

Numerical Experiments for Verifying Demand Driven Deployment Algorithms

Authors

Jin Whan BAE
Gwendolyn CHEE

Principal Investigator

Kathryn D. HUFF

UIUC-ARFC-2018-01

April 23, 2018

ADVANCED REACTORS AND FUEL CYCLES

DEPT. OF NUCLEAR, PLASMA, & RADIOLOGICAL ENGINEERING
UNIVERSITY OF ILLINOIS 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.

Numerical Experiments for Verifying Demand Driven Deployment Algorithms

Jin Whan Bae, Gwendolyn Chee, Kathryn Huff

April 21, 2018

1 Introduction

For many fuel cycle simulations, it is currently up to the user to define a deployment scheme, or facility parameters, to make sure that there's no gap in the supply chain. Or, the same goal is achieved by setting the **facility** capacity to infinity, which does not reflect real-world conditions.

The Demand-Driven Cycamore Archetype project (NEUP-FY16-10512) aims to develop CYCAMORE demand-driven deployment capabilities. The developed algorithm, in the form of CYCLUS **Institution** agent, deploys **Facilities** to meet the front-end and back-end demands of the fuel cycle.

This report describes numerical tests for non-optimizing, deterministic-optimizing and stochastic-optimizing prediction algorithms.

These prediction models are being developed by the University of South Carolina. In this report, we discuss numerical experiments for testing the non-optimizing, deterministic optimizing and stochastic optimizing methods. The numerical experiments will be designed for both the once through nuclear fuel cycle and advanced fuel cycles.

2 Method

This report lists necessary capabilities of the new CYCLUS **institution** for demand-driven deployment of fuel cycle facilities. Then the report lists tests to check correct implementation of the capabilities, with a sample fuel cycle with well-defined facility parameters.

3 Configuration

The user defines prototypes to be deployed for fuel facilities, and reactor deployment scheme. The reactor deployment causes the demand of fuel which triggers fuel facility deployment. The detailed input file XML input schema is shown in Appendix A.

4 Algorithm Flow

The algorithm, upon entering, creates a supply chain with the fuel facilities and the reactor. Then, at every timestep it calculates the expected demand from each fuel cycle facility and makes decisions to deploy or decommission. As a reference, the time step execution for CYCLUS is illustrated in figure 1.

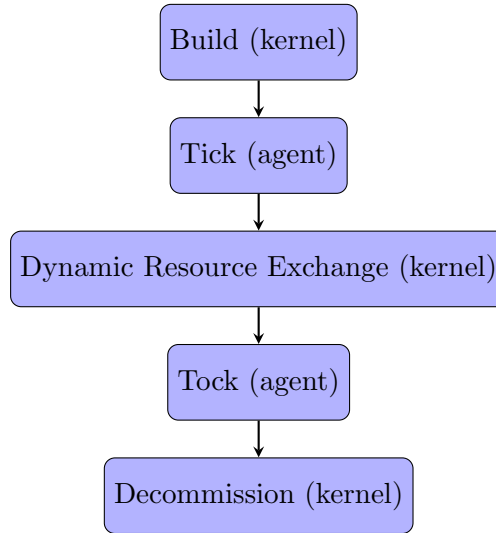


Figure 1: Each timestep in CYCLUS follows the five steps in order. Processes labeled kernel are executed by the CYCLUS framework, whereas processes labeled agent are executed by individual agents. What happens in the ‘Tick’ and ‘Tock’ is thus unique to each archetype.

4.1 Upon Entering (Enternotify)

The algorithm creates a supply chain with the fuel facilities, then calculates the demand for each facility for a unit quantity of fuel (example in figure 2). It also orders to build fuel cycle facilities for the reactors at timestep 1.



Figure 2: Simple demand flow of materials. The values in the bracket are demands calculated by the algorithm. The **Reactor** demands x amount of fuel, which translates into demands of x from **Sink**.

