DDCA Summer 2019 Report

Gwendolyn J. Chee, Roberto E. Fairhurst, Kathryn D. Huff

July 8, 2019

1 Introduction

Nuclear Fuel Cycle (NFC) simulation scenarios are constrained objective functions. The objectives are systemic demands such as "1% power growth", while an example of a constraint is the availability of new nuclear technology. To add ease in setting up nuclear fuel cycle simulations, NFC simulators should bring demand responsive deployment decisions into the dynamics of the simulation logic [1]. While automated power production deployment is common in most fuel cycle simulators, automated deployment of supportive fuel cycle facilities is non-existent.

Instead, the user must detail the deployment timeline of all supporting facilities or have infinite capacity support facilities. Thus, a next generation NFC simulator should predictively and automatically deploy fuel cycle facilities to meet user defined power demand.

CYCLUS is an agent-based nuclear fuel cycle simulation framework [2]. Each entity (i.e. Region, Institution, or Facility) in the fuel cycle is modeled as an agent. Institution agents are responsible for deploying and decommissioning facility agents and can represent a legal operating organization such as a utility, government, etc [2].

The Demand-Driven CYCAMORE Archetypes project (NEUP-FY16-10512) aims to develop CYCLUS's demand-driven deployment capabilities. This capability is developed in the form of 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. Its goal is to meet supply for any commodity while minimizing undersupply. This demand-driven deployment capability is referred to as `d3ploy`.

In this paper, we will explain the capabilities of `d3ploy`, demonstrate how `d3ploy` is used to meet the primary objective of minimizing undersupply of all commodities in a simulation. The goal is to study a basic transition scenarios with constant, linearly increasing and sinusoidal power demand. Such study provides recommendations and insights to inform decisions about parameter inputs when setting up larger transition scenarios that include many facilities. The last cases analyzed are such transition scenarios.

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 to minimize the number of time steps of undersupply of any commodity.

When there is a predicted undersupply of a commodity, `d3ploy` looks at what facilities it has that provides that commodity and will deploy the fewest number of facilities to meet the predicted demand. This logic is available in `solver.py`.

2.2 Basic User-Defined Input Variables

The user is able to input 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 for the institution to control and their corresponding capacities. The user must also define the driving commodity, its demand equation and what calculation method the institution predicts demand and supply with.

They also have the option to give a time dependent equation that governs preference for that facility compared to other facilities that provide the same commodity. The user also has an option to constrain deployment of a facility until there is a accumulation of the inventory of a specific commodity. The user can also define an initial list of facilities that are 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. For example, the triple exponential smoothing method is most effective for predicting demand and supply for the power commodity in a scenario with a sinusoidal power demand compared to a linearly increasing power demand. Whereas, the fast fourier transform method is more effective than triple exponential smoothing for the non-power commodities in the same scenario.

2.4 Difference between Demand and Supply Driven Institutions

Within `d3ploy`, there are two institutions: `DemandDrivenDeploymentInst` and `SupplyDrivenDeploymentInst`. The prior is used for the front-end of the fuel cycle and the latter is used for the back-end. Front-end facilities are those that exist before the reactor in a nuclear fuel cycle such as a fuel fabrication facility. Back-end facilities go after the reactor in a nuclear fuel cycle such as a reprocessing facility. The reason for this separation is to let facilities have the choice to demand for supply or demand for capacity. For example, in the front-end facilities, the reactor has a demand for fuel that triggers the deployment of fuel fabrication facilities. Such facilities will create a supply to meet the demand. Whereas, for the back end facilities, the reactor generates spent fuel, there is a demand for a waste repository facility to accept the spent fuel. This triggers the deployment of a waste repository that will create a capacity to receive the available 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. There are two reasons for wanting to use installed capacity over predicted supply. The first is for facilities that provide intermittent supply, such as a reactor facility that has a designated refueling time. During time steps where 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 where the input commodity for a facility has run out in a simulation 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 to fast reactors that require plutonium from Light Water Reactor (LWR)'s spent nuclear fuel (SNF), if the fast reactor's demand for plutonium exceeds the inventory provided by LWRs before they were decommissioned, it will result in the deployment of mixer facilities that generate the fast reactor fuel despite the lack of plutonium to generate the fuel. This is an example of a poorly set up transition scenario.

2.6 Supply/Capacity Buffer

In `DemandDrivenDeploymentInst`, the user can choose to provide a buffer for the predicted supply so that `d3ploy` will ensure that the predicted supply will meet the predicted demand with the additional buffer.

In `SupplyDrivenDeploymentInst`, the user can choose to provide a buffer for the predicted capacity so that `d3ploy` will ensure predicted capacity meets the predicted supply with the additional buffer. The buffer can be defined as a percentage value or an absolute value.

3 Demonstration of d3ploy capabilities

To demonstrate `d3ploy`'s capabilities we run simulations with constant, linearly increasing, and sinusoidal power demand. A balance between the various system parameters must be met for each type of simulation to minimize the undersupply and under capacity for the various commodities.

