POLARIS: An Integrated Agent-Based Simulation Model of Activity Travel Behavior and Network Operations with Capability for Real-Time Transportation Simulation

Joshua Auld, Michael Hope
Hubert Ley, Vadim Sokolov, Bo Xu, Kuilin Zhang
Transportation Research and Analysis Computing Center
Argonne National Laboratory
April 27, 2014
What is POLARIS?

- Middleware for Developing Agent-based Models
  - Data Interchange
  - Visualization
  - Case Study Generation and Analysis
  - Discrete Event Simulation
  - Interprocess Communication

- A Repository of Transportation Libraries
  - Common Algorithms
  - Extended by Researchers
  - Standardized Style and Structure

- Fully Developed Applications
  - Transportation Network Simulation
  - Integrated Activity Based Simulation
Usage of the POLARIS Modeling Suite

Plug and play repository

Low-level Capabilities

Core Libraries
- Memory Allocator
- Common Design Patterns
- Discrete Event Engine
- Custom Data Containers
- Interprocess Communication

Libraries
- Population Synthesizer
- Demand Simulation
- Traffic Manager
- Antares Graphics
- Input-Output
- Network Simulation
- Event Manager
- Graphics Driver
- Geospatial Database
- Router
- ITS
- GUI Toolkit

User Libraries
- Modified Functionality
- New functionality
- ...
POLARIS Core: Re-Usable Low Level Capabilities

- Discrete Event Engine
  - Enables writing from an agent-based perspective

- Memory Management Library
  - Optimized for the type of allocation needed in transportation modeling applications

- High Performance Data Structures
  - Non-standard structures relevant for use in transportation modeling applications

- Interprocess Engine
  - Enables parallel cluster execution

- Library of Common Design Patterns
  - Generic programming toolbox and structural guidelines for extensible development
POLARIS Core: Memory Manager Overview

- Motivation
  - Transportation code can be performance critical
  - Simulation code tends to follow distinct memory allocation / deallocation patterns
  - Time Step Scheduling execution requires a global tracking of allocated objects

- Technique
  - Hierarchical memory layout: divide by type, then block, then object
  - Memory blocks owned by threads for allocation
  - Effective structure for each type is an unrolled linked list of memory pools
  - Block size is optimized by user input, object count, and object size
  - Replace the built-in memory allocator with tc_malloc

- Performance Benefits
  - Allocations and deallocations are fully parallelized
  - Unrolled linked list structure provides balance of stride optimization vs memory alteration
  - Can make use of user input to further optimize the structure

- Developer Benefits
  - Global tracking of memory allows global tracking of objects by ID
  - Tracking of memory usage without an external tool
Polaris Memory Allocator - Implementation Details

Interaction with General Pool

Type Specialized Pool A

Type Object Width

Type Specialized Pool A

Type Execution / Memory Information

Page Execution / Memory Information

User Code

Type_A* object = Allocate<Type A>()
POLARIS Core: Discrete Event Engine Overview

- Allows the developer to:
  - Create an agent of any kind (Traveler, Traffic manager, Transit authority, etc...)
  - Describe how and when it wants to act
  - Define what it does when it does act
  - Define one or more actions which the agent can choose among
  - Create autonomously executing agents

- Structural Benefits
  - Multiple base agent types for different execution patterns
  - Eliminates the traditional execution loop over time and objects
  - Provides universal “time” in the simulation
  - Eases the task of coordinating the actions of agents
  - Allows user to write from the agents’ point of view

- Performance Benefits
  - Enables automated multi-threading which is highly scalable
  - System can self-optimize to balance workload among threads
  - Hierarchical structure allows skipping the polling of large numbers of agents
Discrete Event Engine Hierarchical Scheduling

If Root To Exec
World -> Root

If Type To Exec
Root -> Type

If Block To Exec
Type -> Block

If Object To Exec
Block -> Object

Object Next Exec -> Block Next Exec
Block Next Exec -> Type Next Exec
Type Next Exec -> Root Next Exec