4.2 Tick

The algorithm calculates the fuel demand from the fleet of reactors at one timestep ahead, and the corresponding demand for fuel facilities. The current capacity of each fuel cycle facility is also calculated. If the capacity is smaller than the demand, the algorithm orders to build more facilities to meet the demand, so that the fuel demand is met for the next timestep.

5 Simulation parameter for Test Scenarios

Simple parameters are given to fuel cycle facilities for the numerical testing of the algorithm. Only **source** and **reactor** facilities are used in the test scenarios.

Table 1 provides basic parameters for each test scenario. Table 2 provides the parameters for the **source**, **reactors** and **sink** in the test scenarios.

Table 1: Basic Test Parameters

Test Scenario Parameters	Value	Units
Duration	15	timesteps
Timestep	1	month
Start Month	1	month
Start Year	2000	year

Table 2: Source, Reactor and Sink Parameters

Source Parameters	Value	Units
Throughput	1	kg
Output Commodity	fuel	kg
Reactor Parameters	Value	Units
Cycle Time	1	timesteps
Refuel Time	0	timesteps
Lifetime	1	timesteps
Power Capacity	1	MWe
Assembly Size	1	kg
# assemblies per core	1	
# assemblies per batch	1	
Input Commodity	fuel	kg
Output Commodity	power	MW
Sink Parameters	Value	Units
Throughput	1	kg
Input Commodity	spent uox	kg

6 Numerical Tests for the Non-optimizing prediction method

The non-optimizing prediction method is tested by comparing its output for various scenarios against their analytical solutions. In this section, the tests that must be met is described based on the parameters defined in table 1 and 2 and analytical solution of a defined simple scenario. Unit test examples are included in Appendix B.

The tests are split into test A types and test B types. Test A refers to the test scenarios where facilities are expected to be deployed. Test B refers to the test scenarios where facilities are expected to be decommissioned.

The prediction algorithm for the non-optimizing method has three user-defined input parameters. The aim of the various test scenarios are to check if the non-optimizing method archetype will deploy or decommission facilities correctly when there is a variation in the combination of the three input parameters. The input parameters are:

1. Initial demand value
2. Number of initial facilities (initial supply)
3. Growth rate of initial demand

The growth in demand is governed by the Equation 1.

$$D_f(timestep) = D_i(1 + g)^{(\frac{timestep}{12})} \quad (1)$$

Where D_f is demand of resource at specific time step, D_i is initial demand and g is growth rate.

Source and **reactor** facilities are used in the test scenarios. Test scenarios A1 to A4 and B1 to B2 only have a **source** facility and test scenarios A5 to A7 have both **source** and **reactor** facilities. For each test scenario, there is one table that states the test scenario's input parameters and another table that states the exact analytical solution. The analytical solution table does not include deployment of the initial facility that is stated in the first table.

Additionally, we created base tests for each A-type test scenario that passes when the supply meets the demand within a given facility number tolerance. In other words, when the supply exceeds the demand by the specified tolerance quantity, the test still passes. For this report, the tolerance is set to one facility. For example in test A-1, the expected total number

facilities deployed is 1, and since the facility over-prediction tolerance is 1, the acceptable range of total number of facilities (x) deployed in the entire test scenario is $1 < x < 2$. If the total number of facilities deployed is within this range, the base case test will pass.

6.1 Test A-1

In test A-1, only a **source** facility is present in the test scenario. Table 3 shows the input parameters of the source facility in the test scenario. Table 4 shows the expected analytical solution based on the test scenario. Table 5 shows the accepted range of total number of facilities deployed over the test scenario which will pass the base test, which factors in the facility over-prediction tolerance of 1.

Table 3: Test A-1 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	1	kg
Initial facilities	0	#
Growth Rate	0	

Table 4: Test A-1 Analytical Solution

Time Step	No. of Source Facilities Deployed
1	1
2 to 15	0

Table 5: Test A-1 Base Test Acceptance

Acceptable total No. of Source Facilities Deployed + tolerance
$1 < x < 2$

6.2 Test A-2

In test A-2, only a **source** facility is present in the test scenario. Table 6 shows the input parameters of the source facility in the test scenario. Table 7 shows the expected analytical solution based on the test scenario. Table 8 shows the accepted range of total number of facilities deployed over the

test scenario which will pass the base test, which factors in the facility over-prediction tolerance of 1.

