Introduction to Cyclus

Paul Wilson
and the Cyclus Development Team
Cyclus Development Team

- University of Wisconsin
  - Nuclear Engineering: Robert Carlsen, Vincent Cloitre, Matthew Gidden, Michael Gionet, Kathryn Huff\(^1\), Meghan McGarry, Baptiste Mougnot, Arrielle Opotowsky, Olzhas Rakhimov, Zach Welch, Paul Wilson
  - Life Science Communication: Ashley Anderson, Dominique Brossard, Nan Li, Dietram Scheufele

- University of Texas
  - Nuclear Engineering: Cem Bagdatlioglu, Erich Schneider

- University of South Carolina
  - Nuclear Engineering: Anthony Scopatz, Robert Flanagan

- University of Utah
  - Computer Science: Haya Agur, Yarden Livnat

- University of Idaho
  - Computer Science: Robert Hiromoto, Teva Velupillai

\(^1\) Currently University of Illinois at Urbana-Champaign
Overview

● Fuel Cycle Simulators Background
● Next Generation Fuel Cycle Simulator
● Cyclus History
● Cyclus Strategy
● Moving Forward
Fuel Cycle Simulator - Purpose

Track mass flows and facility deployments during transition between alternative nuclear fuel cycles
Fuel Cycle Simulator - Purpose

- Evaluate environmental and socio-economic impact
- Inform technical and non-technical decision-makers
Motivations for New Simulators

**Flexibility**
- Accommodate innovative systems and cycles
- Carefully study impacts of modeling choices
- Implement as part of optimization and sensitivity analysis
- Minimize inherent technology assumptions
- Allow for maximum fidelity
  - Discrete facilities with discrete material quanta

**Accessibility**
- Open source development using commonly available tools
Cyclus Development Strategy

1. Open source simulation kernel

2. Ecosystem of plug-in modules

3. Open source analysis and visualization tools
Agent-Based Approach
- Encapsulate physics and interaction behavior in each Facility
- Each facility operated by an Institution in a geopolitical Region

Dynamic Resource Exchange
- Constant deployment gives changing material flow paths
- Material substitution complicates matching of supply/demand

Discrete Material Tracking
- Enable analysis based on tracking history of individual material objects
- Investigate: transportation, forensics, etc.
Cyclus Kernel Basics: Simulation

- Provides ecosystem for agent interaction and resource transactions

- Steps through time in uniform increments
  - Regions request that Institutions deploy Facilities
  - Dynamic Resource Exchange determines resource transactions
  - Regions request that Institutions decommission Facilities
Vocabulary: Example

In library:
- Growth Region
- Corporation
- Reactor

In input file:
- Western Country
- Domestic
- AP-1000

In simulation:
- USA
- Southern Company
- Vogtle 3
Dynamic Resource Exchange

- DRE: Core algorithm for fuel cycle simulation
- Recomputed at each time step
- Solves economic problem dynamically
  - no hard-coded supply-demand behavior
- Enables complicated/creative fuel cycles
Dynamic Resource Exchange

Request for Bids
Queries each requesting Agent in the simulation that \textit{demands} a resource

Phase 1: Request for bids
Dynamic Resource Exchange

**Request for Bids**
Queries each requesting Agent in the simulation that *demands* a resource

**Response to Request for Bids**
Queries each responding Agent in the simulation that *supplies* a resource

Phase 1: Request for bids

Phase 2: Response to request for bids
Dynamic Resource Exchange

Preference Adjustment
Agent *reviews* all matches; opportunity for preference adjustment

Solution
Matches *selected* for satisfiable requests

Phase 3: Preference Adjustment

Phase 4: Market Resolution

\[ \max_{x,n} \sum a_{xn} \]
Dynamic Resource Exchange

**Request for Bids**
Queries each requesting Agent in the simulation that *demands* a resource

**Response to Request for Bids**
Queries each responding Agent in the simulation that *supplies* a resource

**Preference Adjustment**
Agent *reviews* all matches; opportunity for preference adjustment

**Solution**
Matches *selected* for satisfiable requests

---

Phase 1: Request for bids

Phase 2: Response to request for bids

Phase 3: Preference Adjustment

Phase 4: Market Resolution
Dynamic Resource Exchange

**Request for Bids**
Queries each requesting Agent in the simulation that *demands* a resource.

**Response to Request for Bids**
Queries each responding Agent in the simulation that *supplies* a resource.

**Preference Adjustment**
Agent *reviews* all matches; opportunity for preference adjustment.

**Solution**
Matches *selected* for satisfiable requests.

**Trade Execution**
Material *transaction* takes place.

---

Phase 1: Request for bids
Phase 2: Response to request for bids
Phase 3: Preference Adjustment
Phase 4: Market Resolution
## LP Formulation