World
Root
Type
Block
Object
POLARIS Common Design Patterns

- The POLARIS style conventions are generally encapsulated in a series of custom keywords and macros which simplify development of POLARIS compliant objects
  - Compile time concept check convey information to developers & function overloading
  - Alternative to virtual / abstract base classes
  - Higher performance while still providing abstraction: create “views” on objects for different use cases

- POLARIS Component: Fundamental POLARIS Type
  - Connects object to memory allocator, interprocess engine, and discrete event engine

- POLARIS Prototype: Extremely Abstract Definition of Type
  - For example: vehicle rather than car, bus, truck

- POLARIS Implementations: Concrete Definition of Type
  - For example: car, bus, truck rather than vehicle

- POLARIS Variables: Basic Types with Relevant Semantic Information
  - Think “feet in meters” instead of “float”
Antares Graphical Library

- Visualize in 3D or Plot in 2D
- Agent drives the visualization
- General Purpose Drawing Not Transportation-Specific

- Image Texturing
- Identification Capabilities
- Online Interactivity with Underlying Simulation
- High Performance
Data Interchange System

- Lightweight no maintenance SQLite engine

Design:

- Spatial queries are handled by spatiality extension

- POLARIS project was granted permission to use ODB library ([http://www.codesynthesis.com/products/odb/](http://www.codesynthesis.com/products/odb/)) under BSD license. It is GPL otherwise.
 Scenario Configuration

- Scenario configuration
  - Semantic-rich for validation of scenario configuration
  - Human readable
  - Parameterized scenarios: probability distributions

- Parallel execution
  - A separate job is generated for each scenario
  - Submission to cluster
  - Report generation
POLARIS Activity-Based Simulation Model
POLARIS: Agent Based Transportation Simulation System

- Activity Generation
- Activity Scheduling
- Route Choice
- Activity Planning

Person

Traveler Movements

- ITS Infrastructure
- Intersection Simulation
- Link Simulation

Network

- Network Monitoring
- Information Dissemination
- ITS Response Strategies

ITS Responses

Traffic Management Center
POLARIS ABM Model Components

- **Agent Initialization**
  - Population synthesis creates households / persons
  - Home, workplace and school location choice
  - Network / ITS agents read from system database

- **ABM/Demand components**
  - Generation, Planning and scheduling of activities to satisfy needs

- **Network assignment / simulation components**
  - Individualized route selection (multi-modal)
  - Simulation of travel on transportation network (auto only)

- **ITS infrastructure and management components**
  - Simulate ITS infrastructure operation
  - Management strategies (directly input or automated)

- **Visualization and agent-interaction**
  - View model through POLARIS GUI
  - Modify agent / infrastructure states
POLARIS Demand Model: Introduction

- Based on ADAPTS activity-based model:
  - Simulation of how activities are planned and scheduled
  - Extends concept of “planning horizon” to activity attributes
  - Time-of-day, location, mode, party composition

- Modified version of ADAPTS implemented in POLARIS language
  - Adapted to implement models as individual behaviors
  - Combined with route choice and traffic simulation

- Core concept:
  - Set of activity planning / scheduling processes represented by heuristics or models
  - Behavior is fundamental, outcomes are constrained by local context
  - Dynamic state dependence in decision making
Activity-Based Travel Demand Model Framework
Simulation-Based Dynamic Traffic Assignment Model

- Route Choice Mode
  - Individual, prevailing/historical conditions A* routing
  - Customizable, agent-specific multi-modal routing behaviors
  - Based on high-performance, generalized graph structure

- En-route Switching Model
  - Bounded rationality model, triggered by information or excessive delay

- Traffic Control Model

- Mesoscopic Traffic Simulation Model
  - Newell’s kinematic wave model with back-wave propagation
  - Simulated controls blocks link transfer during red time
Simulation-Based Dynamic Traffic Assignment Model

Routing Agent
- Network Topology
- Route Generation Model
- Router
- Network / ITS Information
  - Route Decisions
  - Routes
- Person Planner
- Activity Plan
- Person Mover
- En-route switching Model