Table 6: Test A-2 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	2	kg
Initial facilities	1	#
Growth Rate	0	

Table 7: Test A-2 Analytical Solution

Time Step	No. of Source Facilities Deployed
1	1
2 to 15	0

Table 8: Test A-2 Base Test Acceptance

Acceptable total No. of Source Facilities Deployed + tolerance
$2 < x < 3$

6.3 Test A-3

In test A-3, only a **source** facility is present in the test scenario. Table 9 shows the input parameters of the source facility in the test scenario. Table 10 shows the expected analytical solution based on the test scenario. Table 11 shows the accepted range of total number of facilities deployed over the test scenario which will pass the base test, which factors in the facility over-prediction tolerance of 1.

Table 9: Test A-3 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	1	kg
Initial facilities	0	#
Growth Rate	1	

Table 10: Test A-3 Analytical Solution

Time Step	No. of Source Facilities Deployed
1	2
2 to 12	0
13	1
14 to 15	0

Table 11: Test A-3 Base Test Acceptance

Acceptable total No. of Source Facilities Deployed + tolerance
$3 < x < 4$

6.4 Test A-4

In test A-4, only a **source** facility is present in the test scenario. Table 12 shows the input parameters of the source facility in the test scenario. Table 13 shows the expected analytical solution based on the test scenario. Table 14 shows the accepted range of total number of facilities deployed over the test scenario which will pass the base test, which factors in the facility over-prediction tolerance of 1.

Table 12: Test A-4 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	1	kg
Initial facilities	1	#
Growth Rate	1	

Table 13: Test A-4 Analytical Solution

Time Step	No. of Source Facilities Deployed
1	1
2 to 12	0
13	1
14 to 15	0

Table 14: Test A-4 Base Test Acceptance

Acceptable total No. of Source Facilities Deployed + tolerance
$3 < x < 4$

6.5 Test A-5

In test A-5, both a **source** and **reactor** facility is present in the test scenario. Table 15 shows the input parameters of the source facility in the test scenario. Table 16 shows the expected analytical solution based on the test scenario. Table 17 shows the accepted range of total number of facilities deployed over the test scenario which will pass the base test, which factors in the facility over-prediction tolerance of 1.

Table 15: Test A-5 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	1	kg
Initial facilities	0	#
Growth Rate	0	
Reactor Parameter	Value	Units
Initial demand	1	MW
Initial facilities	0	#
Growth Rate	0	

Table 16: Test A-5 Analytical Solution

Time Step	No. of Source Facilities Deployed	No. of Reactor Facilities Deployed
1	1	1
2 to 15	0	0

Table 17: Test A-5 Base Test Acceptance

Acceptable total No. of Source Facilities Deployed + tolerance	Acceptable total No. of Reactor Facilities Deployed + tolerance
$1 < x < 2$	$1 < x < 2$

6.6 Test A-6

In test A-6, both a **source** and **reactor** facility is present in the test scenario. Table 18 shows the input parameters of the source facility in the test scenario. Table 19 shows the expected analytical solution based on the test scenario. Table 23 shows the accepted range of total number of facilities deployed over the test scenario which will pass the base test, which factors in the facility over-prediction tolerance of 1.

Table 18: Test A-6 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	1	kg
Initial facilities	1	#
Growth Rate	0	
Reactor Parameter	Value	Units
Initial demand	1	MW
Initial facilities	1	#
Growth Rate	0	

Table 19: Test A-6 Analytical Solution

Time Step	No. of Source Facilities Deployed	No. of Reactor Facilities Deployed
1	1	1
2 to 15	0	0

Table 20: Test A-6 Base Test Acceptance

Acceptable total No. of Source Facilities Deployed + tolerance	Acceptable total No. of Reactor Facilities Deployed + tolerance
$1 < x < 2$	$1 < x < 2$

6.7 Test A-7

In test A-7, both a **source** and **reactor** facility is present in the test scenario. Table 21 shows the input parameters of the source facility in the test scenario. Table 22 shows the expected analytical solution based on the test scenario. Table ?? shows the accepted range of total number of facilities

deployed over the test scenario which will pass the base test, which factors in the facility over-prediction tolerance of 1.