These simulations were basic transition scenarios that only included three types of facilities: `source`, `reactor` and `sink`. All of the simulations began with ten reactor facilities, `reactor1` to `reactor10`. These reactors had staggered cycle lengths and lifetimes so that they did not refuel and decommission all at the same time step. `D3ploy` deploys reactor facilities of `new reactor` type to meet the power demand that occurred when the ten initial reactor facilities began to decommission.

All the simulations deployed facilities based on the relationship between predicted demand and installed capacity, capability discussed in the previous section. Table 1 shows the simulation parameters that are consistent across all the discussed scenarios. Table 2 displays the number of time steps where there was an undersupply for each commodity.

The reason for setting up these basic transition scenarios is to demonstrate `d3ploy`'s capabilities for use in simulating transition scenarios and also to inform decisions about parameter inputs when setting up larger demand transition scenarios that include many facilities.

Table 1: Transition Scenario Parameters for the constant, linear increasing, and sinusoidal power demand simulations.

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

Table 2: Undersupply results for each commodity in each scenario.

Transition Scenario	**Commodity**	**No. of time steps with undersupply**
Constant Power	Fuel	1
	Power	0
	Spent Fuel	0
Linearly Increasing Power	Fuel	1
	Power	0
	Spent Fuel	0
Sinusoidal Power	Fuel	1
	Power	1
	Spent Fuel	0

3.1 Transition Scenario: Constant Demand

This section shows a constant power transition scenario. Table 3 displays the simulation parameters. The input file used to generate this simulation can be found in:
/d3ploy/input/constant_transition.xml
and the file used to run the simulation and generate the plots can be found in:
/d3ploy/tests/performance_tests/algorithm_performance_tests_transitions.py

Figures 1a, 1b and 1c demonstrate the capability of `d3ploy` to deploy reactor and supporting facilities to meet the user determined power demand and subsequently demanded secondary commodities with minimal time steps where there is an undersupply.

Table 2 shows the number of time steps where there was an undersupply

Table 3: Constant Power Demand Transition Scenario's Parameters.

	Parameters	**Description**
Overall	Demand Equation	10000 MW
Power Commodity	Prediction Method	Fast Fourier Transform
	Supply Buffer	3000 MW
Fuel Commodity	Prediction Method	Moving Average
	Supply Buffer	0 kg
Spent Fuel Commodity	Prediction Method	Moving Average
	Capacity Buffer	0 kg

for each commodity in this scenario. In figure 1a, there are no time steps where the supply of power falls under demand.

The use of the fast fourier transform method for predicting the demand and setting the supply buffer to 3000MW (the capacity of 3 reactors) minimized the number of undersupply time steps.

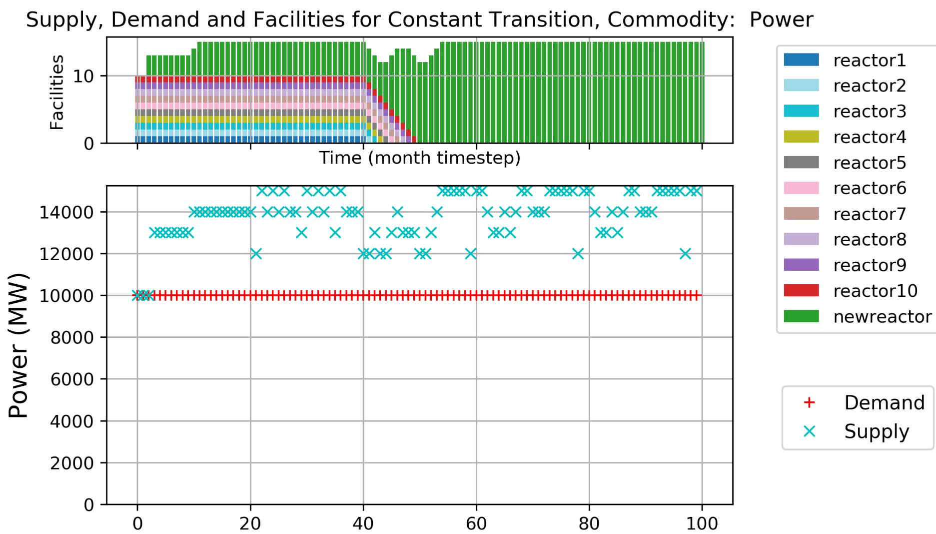
It is important to perform a small sensitivity analysis of the size of buffer to use for each commodity to ensure that there is no undersupply based on the nuances of the facility type: refueling in a reactor etc.

In figure 1b, a facility with a large throughput of fuel is initially deployed to meet the large initial fuel demand for the starting up of ten reactors. This is a reflection of reality where reactor manufacturers will accumulate an appropriate amount of fuel inventory before starting up reactors. There is one time step where there is an undersupply after the decommissioning of the large initial facility. This is unavoidable as the prediction methods in `d3ploy` are unable to predict this sudden drop in demand.