Person Agent
- Traveler Characteristics

Link Agent
- Link Simulation Model
- ITS model
- Traffic Events
- Arrival States
- Intersection Simulation Model
- Departure States
- Network Performance
- Signal State
- Traffic Operations and Control Model
Agent-Based Intelligent Transportation System Model

![Diagram of Agent-Based Intelligent Transportation System Model]

- **Data source**
- **Model**
- **Model result**

**Traffic Simulation Model**

- **Network Performance**
  - **Sensor Model**
    - **Sensor Readings**
      - **System Manager Agent**
        - **New Events**
          - **Event Manager**
            - **ITS Infrastructure**
            - **Network Topology**
            - **Network Event**

- **Route Decisions**
  - **Route Choice Model**
    - **Management Strategy**
      - **Activity Planner**
        - **Person Agent**

**Routing Agent**
Polaris ABM - Chicago Region Model

- Model implemented for Chicago region
  - Demand components estimated from CMAP and UTRACS data
  - Network based on CMAP planning network, with extensive modification
  - Synthesized intersection control from TRANSIMS

- Model characteristics:
  - 31,000 links with 18,900 nodes
  - 171,000 individual activity locations (based on land-use polygons)
  - 10.2 million persons in 3.8 million HH
  - 32.8 million trips (27 million by auto)

- Performance:
  - 32-core machine with 64GB RAM
  - ~70 min. run-time on 28 threads
  - 46GB maximum memory utilization
Model Validation - Chicago Model

Vehicles In-network Comparison

- Vehicles in network: SIMULATED
- Vehicles in network: CMAP Survey moving avg.
Case Study

- Chicago CBD and surrounding area:
  - 112 Traffic analysis zones
  - 8037 activity locations
  - 5280 links, 3005 nodes
  - 213k HH, 403k people

- ITS infrastructure available

- All are configurable from the GUI as the model runs: human-in-the-loop

- Scenario:
  - A snowstorm on southeastern portion of study area
  - Accident: I-290 east of I-94 at 6AM
  - Accident: LSD south of Jackson at 6AM
ITS Demo

- Accident on Interstate 290, with and without VMS warnings on other interstates
The use of ITS saves 3,600 hours of delay which is a reduction of 7% of the excess delay caused by the network events.
Conclusion

- POLARIS agent-based framework used to create a fully integrated, agent-based simulation of travel demand, network supply and ITS operations
- Implemented for Chicago region
  - Model being constantly updated, validation work continues
  - Code base is extensible and can be added-to or adapted by others
  - High-performance allows for many interesting analyses
- ITS case-study
  - Demonstration of benefits of ITS; models are still highly preliminary
  - Test both automated management routines and “human-in-the-loop”
- Upcoming applications:
  - Statewide travel demand model
  - Freight modeling
  - Emergency evacuation

For more information and code go to: https://github.com/anl-tracc/polaris
Thank You!
POLARIS

(Planning and Operations Language for Agent-based Regional Integrated Simulation)

- Mandates from FHWA:
  1. Model Traffic Control Centers and other ITS Systems
  2. Enhance Interoperability among Existing Tools

- Core Goals and Philosophies of the POLARIS Effort:
  - Develop Transportation Modeling Standards and Protocols
  - Create an Open Source Model Development Environment
  - Seek Out Opinions from and Listen to the Transportation Community
  - Connect Sub-Communities with a Common Modeling Framework
  - Offer Helpful Tools while Maintaining Flexibility and Modularity
External Tool Usage in POLARIS

- **Languages**
  - C++
  - Python

- **General Purpose Libraries**
  - Boost
  - Loki

- **Visual Technologies**
  - OpenGL
  - WxWidgets / WxPython
  - Plplot

- **Interprocess Technologies**
  - Sockets

- **Data Technologies**
  - SQLite Database
  - ODB
  - Spatialite

- **Build Environment**
  - Cmake
  - Microsoft Visual Studio

- **Source Control and Documentation**
  - Doxygen
  - SVN

- **Code Analysis**
  - Google Testing Framework
  - Intel Vtune Profiling
  - Valgrind
Agent Based Modeling

- Agent will encapsulate a set of behaviors that govern their interactions with other agents and with their environment.