Table 21: Test A-7 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	1	kg
Initial facilities	0	#
Growth Rate	1	
Reactor Parameter	Value	Units
Initial demand	1	MW
Initial facilities	0	#
Growth Rate	1	

Table 22: Test A-7 Analytical Solution

Time Step	No. of Source Facilities Deployed	No. of Reactor Facilities Deployed
1	2	2
2 to 12	0	0
13	1	1
14 to 15	0	0

Table 23: Test A-7 Base Test Acceptance

Acceptable total No. of Source Facilities Deployed + tolerance	Acceptable total No. of Reactor Facilities Deployed + tolerance
$3 < x < 4$	$3 < x < 4$

6.8 Test B-1

In test B-1, only a **source** facility is present in the test scenario. Table 24 shows the input parameters of the source facility in the test scenario. Table 25 shows the expected analytical solution based on the test scenario.

Table 24: Test B-1 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	0	kg
Initial facilities	1	#
Growth Rate	0	

Table 25: Test B-1 Analytical Solution

Time Step	No. of Source Facilities Deployed	No. of Source Facilities Decomissioned
1	1	0
2	0	1
3 to 15	0	0

6.9 Test B-2

In test B-2, only a **source** facility is present in the test scenario. Table 26 shows the input parameters of the source facility in the test scenario. Table 27 shows the expected analytical solution based on the test scenario.

Table 26: Test B-2 Scenario Input Parameters

Source Parameter	Value	Units
Initial demand	1	kg
Initial facilities	1	#
Growth Rate	-1	

Table 27: Test B-2 Analytical Solution

Time Step	No. of Source Facilities Deployed	No. of Source Facilities Decomissioned
1 to 12	1	0
13	0	1
14 to 15	0	0

7 Numerical Test Results

It was found that none of the exact tests and base case tests passed. The failures are attributed to three reasons:

1. There is a test failure when there is no initial facility present. An initial condition must be given for the algorithm to understand the capacity of the facility it deploys.
2. There is a test failure when there is growth in the demandhe algorithm failed to deploy facilities to meet the increase in demand.
3. There is a test failure when facilities are expected to be decommissioned. There is yet to be an implementation on decommissioning behavior.

Despite failures in these situations, the non-optimizing method proved to have commissioning capabilities to meet demand. Because of the limitation in the algorithm, it is hard to predict demand with precision in such a short time. However, the same numerical experiments can be applied to the deterministic optimizing algorithm, which holds more promise. The goal is that the deterministic optimization method will be able to overcome the issues faced by the non-optimizing method with reference to the requirement of an initial condition, growth in demand. Also, the capability to decommission facilities upon oversupply will be added.

Appendix C reflects the numerical experiment solution output by the non-optimizing prediction algorithm for each test scenario defined in section 6.