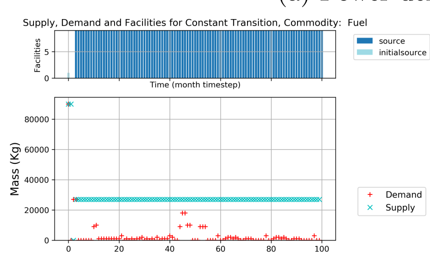
For simulations where a facility requires a large amount of start up commodity such as in this simulation, the user should add an initial facility with a large throughput to exist for the first few time steps in the simulation, so as to prevent `d3ploy` from deploying a large amount of supporting facilities that end up being redundant at the later parts of the simulation. This could also be circumvented by introducing decommissioning capability into `d3ploy`.

3.2 Transition Scenario: Linearly Increasing Demand

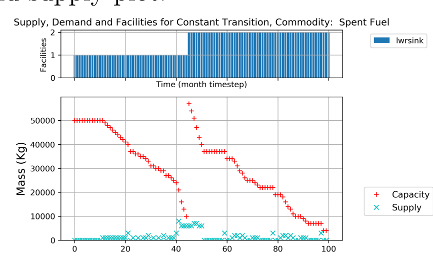
This section presents a transition scenario where there is a linearly increasing power demand. Table 4 displays the simulation parameters used in this transition scenario.



(a) Power demand and supply plot.



(b) Fuel demand and supply plot.



(c) Spent Fuel demand and supply plot.

Figure 1: Transition Scenario: Constant Power Demand of 10000MW.

Table 4: Linearly Increasing Power Demand Transition Scenario's Parameters.

	Parameters	**Description**
Overall	Demand Equation	Time<40: 10000 MW, Time>40: 250*t MW
Power Commodity	Prediction Method	Fast Fourier Transform
	Supply Buffer	2000 MW
Fuel Commodity	Prediction Method	Moving Average
	Supply Buffer	1000 kg
Spent Fuel Commodity	Prediction Method	Fast Fourier Transform
	Capacity Buffer	0 kg

Figures 2a, 2b and 2c demonstrate the capability of `d3ploy` to deploy reactor and supporting facilities to meet the user determined power demand and subsequently demanded secondary commodities for a linearly increasing power demand.

This scenario made use of the fast fourier transform method for predicting power demand, similar to what the constant power demand transition scenario used. The supply buffer for power was of 2000MW.

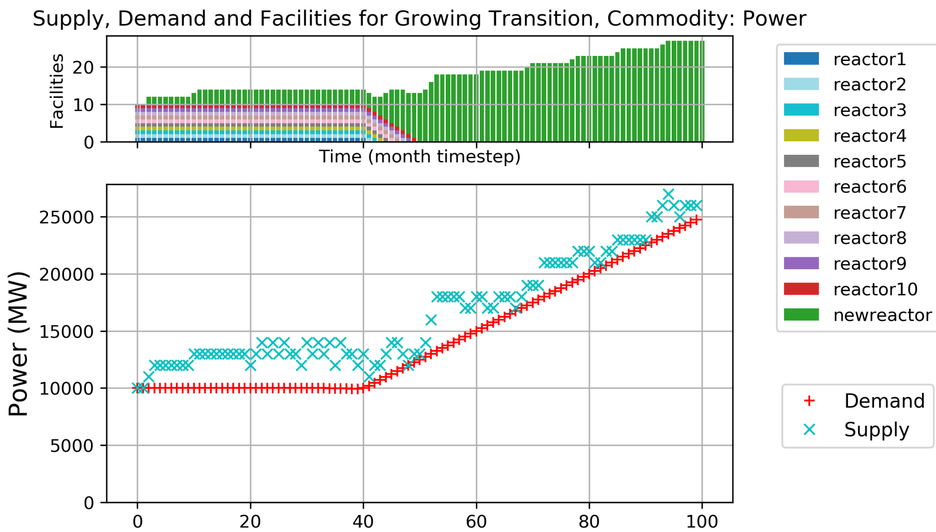
The input file used to generate this simulation can be found in:
/d3ploy/input/growing_transition.xml
and the file used to run the simulation and generate the plots can be found in:
/d3ploy/tests/performance_tests/algorithm_performance_tests_transitions.py

3.3 Transition Scenario: Sinusoidal Demand

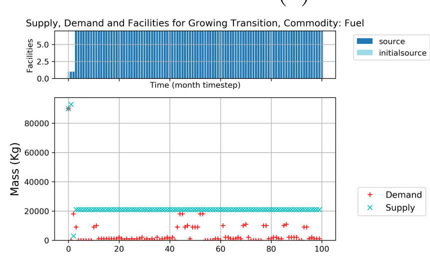
This section shows a transition scenario with sinusoidal power demand. A sinusoidal power demand is the reflection of power demand in the real world where power usage is higher in the winter and summer and it is smaller in the spring and fall. Table 5 displays the simulation parameters used in this transition scenario. The power demand had an amplitude of 1000MW.