- Agent-based methodologies have proven to provide a structure that can be used to model a vast array of phenomena:
  - social processes
  - software systems
  - manufacturing systems
  - urban dynamics
  - economics
POLARIS Common Design Patterns

- **C++ Concepts:**
  - Compile time checks
  - Conveys information to other developers
  - Programmatic function overloading

- **Prototype / Implementation Pattern:**
  - Alternative to virtual / abstract base classes
  - Higher performance than abstract base classes while still providing abstraction (though of course not quite as much)
  - Can be combined with abstract base classes
  - Good at creating “views” on objects for different use cases
POLARIS Common Design Patterns

- The POLARIS style conventions are generally encapsulated in a series of custom keywords and macros which simplify the development of POLARIS style compliant objects

- POLARIS Component: Fundamental POLARIS Type
  - Connects object to memory allocator, interprocess engine, and discrete event engine

- POLARIS Prototype: Extremely Abstract Definition of Type
  - For example: vehicle rather than car, bus, truck

- POLARIS Implementations: Concrete Definition of Type
  - For example: car, bus, truck rather than vehicle

- POLARIS Variables: Basic Types with Relevant Semantic Information
  - Think “feet in meters” instead of “float”
One-to-One Translation of a C++ Class to a POLARIS Component

```cpp
class Link {
    public:
    float get_length() {
        return length;
    }
    void set_length(float value) {
        length = value;
    }
    void print() {
        printf("this->length\n");
    }

    private:
    float length;
}

prototype struct Link {
    feature_accessor(length, is_integral, no_requirements);
    feature void print() {
        printf("\n");
    }
}

implementation struct Link_Impl : public Polaris_Component {...} {
    member_data(float, length, is_integral, no_requirements);
};
```
POLARIS Antares: Antares Benefits

- **Usability**
  - Based on high performance OpenGL
  - Abstracts the low level so the user need not know anything about OpenGL
  - Provides automated memory management
  - 3D object picking capability performed automatically
  - Organized by layers

- **Interactivity**
  - Designed to execute in tandem with a running simulation
  - User can define what happens when objects are clicked on
    - Highlight objects
    - Change visual appearance
    - Pop up dialog
    - Change simulation parameters
Scenario Configuration

- Scenario configuration
  - Semantic-rich for validation of scenario configuration
  - Human readable
  - Parameterized scenarios: probability distributions

- Parallel execution
  - A separate job is generated for each scenario
  - Submission to cluster
  - Report generation
Network Editor

- General Purpose Network Editor
- WxPython
- OpenGL
- Spatialite
- Sqlite Database
- Intersection Editor
- Layer-Based Drawing
- Label Capabilities
- Selection and Identification Capabilities
- Rule-based Editing
Activity Generation

- Activity generation with joint hazard-duration equations
  - Significant socio-economic variables
  - Impact of hazard rate from other activities

- Failure probability (generation) calculated each timestep
  - Based on time-since-last activity
  - Calculated using observed UTRACS and fit to CMAP survey through updating
  - \( h_0 = \lambda_E \gamma_E (\lambda_E t)^{\gamma_E - 1} + \lambda_L \gamma_L (\lambda_L t)^{\gamma_L - 1} \): where \( \gamma_L > 1 \) and \( \gamma_E < 1 \)
  - decreasing early failure (after trip chain), increasing late failure due to need growth over time

\[
\begin{align*}
  h_{i \text{wev}}^{\text{wev}} (t_i, x_i, h_j^{\text{nev}}, h_k^{\text{nev}}, ...) &= h_0^i e^{-(\beta_i x_i + \beta_{ij} h_j^{\text{nev}} + \beta_{ik} h_k^{\text{nev}} + ...)} \\
  h_{j \text{wev}}^{\text{wev}} (t_j, x_j, h_i^{\text{nev}}, h_k^{\text{nev}}, ...) &= h_0^j e^{-(\beta_k x_k + \beta_{ik} h_i^{\text{nev}} + \beta_{jk} h_j^{\text{nev}} + ...)} \\
  h_{k \text{wev}}^{\text{wev}} (t_k, x_k, h_i^{\text{nev}}, h_j^{\text{nev}}, ...) &= h_0^k e^{-(\beta_k x_k + \beta_{ik} h_i^{\text{nev}} + \beta_{jk} h_j^{\text{nev}} + ...)} \\
  \vdots
\end{align*}
\]