### Variable | Description
--- | ---
$H, h$ | Commodities
$I, i$ | Bids
$J, j$ | Requests
$K, k$ | Capacities
$c$ | Cost of commodity
$x$ | Decision variable
$\beta$ | Capacity coefficient
$s$ | Supply capacity
$d$ | Demand capacity

$$
\begin{align*}
\min_x z &= \sum_{i \in I} \sum_{j \in J} \sum_{h \in H} c_{i,j}^h x_{i,j}^h \\
\text{s.t.} \quad & \sum_{j \in J} \sum_{h \in H} \beta_{i,k} x_{i,j}^h \leq s_{i,k} & \forall k \in K_l, \forall i \in I \\
& \sum_{i \in I} \sum_{h \in H} \beta_{j,k} x_{i,j}^h \geq d_{j,k} & \forall k \in K_j, \forall j \in J \\
& x_{i,j}^h \geq 0 & \forall x \in X
\end{align*}
$$
LP Supply Constraint: Example

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi$</td>
<td>Decision variable</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>Requested enrichment level</td>
</tr>
<tr>
<td>$s$</td>
<td>Supply capacity</td>
</tr>
<tr>
<td>$f$</td>
<td>Conversion function</td>
</tr>
</tbody>
</table>

\[
\sum_{j \in J} f_{SWU}(\varepsilon_j) x_{i,j}^{EU} \leq s_{i,SWU}
\]

\[
\sum_{j \in J} f_{NU}(\varepsilon_j) x_{i,j}^{EU} \leq s_{i,NU}
\]
LP Supply Constraint: General

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H, h$</td>
<td>Commodities</td>
</tr>
<tr>
<td>$I, i$</td>
<td>Bids</td>
</tr>
<tr>
<td>$J, j$</td>
<td>Requests</td>
</tr>
<tr>
<td>$K, k$</td>
<td>Capacities</td>
</tr>
<tr>
<td>$\beta_{i,k}(q^h_i)$</td>
<td>Conversion function</td>
</tr>
<tr>
<td>$x$</td>
<td>Decision variable</td>
</tr>
<tr>
<td>$s$</td>
<td>Supply capacity</td>
</tr>
</tbody>
</table>

These example constraints were a function of the isotopic profile of the request (quality, $q_j$). The conversion function of a supplier would be a function of the quality ($\beta_{i,k}(q^h_i)$):

$$\sum_{j \in J} \beta_{i,k}(q^h_j)x^h_{i,j} \leq s_{i,k} \quad \forall \ k \in K^h_i, \forall \ i \in I, \forall \ h \in H$$

Mixed integer linear program (MILP) can guarantee exclusive trades.
Cyclus Module Ecosystem

- Facility archetypes can be exchanged without changes to the kernel
- Example: increase reactor modeling fidelity
  - Low fidelity: fixed input/output recipes
  - Medium fidelity: lookup tables for output given input
  - High fidelity: burnup calculation based on given input
- Various distribution models are possible
Features of Module Ecosystem

- Archetype modules developed by independent teams
- Quality assessed by community
  - Tests and documentation provided by developers
  - Potential module users perform independent testing
- Diversity driven by use cases of developers
Cycamore: Standard Module Repository

Facilities
● Source
● Enrichment
● Fuel Fabrication
● Recipe Reactor
● Separations
● Storage
● Stream mixing
● Sink

Institutions
● Fixed Deployment
● Demand Response

Region
● Demand Growth
Ongoing Module Development

Bright-lite

- Given an initial composition, calculate a burnup
- Given a target burnup and two or more streams, determine blend

Nuclear Fuel Inventory Module

- Provide ORIGEN capability to Cyclus
- Multiple possible applications including reactor, separations, fuel fabrication
Ongoing Module Development

cyCLASS

- Wrap CLASS neural network methods for
  - Fuel fabrication
  - Depletion

Consortium for Verification Technology

- Facility archetypes with clandestine behavior
- Region/Institution archetypes that track multi-lateral relationships
Cyclus Analysis & Visualization