Appendix A - parameter configuration

Appendix A shows the json file that contains the simulation parameters that are common between all the test scenarios discussed in Section 6.

```
template = {
  "simulation": {
    "archetypes": {
      "spec": [
        {"lib": "agents", "name": "NullRegion"},
        {"lib": "cycamore", "name": "Source"},
        {"lib": "cycamore", "name": "Reactor"},
        {"lib": "cycamore", "name": "Sink"},
        {"lib": "d3ploy.no_inst", "name": "NOInst"}
      ]
    },
    "control": {"duration": "15", "startmonth": "1", "startyear": "2000"},
    "recipe": [
      {
        "basis": "mass",
        "name": "fresh_uox",
        "nuclide": [{"comp": "0.711", "id": "U235"}, {"comp": "99.289", "id": "U238"}]
      },
      {
        "basis": "mass",
        "name": "spent_uox",
        "nuclide": [{"comp": "50", "id": "Kr85"}, {"comp": "50", "id": "Cs137"}]
      }
    ],
    "facility": [{
      "config": {"Source": {"outcommod": "fuel",
        "outrecipe": "fresh_uox",
        "throughput": "1",
        "source_record_supply": "fuel"}}},
      {"name": "source"},
    ],
    {
      "config": {"Sink": {"in_commods": {"val": "spent_uox"},
        "max_inv_size": 1,
        "sink_record_demand": "fuel_cap"}}},
    ]
  }
}
```

```

"name": "sink"
},
{
  "config": {
    "Reactor":{
      "assem_size": "1",
      "cycle_time": "1",
      "fuel_incommods": {"val": "fuel"},
      "fuel_inrecipes": {"val": "fresh_uox"},
      "fuel_outcommods": {"val": "spent_uox"},
      "fuel_outrecipes": {"val": "spent_uox"},
      "n_assem_batch": "1",
      "n_assem_core": "1",
      "power_cap": "1",
      "refuel_time": "0",
      "reactor_fuel_demand": "fuel_reactor"
    }
  },
  "name": "reactor"
}}}]

```


Appendix B - Sample Test Code

Sample test code for test A-1

Appendix B shows the python file that contains the a segment of the simulation parameters that are unique to test A-1 and the code for test A-1.

```
# Test A_1
INIT_DEMAND = copy.deepcopy(TEMPLATE)
INIT_DEMAND["simulation"].update({"region": {
    "config": {"NullRegion": "\n        "},
    "institution": {
        "config": {
            "NOInst": {
                "calc_method": "arma",
                "demand_commod": "POWER",
                "demand_std_dev": "0.0",
                "growth_rate": "0.0",
                "initial_demand": "1",
                "prototypes": {"val": "source"},
                "steps": "1",
                "supply_commod": "fuel"
            }
        },
        "name": "source_inst"
    },
    "name": "SingleRegion"
}})

@pytest.mark.base
def test_a1_init_demand():
    # tests if NOInst deploys a source
    # given initial demand and no initial facilities
    output_file = 'init_file.sqlite'
    input_file = output_file.replace('.sqlite', '.json')
    with open(input_file, 'w') as f:
        json.dump(INIT_DEMAND, f)
    s = subprocess.check_output(['cyclus', '-o', output_file, input_file],
        universal_newlines=True, env=ENV)
    # check if ran successfully
    assert("Cyclus run successful!" in s)
```

```

# getting the sqlite file
cur = get_cursor(output_file)

# check base solution
source_base = cur.execute(query).fetchone()
assert(1 <= source_base[0] <= (1 + tol))

@pytest.mark.exact
def test_a1_init_demand_exact():
    output_file = 'init_file.sqlite'
    cur = get_cursor(output_file)
    # check exact solution
    source_exact = cur.execute(query + " AND EnterTime = 1").fetchone()
    assert(source_exact[0] == 1)

```

Appendix C - Numerical Experiment Solution for test scenarios

Test A-1

Table 28: Test A-1 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed
1 to 15	0

Test A-2

Table 29: Test A-2 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed
1	0
2	1
3	1
4	1
5	0
6	1
7 to 15	0

Test A-3

Table 30: Test A-3 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed
1 to 15	0

Test A-4

Table 31: Test A-4 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed
1 to 15	0

Test A-5

Table 32: Test A-5 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed	No. of Reactor Facilities Deployed
1 to 15	0	0

Test A-6

Table 33: Test A-6 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed	No. of Reactor Facilities Deployed
1	0	0
2	1	0
3	1	0
4	1	0
5	0	0
6	1	0
7 to 15	0	0

Test A-7

Table 34: Test A-7 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed	No. of Reactor Facilities Deployed
1 to 15	0	0

Test B-1

Table 35: Test B-1 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed	No. of Source Facilities Decomissioned
1 to 15	0	0

Test B-2

Table 36: Test B-2 Numerical Experiment Solution

Time Step	No. of Source Facilities Deployed	No. of Source Facilities Decomissioned
1 to 15	0	0