\( h^{\text{wev}} \) = hazard with exogenous hazard covariates

\( h^{\text{nev}} \) = without exogenous covariates
Results for eat-out activity

- Baseline hazard shows:
  - Slight chance of chaining
  - Increasing hazard over time
- Large households eat out less
- Wealthy eat out more, etc.
- Religious/leisure activities delay eating out
- Social activities increase hazard

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>t-stat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma_L</td>
<td>1.13</td>
<td>13.8</td>
</tr>
<tr>
<td>Gamma_E</td>
<td>0.22</td>
<td>8.0</td>
</tr>
<tr>
<td>Const_L</td>
<td>0.74</td>
<td>1.8</td>
</tr>
<tr>
<td>Const_E</td>
<td>0.60</td>
<td>1.4</td>
</tr>
<tr>
<td>INC_HI</td>
<td>-0.74</td>
<td>-4.7</td>
</tr>
<tr>
<td>HHSIZE</td>
<td>0.36</td>
<td>2.7</td>
</tr>
<tr>
<td>NSTUDENT</td>
<td>-0.48</td>
<td>-3.0</td>
</tr>
<tr>
<td>MALE</td>
<td>-0.66</td>
<td>-3.9</td>
</tr>
</tbody>
</table>

**Endogenous Factors**

<table>
<thead>
<tr>
<th>Factor</th>
<th>Estimate</th>
<th>t-stat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Religious on Eat-out</td>
<td>0.61</td>
<td>1.7</td>
</tr>
<tr>
<td>Leisure on Eat-out</td>
<td>5.01</td>
<td>2.4</td>
</tr>
<tr>
<td>Social on Eat-out</td>
<td>-0.32</td>
<td>-1.8</td>
</tr>
</tbody>
</table>
Activity Generation: Validation

- Application to Chicago-region
  - Calibrated to 2007 data
  - Backcast validation to 1990 HHTS
  - Validated by activity-type, HH Type, etc.
Planning Order Model Framework

- Assign plan horizon to each attribute
  - After activity generated
- Plan order model process
  - Assigns attribute flexibility
  - Get activity plan horizon
  - Attribute plan horizons
- Plan horizons for each attribute based on:
  - Attribute flexibilities
  - Activity plan horizon
  - General activity attributes
  - Socio-demographics, etc.
- Defines the *meta-attributes* of the activity attributes
Planning Order Model Specification

- Planning order model framework has three linked models
  - Activity attribute flexibility, Activity plan horizon, Attribute plan horizons
  - Linked together through lagged variables
  - No specific attributes of activities known

- Required model characteristics
  - Multiple responses for flexibility and attribute plan horizons
  - Repeated unit of observation (responses from same activity)
  - Ordinal responses for plan horizon models
  - Binary response for flexibility (flexible vs. not flexible)

- Use of a variety of probit models
  - Multivariate ordinal probit (MVOP) models for flexibility and plan horizon
  - Allows error correlation – multivariate normal error terms

\[ P(Y_1 = k_1, Y_2 = k_2, \ldots Y_J = k_J) = \int_{x_1, x_2, \ldots} \phi(x_1, x_2, \ldots) dx_1 dx_2 \ldots dx_J \]
## Attribute Planning Results

- Follows overall activity plan horizon
- ICT use:
  - Increases mode preplan
  - Increase work location preplan
  - Decrease preplanning of discretionary/shop location
- Inflexible attributes increase preplanning
- Teleworkers have more preplanned work
- Long acts more preplanned, except for discretionary

