DDCA Summer 2019 Report

Gwendolyn J. Chee, Kathryn D. Huff

May 23, 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. Constant power, linearly increasing and sinusoidal power demand transition scenarios are demonstrated. The goal is to study a basic transition scenarios and provide recommendations and insights to inform decisions about parameter inputs when setting up larger transition scenarios that include many facilities.

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 undersupply. 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 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 facility 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, deterministic optimizing and stochastic optimizing.

There are three methods implemented for the non-optimizing model: Autoregressive moving average (ARMA), autoregressive conditional heteroskedasticity (ARCH). There are four methods implemented for the deterministic optimizing model: Polynomial fit regression, simple exponential smoothing, triple exponential smoothing (holt-winters) and fast fourier transform (fft). There is one method implemented for stochastic optimizing model: stepwise seasonal.

The user can choose which prediction algorithm governs each specific `d3ploy` facility. The effectiveness of a prediction algorithm is dependent 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: `Demand- DrivenDeploymentInst` and `SupplyDrivenDeployment. Inst`. 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 facilities that exist before the reactor in a nuclear fuel cycle such as a fuel fabrication facility etc. Back-end facilities are facilities that exist after the reactor in a nuclear fuel cycle, such as a reprocessing facility etc. 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, using `DemandDrivenDeploymentInst`, it triggers the fuel fabrication facility to deploy facilities to create supply to meet the demand. Whereas, for the back end facilities, the reactor generates spent fuel , there is a demand for waste repository facility to accept the spent fuel, using `SupplyDrivenDeploymentInst`, it triggers the deployment of a waste repository to create a capacity for spent fuel to meet the available supply.

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

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.

These simulations are basic transition scenarios that only includes three types of facilities: `source`, `reactor` and `sink`. All of the simulations 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. `D3ploy` deploys reactor facilities of `new reactor` type to meet unmet demand for power that occurs when the ten initial reactor facilities begin to decommission. 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.

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 that are consisted for 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

3.1 Transition Scenario: Constant Demand

In this section, a constant power transition scenario is shown. Insights on specific parameters are discussed. Table 3 shows the simulation parameters used in this transition scenario. 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 is undersupply for each commodity in this scenario. In figure 1a, there are no time steps where the supply of power falls under demand. By using a combination of using the fast fourier transform method for predicting demand and setting the supply buffer to 3000MW (the capacity of 3 reactors), the user is able to minimize the number of time steps where there is an undersupply of every commodity. 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 the prediction methods in

Table 2: Undersupply results for each commodity in each scenario

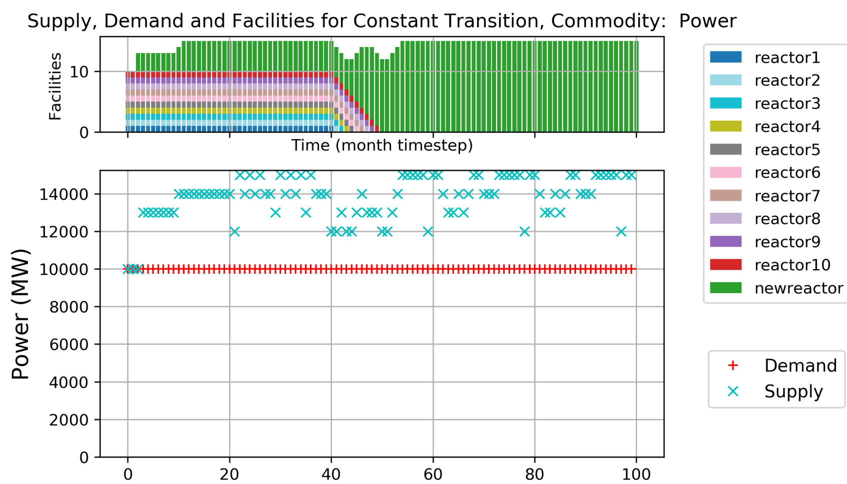
Transition Scenario	Commodity	No. of time steps with undersupply
Constant Power	Fuel	1
	Power	0
	Spent Fuel	0
Linearly Increasing Power	Fuel	1
	Power	0
	Spent Fuel	0
Sinusoidal Power	Fuel	1
	Power	1
	Spent Fuel	0

`d3ploy` are unable to predict this sudden drop in demand.