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

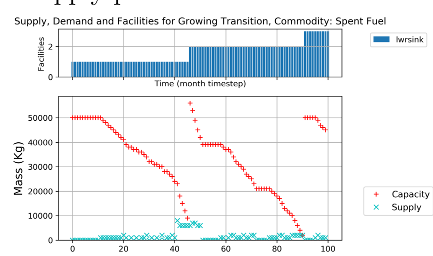
For a sinusoidal power demand, the use of the triple exponential method (holt winters) for predicting demand is more effective than the fast fourier transform method which was used for the constant and linearly increasing



(a) Power demand and supply plot.



(b) Fuel demand and supply plot.



(c) Spent Fuel demand and supply plot.

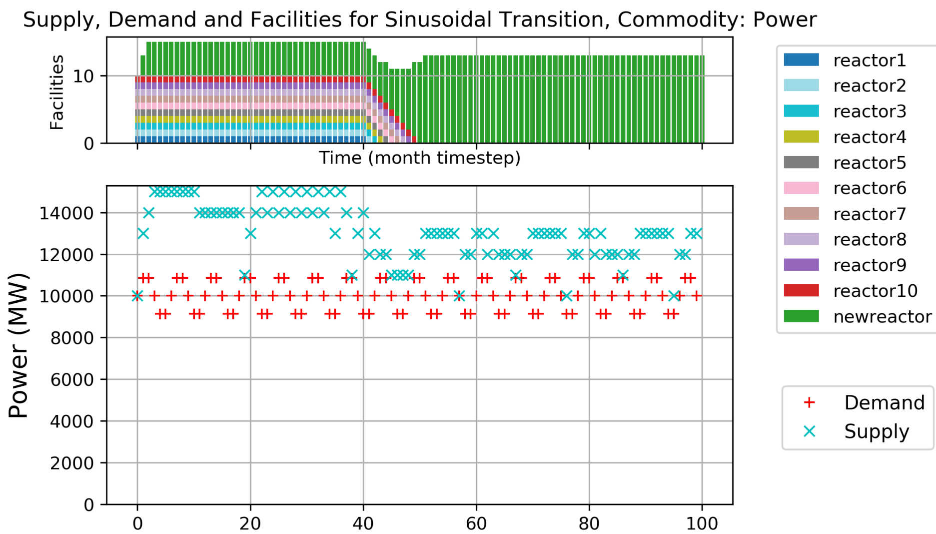
Figure 2: Transition Scenario: Linearly Increasing Power Demand.

power demand transition scenarios. This is because the triple exponential smoothing method excels in forecasting data points for repetitive seasonal series of data.

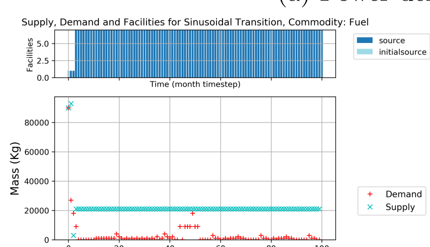
Table 5: Sinusoidal Power Demand Transition Scenario's Parameters.

	Parameters	**Description**
Overall	Demand Equation	$1000sin(\frac{\pi * t}{3}) + 10000$
Power Commodity	Prediction Method	Triple Exponential Smoothing
	Supply Buffer	2000 MW
Fuel Commodity	Prediction Method	Moving Average
	Supply Buffer	1000 kg
Spent Fuel Commodity	Prediction Method	Fast Fourier Transform
	Capacity Buffer	0 kg

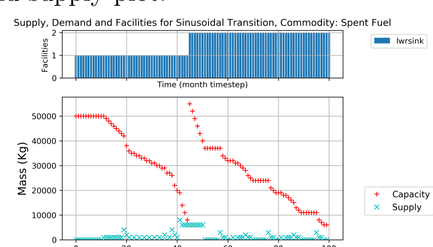
The input file used to generate this simulation can be found in:
/d3ploy/input/sine_transition.xml
and the file used to run the simulation and generate the plots can be found in:
/d3ploy/tests/performance_tests/algorithm_performance_tests_transitions.py



(a) Power demand and supply plot.



(b) Fuel demand and supply plot.



(c) Spent Fuel demand and supply plot.

Figure 3: Transition Scenario: Sinusoidal Power Demand.

4 Transition Scenarios

The objective of this section was to carry out various simulations to prove `D3ploy`'s current capabilities for simulating complex cycles. The Idaho National Laboratory Nuclear Fuel Cycle Evaluation and Screening Report [3] 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 the reactors were LWR's. The cycle is once-through and feeds the reactors with enriched-U. In EG23 Fast reactors (FR's) produced all the power. The reactors relied on the continuous recycle of U/Pu with 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.