### Individual
- Frequent ICT usage

### Activity
- Actplan - Impulsive
- Actplan - Same day
- Actplan - Same week
- Inflexible Start Time
- Inflexible Duration
- Avg Duration (in days)
- Avg Daily Frequency

### Activity Type
- Act1 (Work/School)
- Act2 (Personal)
- Act3 (Maintenance)
- Act4 (Discretionary)
- Act5 (Shopping)

### Limits
- $\alpha_2$
- $\alpha_3$
- $\alpha_4$

### Correlation Coefficients

### Table

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mode</th>
<th>WHO-WITH</th>
<th>LOCATION</th>
<th>START</th>
<th>DURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\beta$</td>
<td>t-stat</td>
<td>$\beta$</td>
<td>t-stat</td>
<td>$\beta$</td>
</tr>
<tr>
<td>Constant</td>
<td>0.618</td>
<td>3.09</td>
<td>4.096</td>
<td>22.37</td>
<td>2.079</td>
</tr>
<tr>
<td>Employed</td>
<td>0.634</td>
<td>5.83</td>
<td>--</td>
<td>--</td>
<td>0.203</td>
</tr>
<tr>
<td>Student</td>
<td>1.044</td>
<td>9.24</td>
<td>--</td>
<td>--</td>
<td>0.351</td>
</tr>
<tr>
<td>Senior</td>
<td>-0.115</td>
<td>-1.61</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Male</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Frequent ICT usage</td>
<td>0.130</td>
<td>1.60</td>
<td>--</td>
<td>--</td>
<td>0.840</td>
</tr>
<tr>
<td>Actplan - Same day</td>
<td>-0.492</td>
<td>-3.36</td>
<td>-2.912</td>
<td>-16.85</td>
<td>-1.401</td>
</tr>
<tr>
<td>Actplan - Same week</td>
<td>-0.272</td>
<td>-1.74</td>
<td>-1.231</td>
<td>-7.13</td>
<td>-0.582</td>
</tr>
<tr>
<td>Inflexible Start Time</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Inflexible Duration</td>
<td>0.106</td>
<td>1.55</td>
<td>--</td>
<td>--</td>
<td>0.163</td>
</tr>
<tr>
<td>Avg Duration (in days)</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>1.846</td>
</tr>
<tr>
<td>Avg Daily Frequency</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

### Model

### Table

<table>
<thead>
<tr>
<th>Variable</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
<th>$\alpha_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\beta$</td>
<td>t-stat</td>
<td>$\beta$</td>
</tr>
<tr>
<td>Constant</td>
<td>0.835</td>
<td>17.43</td>
<td>2.029</td>
</tr>
<tr>
<td>Actplan</td>
<td>1.486</td>
<td>26.87</td>
<td>3.614</td>
</tr>
<tr>
<td>Actplan</td>
<td>1.794</td>
<td>30.76</td>
<td>4.402</td>
</tr>
</tbody>
</table>

### Correlation Coefficients

<table>
<thead>
<tr>
<th>Mode</th>
<th>$\rho$</th>
<th>t-stat</th>
<th>Mode</th>
<th>$\rho$</th>
<th>t-stat</th>
<th>Mode</th>
<th>$\rho$</th>
<th>t-stat</th>
<th>Mode</th>
<th>$\rho$</th>
<th>t-stat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>1</td>
<td>--</td>
<td>WHO-WITH</td>
<td>1</td>
<td>--</td>
<td>Location</td>
<td>0.143</td>
<td>3.8</td>
<td>Start time</td>
<td>0.159</td>
<td>4.1</td>
</tr>
<tr>
<td>WHO-WITH</td>
<td>0.148</td>
<td>4.2</td>
<td>0.104</td>
<td>2.2</td>
<td>0.187</td>
<td>4.5</td>
<td>0.178</td>
<td>3.2</td>
<td>0.218</td>
<td>5.7</td>
<td>0.539</td>
</tr>
<tr>
<td>Location</td>
<td>0.104</td>
<td>2.2</td>
<td>0.181</td>
<td>3.4</td>
<td>0.187</td>
<td>4.5</td>
<td>0.178</td>
<td>3.2</td>
<td>0.218</td>
<td>5.7</td>
<td>0.539</td>
</tr>
<tr>
<td>Start time</td>
<td>0.181</td>
<td>3.4</td>
<td>0.187</td>
<td>4.5</td>
<td>0.178</td>
<td>3.2</td>
<td>0.218</td>
<td>5.7</td>
<td>0.539</td>
<td>19.2</td>
<td>1</td>
</tr>
</tbody>
</table>
### Activity Scheduling - Overall System

