DDCA Summer 2019 Report

Gwendolyn J. Chee, Kathryn D. Huff

May 21, 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. This report discusses demonstration of a new solution, `d3ploy`, enabling demand driven deployment in Cyclus.

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 report, we will explain the capabilities of `d3ploy`, demonstrate how `d3ploy` can be used in a flat power transition to meet the primary objective of minimizing undersupply of all commodities in the simulation. The goal is to study a basic flat power demand scenario and provide recommendations and insights to inform decisions about parameter inputs when setting up larger flat power demand 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 repository [https://github.com/arfc/d3ploy].

Essentially, the user must define the facilities they want the institution to control and their corresponding capacities. The user must also define the driving commodity, its demand equation and what method they want the institution to predict demand and supply with.

They also have the option to give an 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.

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 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 Methods for demonstration of d3ploy capabilities

A flat power simulation is set up to demonstrate `d3ploy`'s capabilities. A balance between the various system parameters must be met for each type of simulation to ideally meet the goal of minimizing undersupply and under capacity for the various commodities.

This simulation is a basic transition scenario that only includes three types of facilities: `source`, `reactor` and `sink`. The reason for setting up a basic flat power demand scenario is to inform decisions about parameter inputs when setting up larger flat power demand transition scenarios that include many facilities.

3.1 Transition Scenario: Flat Demand

In this section, a flat power transition scenario is studied and recommendations and insights on specific parameters are discussed. Table 1 shows the parameters used for 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_constant_transition.py

The ten initial reactors in the simulation have different cycle lengths so that they refuel at different times. In figure 1, there are two time steps where supply falls under demand, this is due to the combined refueling of multiple `new reactors`. By 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. 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. If the `new reactors` were configured such that they had no refueling, there will be no time steps where supply falls under demand. This is shown in figure 2, however, a reactor without refueling is unrealistic. Future work should allow for randomization of reactor cycle length so that refueling occurs at different times when the same reactor facility type is deployed to avoid the issue where all the same reactors that were deployed at the same time step having to refuel at the same time.

Figure 3 shows the demand and supply for fuel in the simulation. An initial facility with a large capacity of fuel is initially deployed to meet the large initial fuel demand for the starting up of ten reactors. There are no time steps where there is an undersupply. 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 1: Transition Scenario Flat Demand Parameters

Parameter	
Facilities Present	`Source` (Capacity: 3000kg), `Reactor` (Capacity: 1000MW), `Sink`
New Reactor Parameters	Cycle time: 18, Refuel time: 1
Driving Commodity	`Power`
Demand Equation	10000 MW
Supply Buffer	3000 MW (3 reactor capacities)

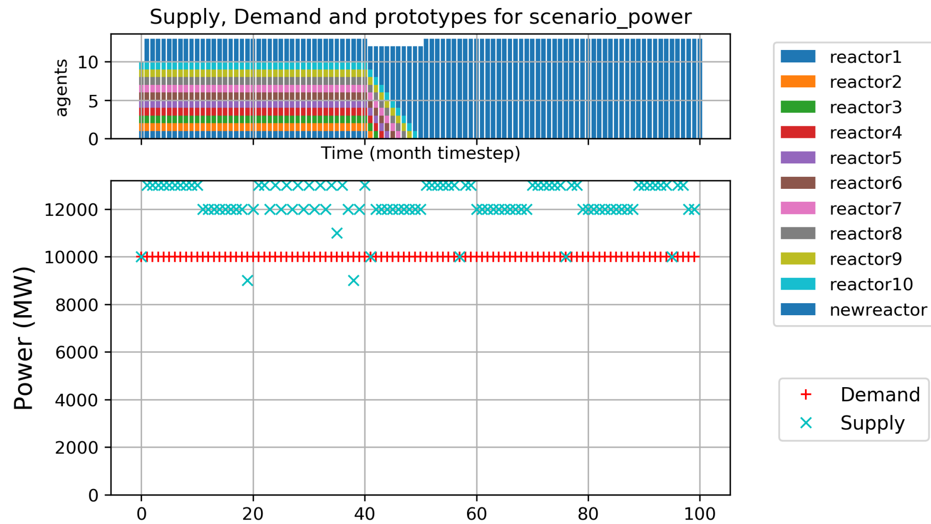

Figure 1: Power demand and supply plot for a flat power demand transition scenario where new reactor has cycle length = 20, refueling length time = 1