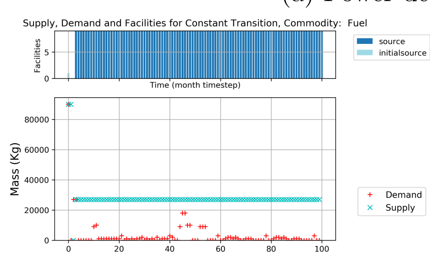
I suggest that for future simulations where a facility requires a large amount of start up commodity such as in this simulation where ten reactors start up at the same time, and thus require a large amount of start up fuel. The user should add an initial facility with a large throughput to exist for 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`.

Table 3: Constant Power Demand Transition Scenario's Parameters

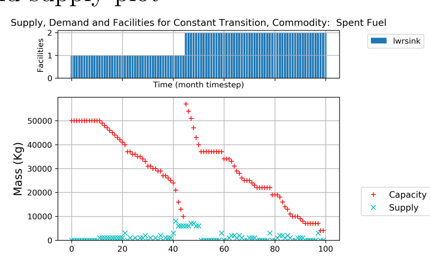
	Parameters	Description
Overall	Demand Equation	10000 MW
Power Commodity	Prediction Method	Fast Fourier Transform
	Supply Buffer	3000 MW (3 reactor capacities)
Fuel Commodity	Prediction Method	Moving Average
	Supply Buffer	0 kg
Spent Fuel Commodity	Prediction Method	Moving Average
	Capacity Buffer	0 kg



(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

3.2 Transition Scenario: Linearly Increasing Demand

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

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. The fast fourier transform method for predicting power demand is used for this scenario which is similar to what was used for the constant power demand transition scenario. A supply buffer for power of 2000MW was used which is smaller than what was required for the constant demand scenario.

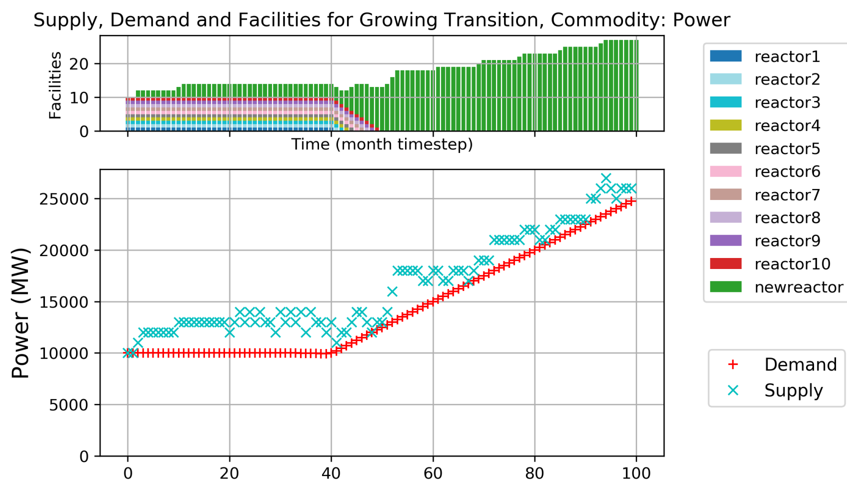
The input file used to generate this simulation can be found in */d3ploy/input/growing_transition.xml* and the file used to run the simulation and generate the plots can be found in */d3ploy/tests/performance_tests/algorithm_performance_tests_transitions.py*

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

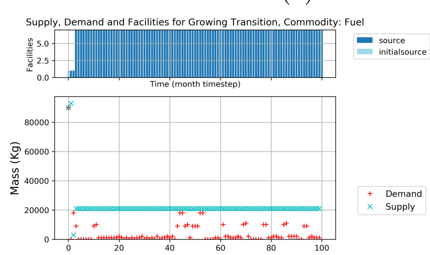
	Parameters	**Description**
Overall	Demand Equation	Time¡40: 10000 MW, Time¿40: 250*t MW
Power Commodity	Prediction Method	Fast Fourier Transform
	Supply Buffer	2000 MW (2 reactor capacities)
Fuel Commodity	Prediction Method	Moving Average
	Supply Buffer	1000 kg
Spent Fuel Commodity	Prediction Method	Fast Fourier Transform
	Capacity Buffer	0 kg

3.3 Transition Scenario: Sinusoidal Demand

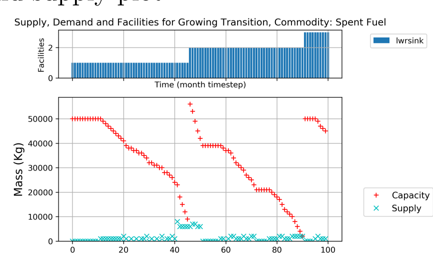
In this section, a transition scenario with sinusoidal power demand is shown. A sinusoidal power demand is the reflection of power demand in the real world where power usage is higher in the winter and summer and is smaller in the spring and fall. Table 5 shows the simulation parameters used in this transition scenario.