- Separate from simulation kernel
- Different tools for different purposes
  - Interactive data exploration
  - Automated generation of standardized images
  - Parameter sweeps
  - Wrappers for
    - Sensitivity study
    - Optimization
- Each tool uses state-of-the-art technology
- Open source development options
Cyclist Simulation Building

- Drag-and-drop interface
- Enables creative fuel cycle design
- Different modes for various user types
Cyclist Data Analysis Environment

- Explore data dynamically
- Visualization mode matched to combination of data type and user needs.
Cymetric Extensible Tool

- Extensible metrics design: Users can add new metrics derived from existing metrics
Cyclus Application:
Fuel Cycle Options Transition Analysis

- Began as participant in code comparison/benchmarking
- Now performing transition analysis in parallel with
  - VISION (INL)
  - DYMOND (ANL)
  - ORION (ORNL)
- Confirm that tools can perform transition analysis with necessary metrics

Flanagan - Wed - 14:00; Mouginot - Fri - 9:40
Cyclus Application:

**Deployment Optimization**

- Transition deployment optimization study
- Large-scale parallelization
- Disruption analysis research

<table>
<thead>
<tr>
<th>Facility</th>
<th>LWR</th>
<th>Repository</th>
<th>Fuel Fab</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>...</td>
</tr>
<tr>
<td>Trial 1</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>...</td>
</tr>
<tr>
<td>Trial 2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>...</td>
</tr>
</tbody>
</table>
Cyclus Application: Treaty Verification

Political Science, International Relations, Psychology
Behavior of Illicit Actors

Nuclear Engineering, Physics, Chemistry
Signals available to inspectors and observers

Nuclear Engineering, Chemical Engineering
Facility operations and mass flows

Applied Mathematics, Computer Science
Anomaly Detection Techniques
Cyclus Documentation

www.fuelcycle.org

- Introduction to Cyclus Fundamentals
- User guide
- Archetype developer guide
- Kernel developer guide
- Cyclus enhancement proposals
Cyclus Funding History

Heavily leveraging support from:

[Logos of NEUP, Argonne, NSF, U.S. NRC]
Potential Users

- DOE and DOE-funded group
  - NEUP funded universities

- Industry users, e.g. AREVA, EPRI

- Foreign DOE-equivalents, e.g. CEA, AECL

- Foreign universities, e.g. Cambridge University
Moving Forward with Cyclus

- Grow community of developers and users
- Expand portfolio of archetype modules
- Community Facilitator
- Demonstrate on community-relevant problems
Motivations for New Simulators

Current suite of simulators are difficult to benchmark:

- Commercial fuel cycle management
  - High-fidelity in-reactor simulation
  - Limited flexibility for novel systems/technologies

- Strategic decision making
  - Low-fidelity flow sheet approach
  - Complexity increases with need for detail
  - Limitations of software infrastructure
  - Low accessibility to non-technical audiences
Next Generation FCS Goals

- **Flexibility**
  - Model innovative/unconventional technologies
  - Minimal inherent technology assumptions

- **Modeling**
  - Discrete facilities with discrete material tracking
  - Optimization and sensitivity analysis

- **Software**
  - Low barrier to adoption with rapid payback
  - Commonly available software infrastructure
Cyclus is Flexible

- Individual facility modeling
  - Startup/shutdown
  - Disruptions
- Discrete material tracking at nuclide level
  - Effects of individual facility performance
  - Forensic tracking of material object ownership
- No inherent physics assumptions
  - Low fidelity, systems level models
  - High fidelity, facility level models
- Agent-based approach incorporates social/behavior models
Cyclus v1.0: Released May 30, 2014

Carlsen, Robert W.; Gidden, Matthew; Huff, Kathryn; Opotowsky, Arrielle C.; Rakhimov, Olzhas; Scopatz, Anthony M.; Welch, Zach; Wilson, Paul (2014): Cyclus v1.0.0. figshare. http://dx.doi.org/10.6084/m9.figshare.1041745
Linear Programming (LP) Background

Minimization or Maximization Objectives

\[ \min_x \quad z = c^T x \]
\[ \text{s.t.} \quad Ax \geq b \]
\[ x \geq 0 \]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c )</td>
<td>Cost vector</td>
</tr>
<tr>
<td>( x )</td>
<td>Decision variable</td>
</tr>
<tr>
<td>( A )</td>
<td>Constraint matrix</td>
</tr>
<tr>
<td>( b )</td>
<td>Threshold vector</td>
</tr>
</tbody>
</table>
Mixed Integer-Linear Programming (MILP)