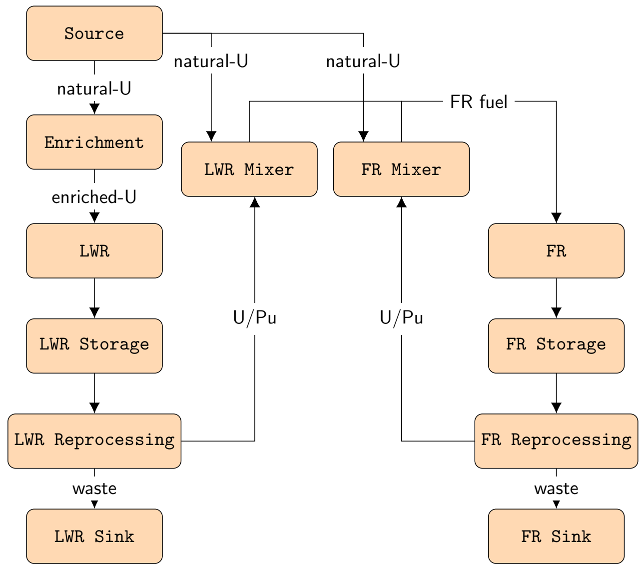
The present work focused on two transition scenarios: EG01-EG23 and EG01-EG24, shown in Figure 4. The simulations started with a fleet of LWR's. After 80 years, the simulation decommissioned the LWR's progressively transitioning to FR's. Finally, the FR's produced all the power. The start up of the FR's relies on the reprocessed Pu from the LWR's. After the transition, the FR's 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 considered was a constant of 60 GW at all times. The transition scenarios used the capability of deploying facilities based on the difference between predicted demand and predicted supply. The supply buffer of power was 2000 MW.

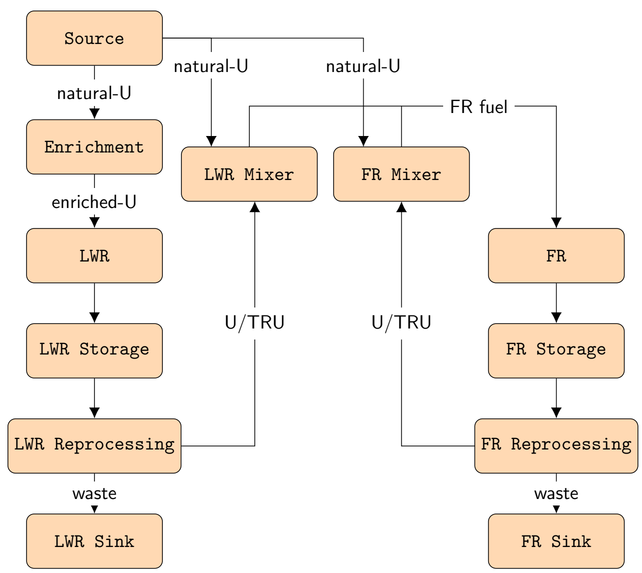
This section counts also with a sensitivity analysis of the buffer size. Another sensitivity analysis shows the dependency of the undersupply on the number of previous time steps used to calculate the predicted demand and supply.

4.1 EG01-EG23

Figure 5 shows the power demand and supply obtained using different prediction methods. Following, the tables 6 and 7 display a comparison of the different algorithms. The table 6 has 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 all the time steps in the simulation. This magnitude could be thought as



(a) EG01-EG23.



(b) EG01-EG24.

Figure 4: Diagrams with facilities and mass flow of the scenarios EG01-EG23 and EG01-EG24.

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

	Power		
Algorithm	No. of time steps of undersupply	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

Energy. For the undersupply, it would be the lack of Energy provided during the time steps where the supply did not meet the demand. On the contrary, the oversupply would be the extra energy produced.

One of the methods that performs the better is ARCH. For this scenario and that method, figure 6 present some of the different supply and demand time series plots for different commodities.

4.2 EG01-EG24

Figure 7 shows the power demand and supply obtained using different prediction methods. Following, the tables 8 and 9 display a comparison of the different algorithms.

4.3 Buffer Size

This section focuses on EG01-EG23 for analyzing the dependency of the undersupply on the buffer size. Table 10 shows number of time steps of undersupply and the cumulative undersupply for different buffer sizes for some of the prediction methods. Figure 8 displays the cumulative undersupply vs the buffer sizes.