- Derived from TASHA (ILUTE) scheduling system
  - Activities added to schedule as simulation progresses
  - Overlaps happen due to planning process or schedule delay
  - Incorporates conflict resolution modeling
  - Resolution strategy determines which rules to follow
  - Resolution strategies and modification rules from observed empirical data

- When a new activity is added:
  1. Determines conflict type (shown below)
  2. Run conflict resolution model to determine resolution type
  3. Modify schedule to fit new activity based on resolution type
  4. Insert new activity or drop it (and restore original schedule)

<table>
<thead>
<tr>
<th>Case 1: Inserted Original</th>
<th>Case 2: Overlapped Original</th>
<th>Case 3: Overlap Start</th>
<th>Case 4: Overlap End</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 5: Overlap End &amp; Start</th>
<th>Case 6: Insert &amp; Overlap Start</th>
<th>Case 7: Overlap End &amp; Insert</th>
<th>Case 8: Insert/Overlap Start /End</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Activity Scheduling Rules

- In the scheduling rules, the situation would be handled as follows:
  1. If resolution type is ‘Delete Original’
     1. Remove Activity B from schedule, add Activity A
  2. If resolution type is ‘Modify Original’
     1. Move Activity B, align start of Activity B with end of Activity A
     2. Truncate Activity B
     3. Insertion is not feasible
  3. If resolution type is ‘Modify Conflicting’
     1. Move Activity A, align end of Activity A with start of Activity B
     2. Truncate Activity A
     3. Insertion is not feasible
  4. If resolution type is ‘Modify Both’
     1. Move Activity A, align end of Activity A with start of Activity B
     2. Move Activity B backward
     3. Truncate Activity A and Activity B proportional to durations;
     4. Insertion is not feasible.
## Conflict Resolution Models: Decision tree results

### Decision Rule Variables for Out-of-Home Activities

<table>
<thead>
<tr>
<th>APLAN</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>4</th>
<th>4</th>
<th>999</th>
<th>999</th>
<th>999</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERLAP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ADUR</td>
<td>-</td>
<td>&lt;15</td>
<td>15-45</td>
<td>45-60</td>
<td>60-299</td>
<td>60-299</td>
<td>&gt;299</td>
<td>&gt;299</td>
<td>&gt;299</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CONTYPE</td>
<td>-</td>
<td>-</td>
<td>&lt;15</td>
<td>15-45</td>
<td>15-45</td>
<td>45-60</td>
<td>60-299</td>
<td>60-299</td>
<td>&gt;299</td>
<td>&gt;299</td>
<td>&gt;299</td>
<td>3.2</td>
<td>1.4</td>
</tr>
<tr>
<td>TAVAIL</td>
<td>-</td>
<td>-</td>
<td>&lt;11</td>
<td>11</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>&lt;30</td>
<td>&gt;30</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B_LOC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Resolution Strategy

<table>
<thead>
<tr>
<th>Resolution Strategy</th>
<th>Mod. Orig.</th>
<th>Mod. Conf.</th>
<th>Mod. Both</th>
<th>Delete. Orig.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>38%</td>
<td>88%</td>
<td>5%</td>
<td>6%</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>8%</td>
<td>86%</td>
<td>3%</td>
</tr>
<tr>
<td></td>
<td>14%</td>
<td>4%</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>19%</td>
<td>0%</td>
<td>5%</td>
<td>91%</td>
</tr>
</tbody>
</table>