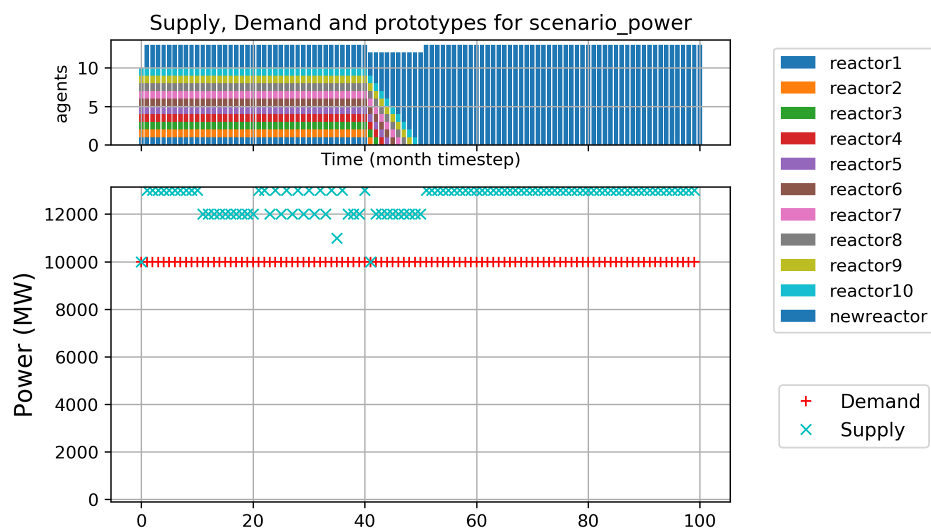

Figure 2: Power demand and supply plot for a flat power demand transition scenario where new reactor has cycle length = 20, refueling length time = 0

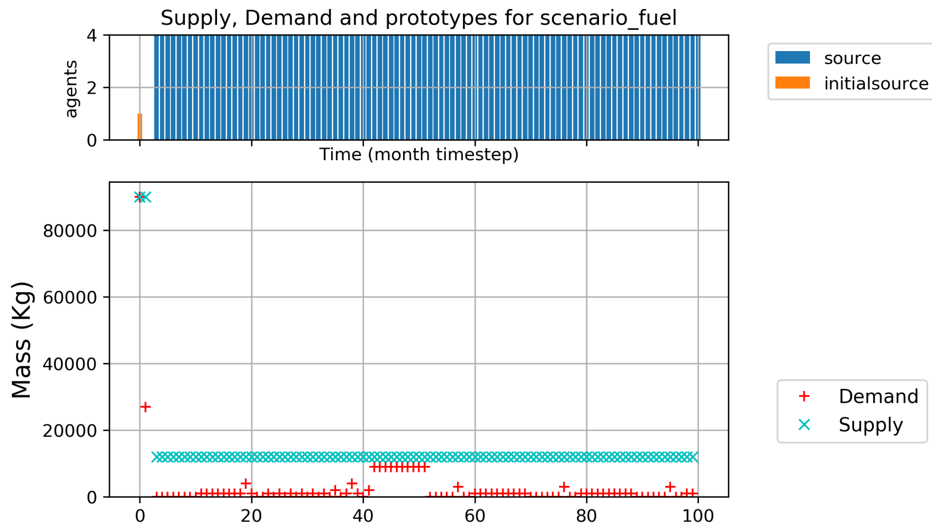

Figure 3: Fuel demand and supply plot for a flat power demand transition scenario where new reactor has cycle length = 20, refueling length time = 1

4 Conclusion

This report describes the capabilities of `d3ploy`, demonstrates the use of `d3ploy` in a basic flat power transition scenario and provides insights on parameter inputs when setting up larger flat power demand transition scenarios that include many facilities.

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.