4.4 Number of Back Steps

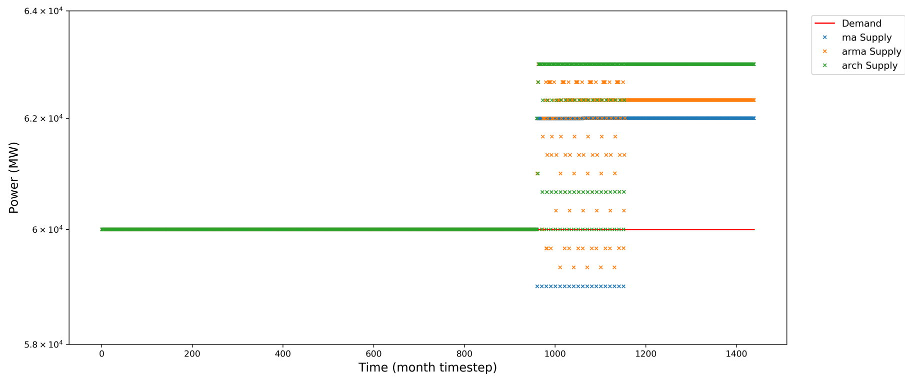
Still for the case EG01-EG23, this section focuses on the dependency on the number of back steps used by the prediction methods to predict de-

Table 7: No. of time steps with undersupply and under capacity of various commodities for the different algorithms used to calculate EG01-EG23.

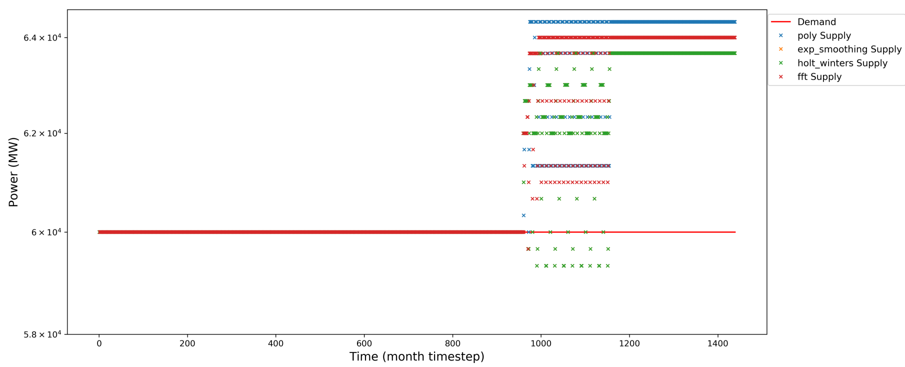
	Undersupply			Undercapcity	
Algorithm	Sourceout	Enrichmentout	FR fuel	LWR PU	FR PU
MA	0	0	0	1	1
ARMA	0	0	0	1	1
ARCH	0	0	0	1	1
POLY	0	0	0	1	1
EXP_SMOOTHING	0	0	0	1	1
HOLT_WINTERS	0	0	0	1	1
FFT	0	1	0	1	1
SW_SEASONAL	0	0	0	1	1

Table 8: Undersupply and oversupply of Power for the different algorithms used to calculate EG01-EG24.

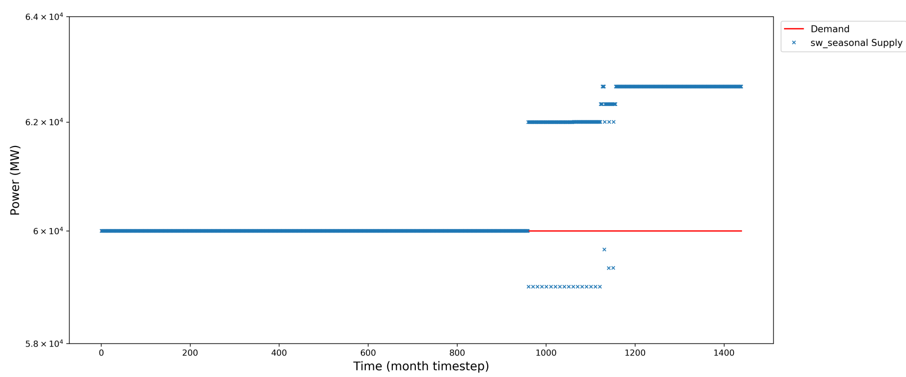
	Power		
Algorithm	No. of time steps of undersupply	Cumulative Undersupply[GW]	Cumulative Oversupply[GW]
MA	20	20.0	920.5
ARMA	18	7.7	1036.5
ARCH	0	0	1320.1
POLY	1	0.3	1783.5
EXP_SMOOTHING	20	11.0	1473.5
HOLT-WINTERS	20	11.0	1473.5
FFT	2	60.3	1751.9
SW_SEASONAL	20	18.6	1119.9



(a) NO algorithms.