### Decision Rule Variables for In-Home Activities

<table>
<thead>
<tr>
<th>A_TRAVEL</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_TRAVEL</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CONTYPE</td>
<td>-</td>
<td>1</td>
<td>2,4</td>
<td>3</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SENIOR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>A_PER</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2,3</td>
<td>2,3</td>
<td>2,3</td>
<td></td>
</tr>
<tr>
<td>B_DUR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>&lt;170</td>
<td>&gt;170</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B_PLAN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3,999</td>
<td>1,2,4</td>
<td>1,2,4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OVERLAP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>&lt;98%</td>
<td>&gt;98%</td>
</tr>
</tbody>
</table>

### Resolution Strategy

<table>
<thead>
<tr>
<th>Resolution Strategy</th>
<th>Mod. Orig.</th>
<th>Mod. Conf.</th>
<th>Mod. Both</th>
<th>Delete. Orig.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
<td>66%</td>
<td>81%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>34%</td>
<td>19%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Planning Constrained Choice Set Formation

- Choose destination location from set of zones
  - Limit based on planned activities
  - Generates “Available Set” depending on Activity Plan Horizons
  - Requires plan horizon model to specify when activities planned

- Use *Stratified Importance Sampling* on “Available Set”
  - Travel time (<45 min, >45 min)
  - Total employment (<500, >500)
Planning Constrained Destination Choice Model

- **Planning Constraint applied in choice set formation**

- Use variety of Competing-Destinations model:
  - Accounts for systematic (observable) spatial dependencies
  - Accessibility of zone to similar surrounding zones
  - Directly solve for distance decay parameter

- **Correct for sampling bias**
  - Introduced by importance sampling procedure
  - Reduces simulation variance - always have reasonable zones to choose

- **Also create version with no planning constraints for comparison**
  - Constraints only from fixed (mandatory) activities

- **Data source: Chicago Travel Tracker survey**
  - Over 10,000 individuals in 1 or 2 day activity-travel survey

- **Estimated for 7 discretionary activity types (i.e. shop, social, eat out...)**
Destination Choice Model Specification

\[ V_{in} = \beta_T T_{in} + \beta_I I_{in} + \beta_R R_{in} + \gamma \ln \left( \sum_j \beta_j A_{ij} + \sum_k \beta_k E_{ik} \right) + \sum_k \theta_k C_k + \ln \left[ \frac{1}{p(i)} \right] \]

Where,
- \( \beta_T \) = travel time parameter
- \( T_{in} \) = travel time to zone \( i \) from home location of decision-maker \( n \)
- \( \beta_I \) = income difference parameter
- \( I_{in} \) = absolute value of average zonal income for \( i \) minus income for decision-maker \( n \)
- \( \beta_R \) = race difference parameter
- \( R_{in} \) = 1 - \( R_i \), where \( R_i \) is the percentage of residents of zone \( i \) of a different race than decision-maker \( n \)
- \( \gamma \) = logsum parameter for zonal size variables
- \( \beta_j \) = parameter for the \( j=1\ldots J \), land use variables
- \( A_{ij} \) = values of the \( j=1\ldots J \), land use area variables for zone \( i \)
- \( \beta_k \) = parameter for the \( k=1\ldots K \), employment sector variables
- \( E_{ik} \) = values of the \( k=1\ldots K \), employment sector variables for zone \( i \)
- \( \theta_k \) = competition/clustering parameter for employment variable \( k \)
- \( C_k \) = Competition/Agglomeration factor, see Equation 19
- \( p(i) \) = probability of selecting zone \( i \) into the current choice set, from Equation 2

\[ C_k = \left( \frac{1}{N_z - 1} \sum_{l \neq i}^{N_z} E_{lk} e^{\gamma t_{il}} \right) \]

Where,
- \( N_z \) = number of zones in region
- \( t_{il} \) = distance between zone \( i \) and another zone \( l \)
- \( \gamma \) = distance decay parameter