Numerical Experiments for Verifying Demand Driven Deployment Algorithms

Draft 2

Jin Whan Bae and Gwendolyn Chee

2017-10-11

Introduction

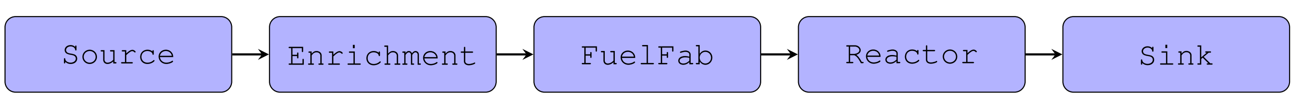
The Demand-Driven Cycamore Archetype project (NEUP-FY16-10512) aims to develop CYCAMORE demand-driven deployment capabilities. The project plans to use non-optimizing, deterministic-optimizing and stochastic-optimizing prediction algorithms.

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.

1 Once through Nuclear Fuel Cycle

This section defines the required tests for each method assuming a once-through fuel cycle.

Figure 1: Flow Chart of Once through Nuclear Fuel Cycle



2 Input File Specification

We assume the archetype to be an `INSTITUTION`, since it governs deployment and decommission of facilities.

The user would only have to define the reactor prototype and reactor deployment. The remaining fuel facilities, both front end and back end, would be recognized by the archetype by looking at each prototypes' in-and-out commodities. Then the recognized, or 'connected' fuel cycle facilities deploy with demand. If an input file does not have the necessary 'connections' for the reactor to receive fuel, it will throw an error.

We suggest an example format: First, the input file would define reactor prototype to be deployed. There can be multiple reactors.

```
<institution>
      <name>NO_ddd</name>
       <config><NO_ddd/></config>
      <reactor_list>
        <val>lwr</val>
        <val>sfr</val>
        <val>mox_lwr</val>
      </reactor_list>
```

Listing 1: One-reactor fleet institution input schema

[Something About Transition Scenario Capabilities]

Then deployment of the reactors is defined, and the user is given the option to manually define the deployment(DeployInst):

```
      <deployment>
      <type>manual</type>

        <build_times>
          <val>1</val>
          <val>10</val>
          <val>20</val>
          <val>40</val>
        </build_times>
        <n_build>
          <val>3</val>
          <val>3</val>
          <val>3</val>
          <val>3</val>
        </n_build>
        <lifetimes>
          <val>960</val>
          <val>960</val>
          <val>960</val>
          <val>960</val>
        </lifetimes>
      </deployment>

```

```
</institution>
```

Listing 2: Reactor deployment input schema

or have the institution deploy reactors according to power demand (GrowthRegion):

```
    <deployment>
    <type>growth</type>

    <growth>
      <piecewise_function>
            <piece>
              <start>0</start>
              <function>
                <type>linear</type>
                <params>1 2</params>
              </function>
            </piece>
      </piecewise_function>
    </growth>

    </deployment>

  </institution>
```

Listing 3: Reactor deployment input schema

2.1 Non-optimizing prediction method

The following conditions need to be satisfied for each segment of the fuel cycle.

2.1.1 `Reactor`

1. Do all the `Reactor`s run at full capacity (not lacking fuel)?
2. Is a new `Reactor` deployed when the energy demand exceeds the energy produced by the current `Reactor`s?
3. Is a `Reactor` decommissioned when the energy demand falls behind the output of the current `Reactor`s?

2.1.2 `Fuelfab`

1. Is the fuel produced by the `Fuelfab` within the upper limit of [insert uncertainty] for the analytic solution of fuel required by the `Reactors` for all of them to run for each time step?

2. Is a new `Fuelfab` deployed when the fuel required by the reactors exceeds the output of current `Fuelfab`?

3. Is a `Fuelfab` decommissioned when the fuel required by the reactors falls behind the output of current `Fuelfab` facilities?

2.1.3 `Enrichment`

1. Is the enriched uranium produced by `Enrichment` within the upper limit of [insert uncertainty] for the analytic solution of the enriched uranium required by the `Fuelfab` for each time step?

2. Is a new `Enrichment` facility deployed when the enriched uranium required by the `Fuelfab` exceeds the enriched uranium produced of current `Enrichment` facilities?

3. Is a `Enrichment` decommissioned when the enriched uranium required by the `Fuelfab` falls behind the enriched uranium produced by current `Enrichment` facilities?

2.1.4 `Source`

1. Is the mined uranium produced by `Source` within the upper limit of [insert uncertainty] for the analytic solution of the mined uranium required by the `Enrichment` facilities for each time step?

2. Does the `Source` mined uranium produced increase when mined uranium required by the `Enrichment` facilities exceeds mined uranium produced by the current `Source`?

3. Does the `Source` mined uranium produced decrease when mined uranium required by the `Enrichment` facilities falls behind the mined uranium required by the current `Source`?

2.2 Deterministic-Optimizing/Stochastic prediction method

The following conditions need to be satisfied for each segment of the fuel cycle.

1. Do all the `Reactors` run at full capacity (not lacking fuel)?

```
TEST(ReactorTests, DDDeploy_DO) {
    [Example input with the following attributes:]
        [int simdur = 20;]
        [Defines reactor with zero refueling cycle and operation cycle of
            1 month]
        [Defines fuel cycle facilities parameters]
        [Defines Reactor Deploy Scheme / Power Demand]
        [Increasing Fuel Demand with Time]
    [Run test]
    [Test if Reactor has no zero values in output Timeseriespower]
}
```

Listing 4: Test to see all reactors run without lack of fuel

2. Is the objective function optimized?

3. Is the constraint followed?

4. Do the related fuel cycle facilities get deployed upon demand?

```
TEST(ReactorTests, DDDeploy_NO) {
    [Example input with the following attributes:]
        [int simdur = 20;]
        [Defines reactor with zero refueling cycle and operation cycle of
            1 month]
        [Defines fuel cycle facilities parameters]
        [Defines Reactor Deploy Scheme / Power Demand]
        [Increasing Fuel Demand with Time]
    [Run test]
    [Test if fuel facility is deployed in the beginning]
    [Test if fuel facility is deployed later in the simulation (have
        analytic solution)]
}
```

Listing 5: Test demand-driven deployment of fuel cycle facility

5. Do the related fuel cycle facilities exit upon demand decrease?

```
TEST(ReactorTests, DDDeploy_NO) {
    [Example input with the following attributes:]
        [int simdur = 20;]
        [Defines reactor with zero refueling cycle and operation cycle of
            1 month]
        [Defines fuel cycle facilities parameters]
```

```
        [Defines Reactor Deploy Scheme / Power Demand]
        [Decreasing Fuel Demand with Time]
    [Run test]
    [Test if fuel facility is deployed in the beginning]
    [Test if fuel facility exits later in the simulation (have analytic
       solution)]
}
```

Listing 6: Test demand-driven exit of fuel cycle facility

3 Advanced Fuel Cycles