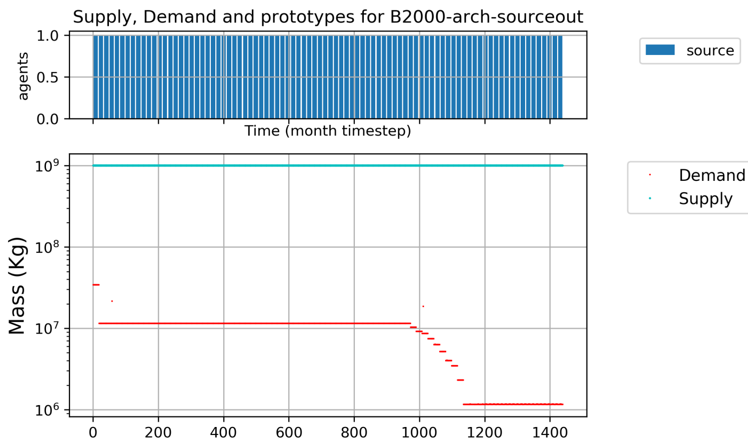


(b) DO algorithms.

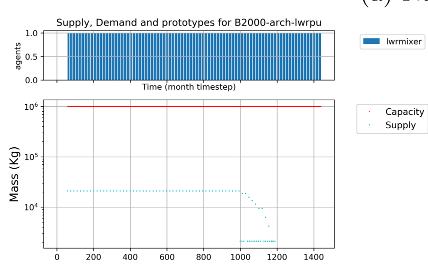


(c) SO algorithms.

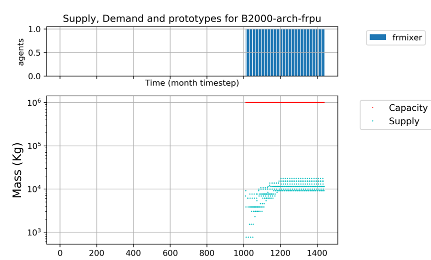
Figure 5: Plot of the power demand and supply of EG01-EG23 for a constant power demand of 60GW for different prediction algorithms.



(a) Natural-U.



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



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

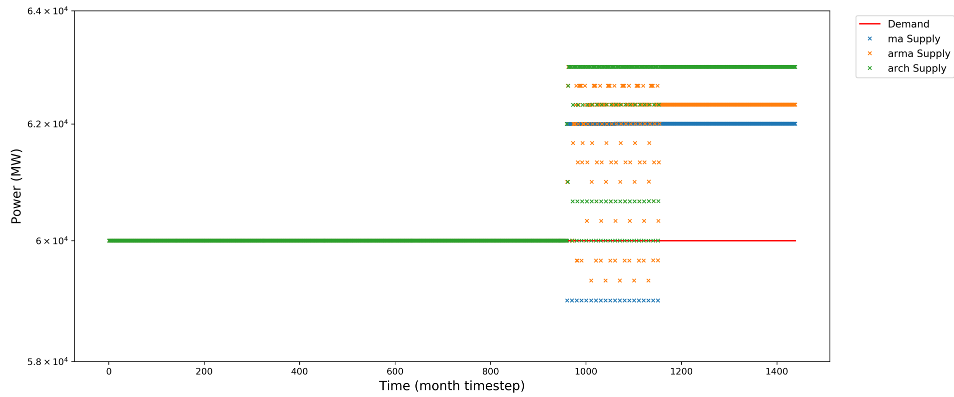
Figure 6: Plot for different commodities EG01-EG23.

Table 9: No. of time steps with undersupply and under capacity of various commodities for the different algorithms used to calculate EG01-EG24.

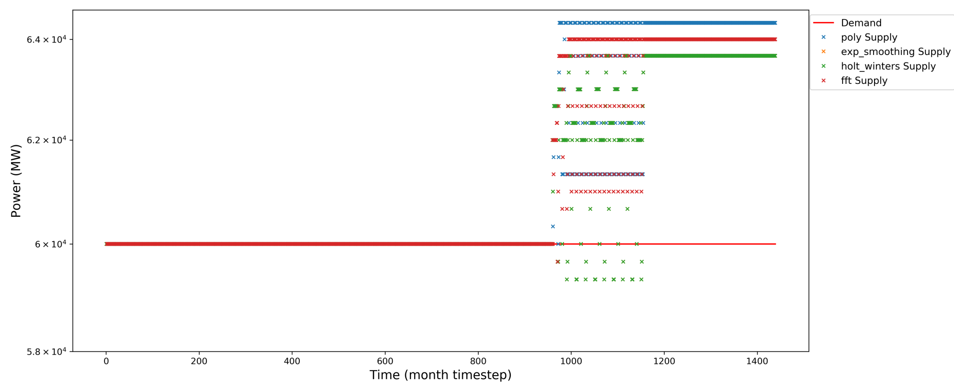
	Undersupply			Undercapcity	
Algorithm	Sourceout	Enrichmentout	FR fuel	LWR PU	FR PU
MA	0	0	0	1	1
ARMA	0	0	0	1	1
ARCH	0	0	0	1	1
POLY	0	0	0	1	1
EXP_SMOOTHING	0	0	0	1	1
HOLT_WINTERS	0	0	0	1	1
FFT	0	1	0	1	1
SW_SEASONAL	0	0	0	1	1

