Anthony M. Scopatz, and Paul P. H. Wilson. Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework. *Advances in Engineering Software*, 94:46–59, April 2016. arXiv: 1509.03604.

[5] R Wigeland, T Taiwo, H Ludewig, M Todosow, W Halsey, J Gehin, R Jubin, J Buelt, S Stockinger, and K Jenni. Nuclear Fuel Cycle Evaluation and Screening – Final Report. *Final Report*, page 10, 2014.