Required to allow two groups of consumers:

1. those that require *exclusive* orders
2. those that allow *partial* orders

Introduce a **binary variable**, $y_{i,j}^h$:

- 1 if consumer $j$ is sent commodity $h$ by supplier $i$
- restrict number of resource flows to consumer $j$ to 1

\[
J = J_p \cup J_e
\]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H, h$</td>
<td>Commodities</td>
</tr>
<tr>
<td>$I, i$</td>
<td>Bids</td>
</tr>
<tr>
<td>$J, j$</td>
<td>Requests</td>
</tr>
<tr>
<td>$y_{i,j}^h$</td>
<td>Binary variable</td>
</tr>
</tbody>
</table>

\[
\sum_{h \in H_j} \sum_{i \in I} y_{i,j}^h = 1 \quad \forall j \in J_e
\]
MILP Supply Constraint: General

\[
\sum_{j \in J_p} \beta_{i,k}(q_j^h)x_{i,j}^h + \sum_{j \in J_e} \beta_{i,k}(q_j^h)y_{i,j}^h x_{j}^h \leq s_{i,k}^h \quad \forall i \in I, \forall k \in K_i^h, \forall h \in H
\]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H, h$</td>
<td>Commodities</td>
</tr>
<tr>
<td>$I, i$</td>
<td>Bids</td>
</tr>
<tr>
<td>$J, j$</td>
<td>Requests</td>
</tr>
<tr>
<td>$K, k$</td>
<td>Capacities</td>
</tr>
<tr>
<td>$\beta_{i,k}(q_i^h)$</td>
<td>Conversion function</td>
</tr>
<tr>
<td>$x, y$</td>
<td>Decision variable</td>
</tr>
<tr>
<td>$s$</td>
<td>Supply capacity</td>
</tr>
</tbody>
</table>
MILP Formulation

\[
\begin{align*}
\min_{x, y} & \quad z = \sum_{h \in H} \sum_{i \in I} \sum_{j \in J} c_{i,j}^h x_{i,j}^h + \sum_{h \in H} \sum_{i \in I} \sum_{j \in J} c_{i,j}^h y_{i,j}^h \bar{x}_{j}^h \\
\text{s.t.} & \quad \sum_{j \in J} \beta_{i,k} q_j^h x_{i,j}^h + \sum_{j \in J} \beta_{i,k} q_j^h y_{i,j}^h \bar{x}_{j}^h \leq s_{i,k}^h \\
& \quad \forall i \in I, \forall k \in K_i^h, \forall h \in H \\
& \quad \sum_{i \in I} \sum_{h \in H} \beta_{i,k} q_j^h x_{i,j}^h \geq d_j(h_j) \quad \forall k \in K_j, \forall j \in J_p \\
& \quad \sum_{i \in I} \sum_{h \in H} \beta_{i,k} q_j^h y_{i,j}^h \bar{x}_{j}^h \geq d_j(h_j) \quad \forall k \in K_j, \forall j \in J_e \\
& \quad \sum_{h \in H} y_{i,j}^h = 1 \quad \forall j \in J_e \\
& \quad x_{i,j}^h \geq 0 \quad \forall x \in X \\
& \quad y_{i,j}^h \in \{0, 1\} \quad \forall y \in Y
\end{align*}
\]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H, h$</td>
<td>Commodities</td>
</tr>
<tr>
<td>$I, i$</td>
<td>Bids</td>
</tr>
<tr>
<td>$J, j$</td>
<td>Requests</td>
</tr>
<tr>
<td>$K, k$</td>
<td>Capacities</td>
</tr>
<tr>
<td>$c$</td>
<td>Cost of commodity</td>
</tr>
<tr>
<td>$x, y$</td>
<td>Decision variable</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Capacity coefficient</td>
</tr>
<tr>
<td>$s$</td>
<td>Supply capacity</td>
</tr>
<tr>
<td>$d$</td>
<td>Demand capacity</td>
</tr>
</tbody>
</table>
Greedy Solver Algorithm

order request portfolios by average preference;
for all the request portfolios do
    order requests by average preference;
    matched ← 0;
    while matched ≤ qJ and ∃ a request do
        get next request;
        order arcs by preference;
        while matched ≤ qJ and ∃ an arc do
            get next arc;
            remaining ← qJ - matched;
            to_match ← min{remaining, Capacity(arc)};
            matched ← matched + to_match;
        end
    end
end