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

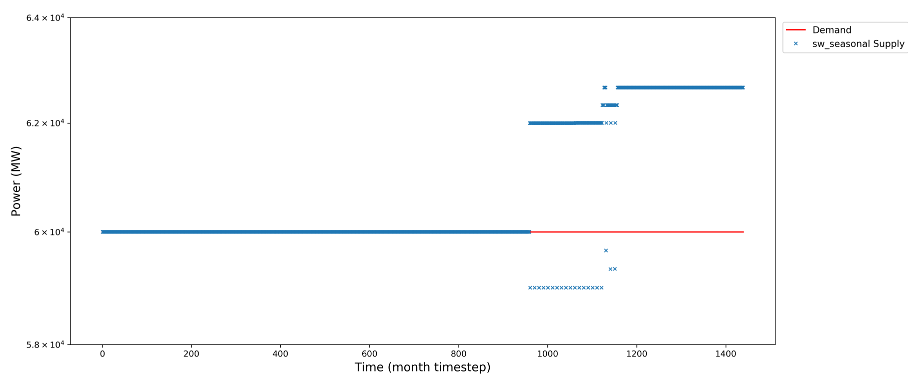
Buffer [MW]	Algorithm	MA	ARMA	POLY	EXP_SMOOTHING	FFT
0	No. of time steps of undersupply	20	60	75	30	28
	Cumulative [GW]	60.0	87.3	52.9	68.3	93.3
2000	No. of time steps of undersupply	20	18	1	20	2
	Cumulative [GW]	20.0	7.7	0.3	11.0	60.3
4000	No. of time steps of undersupply	0	0	0	0	1
	Cumulative [GW]	0	0	0	0	60.



(a) NO algorithms.



(b) DO algorithms.



(c) SO algorithms.

Figure 7: Plot of the power demand and supply of EG01-EG24 for a constant power demand of 60GW for different prediction algorithms.

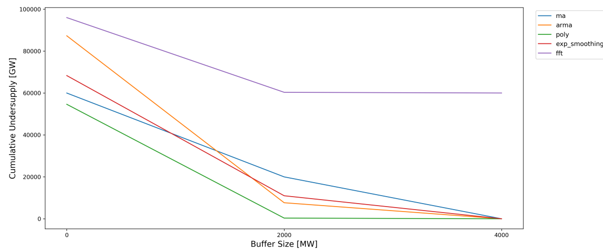


Figure 8: Plot of the dependency of the undersupply of Power on the buffer size.

Table 11: Dependency of the undersupply of Power on the no. of back steps.

No. of back steps	Algorithm	MA	ARMA	POLY	EXP_SMOOTHING	FFT
4	No. of time steps of undersupply	20	60	75	30	28
	Cumulative [GW]	60.0	87.3	52.9	68.3	93.3
8	No. of time steps of undersupply	20	18	1	20	2
	Cumulative [GW]	20.0	7.7	0.3	11.0	60.3
12	No. of time steps of undersupply	0	0	0	0	1
	Cumulative [GW]	0	0	0	0	60.

mand and supply. The buffer size was fixed to 2000 MW. Table 11 shows number of time steps of undersupply and the cumulative undersupply for different buffer sizes for some of the prediction methods. Figure 8 displays the cumulative undersupply vs the buffer sizes.

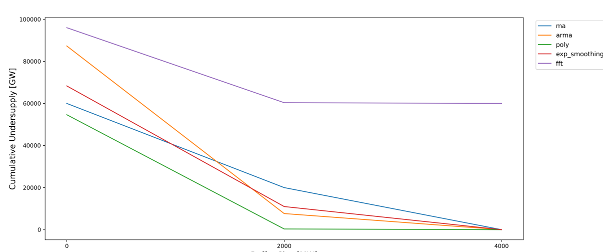


Figure 9: Plot of the dependency of the undersupply of Power on the no. of back steps.

5 Conclusion and Next Steps

This paper describes the capabilities of `d3ploy`, demonstrates the use of `d3ploy` for a simple transition scenario with constant, linearly increasing, and sinusoidal power demand. The demonstration goes a little bit further with more complex transition scenarios EG01-EG23 and EG01-EG24. This paper also provides insights on parameter inputs to ease the setting up of larger transition scenarios that include many facilities.

Future work includes setting up similar power demand transition scenarios for extended nuclear fuel cycles that incorporate different types of reactors that, consequently, use different types of fuel. Such cases are currently under study. [3] established the transition scenarios EG01-EG29 and EG01-EG30. Those 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 ends up being useful for the set up of several nuclear fuel cycles. This capability is also currently under development.

6 Acknowledgements

This research is funded by the Department of Energy (DOE) Office of Nuclear Energy's Nuclear Energy University Program (Project 16-10512) "Demand-Driven Cycamore Archetypes". The authors want to thank members of the Advanced Reactors and Fuel Cycles (ARFC) group at the University of Illinois at Urbana-Champaign. 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] 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.

[2] 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.

[3] 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.