(a) Power demand and supply plot



(b) Fuel demand and supply plot



(c) Spent Fuel demand and supply plot

Figure 2: Transition Scenario: Linearly Increasing Power Demand

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.

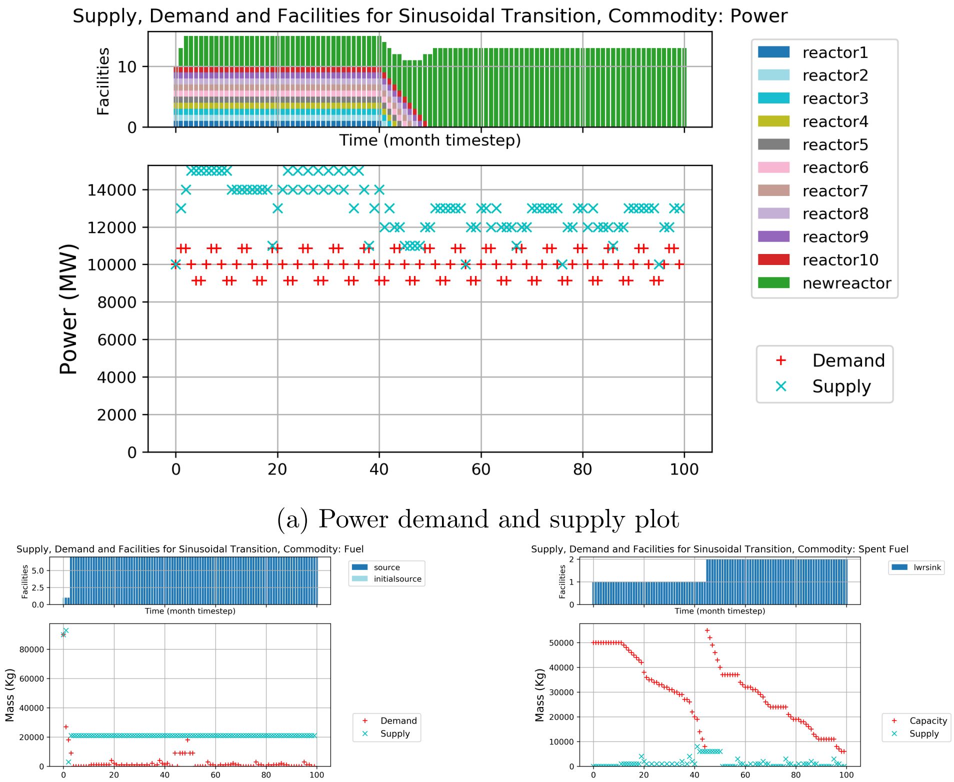
For a sinusoidal power demand, the use of the triple exponential method (holt winters) for predicting demand is more effective than the fast fourier transform method which was used for the constant and linearly increasing power demand transition scenarios. This is because the triple exponential smoothing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

I used an arbitary 1000MW for the magnitude of the sinusoidal curve. There is probably information in literature that tells you the seasonal percentage change in demand.

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
$/d3ploy/tests/performance\_tests/algorithm\_performance\_tests\_transitions.py$

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 (2 reactor capacities)
Fuel Commodity	Prediction Method	Moving Average
	Supply Buffer	1000 kg
Spent Fuel Commodity	Prediction Method	Fast Fourier Transform
	Capacity Buffer	0 kg



(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 Conclusion

This paper describes the capabilities of `d3ploy`, demonstrates the use of `d3ploy` for an assortment of transition scenarios: constant power demand, linearly increasing power demand and sinusoidal power demand. It 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 reprocessing facilities etc. A more realistic transition scenario could be explored such as an increasing power demand that has a sinusoidal pattern to represent seasons in a year for a growing power demand trend. I am aware that Roberto is currently setting up the transition scenarios as a constant power demand, but the above type of scenario (growing sinusoidal) could be an interesting transition scenario to do. Perhaps, since the EG01-EG23 scenario is more or less completed, Sonata, you could look into just changing the power demand equation to be a growing sinusoidal and see if the scenario can be adapted for this new power demand.

5 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.