Atlanta Regional Commission (ARC) ActivitySim Implementation Update

Introduction

- **ARC 21-county modeling domain:**
  - 6,400 sq. mi.
  - 6,000,000+ population
  - ARC is 1 of the 3 Founding Agencies of ActivitySim with MTC and SANDAG Going Back Over a Decade Ago

- **Acknowledgements:**
  - Sijia Wang and the WSP team
  - Jeff Newman, Driftless Lab

History of ActivitySim at ARC

- **Why ActivitySim?: Replace ARC's CT-RAMP Java-Based ABM Software with ActivitySim Python-Based Software**
- **2019: ARC's first ActivitySim v0.9.7 Implementation**
  - Parking location choice sub-model
  - Logit-based trip scheduling choice sub-model
  - With legacy "chunking" training: 15+ hours to simulate all travel choices
- **Lessons learned:**
  - Runnable but not practical as a production model due to:
    - Large memory requirements
    - Long model run times

From ActivitySim v0.9.7 to v1.3.0

- **Goal: Implement 50% Runtime and Memory Relief Saving**
- **Sharrow (yes it works):**
  - open-source Python package
  - extension of Numba
  - offers data formatting and just-in-time compiler to convert ActivitySim spec files into optimized functions
  - Compile ActivitySim UECs only once, re-use many times
- **Data type optimization| reduce memory, speed up process**
  - Convert string data into “pandas” categorical data
  - Conversion for string columns in most chooser table

From v0.9.7 to v1.3.0 (continued)

- **Explicit chunking:**
  - No training to find optimal number of choice makers for each sub-model
  - Allows users to specify either:
    - An integer valuer as the number of choice makers, or
    - A fractional value as the proportion of choice makers to simulate for each sub-model in each chunk
  - Provides greater stability in ARC's ActivitySim model implementation
  - Removes the additional overhead required to train the legacy chunking
  - Explicit chunking implemented in ARC's ActivitySim implementation for memory intensive sub-models: Workplace Location, School Location, Mandatory Tour Scheduling, Non-Mandatory Tour Scheduling, Trip Destination, Parking Location

ARC's ActivitySim v1.3.0 model

- **ARC's v1.3.0 ActivitySim model implementation testing:**
  - Set up with GitHub Actions to run Continuous Integration testing
  - Model runtime performance analysis using Fulton County sub-model:
    - Multiprocessing without Sharrow
    - Multiprocessing with Sharrow
  - ARC ran into the "Law of Diminishing Returns":
    - Runtime savings with more processors diminished
    - Diminishing returns when increasing the number of processors
    - More testing needed to discover the "sweet spot" (optimal efficiency)

  - atlregional/arc-activitysim: ARC ActivitySim implementation
  - https://github.com/atlregional/arc-activitysim

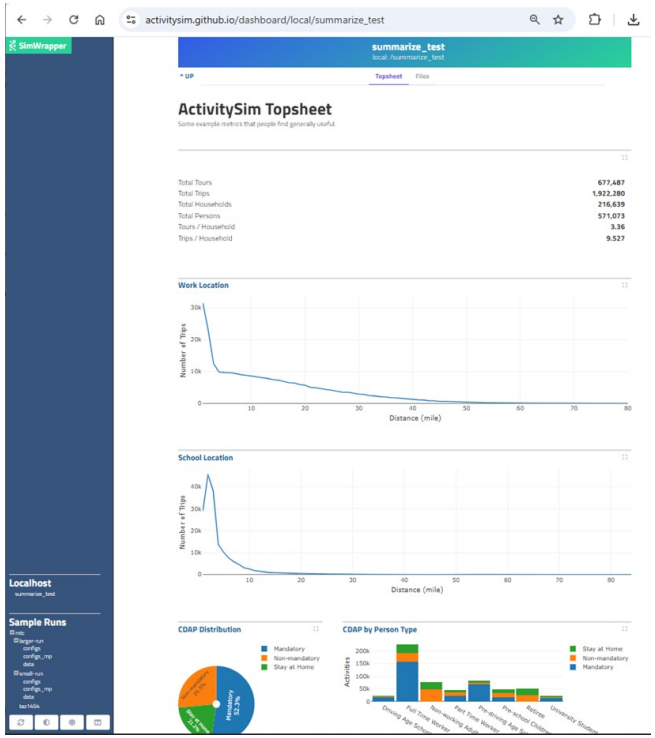
ARC's ActivitySim Implementation Full Scale Model Runtime Testing

- **Single Processor, Large (100,000 households) with Skims and ARC Land Use Data**
  - Sharrow Enabled: 26.8 minutes
  - Sharrow Disabled: 64.5 minutes

- **Multiprocessing, Full Scale, 100% Sample of Households**
  - Sharrow Enabled: 113 minutes
  - Sharrow Disables: 200 minutes

- **Running the Model with Sharrow Enabled Offers ARC Substantial Runtime Benefits with no Meaningful Change in Simulation Results**

ARC's ActivitySim Model Results Visualization with SimWrapper



ActivitySim at ARC: The Future…

- ARC's ActivitySim model is one step closer to becoming ARC's next production model in late 2026 or early 2027
- Update ARC's ActivitySim specs to reflect recent updates to ARC's CT-RAMP model.
- Update ARC's ActivitySim implementation 2015 baseline
- ARC's CT-RAMP model baseline: Pre-Pandemic (2019-20)
- ARC's household travel survey in 2026 (last one in 2011)
- Update ARC's ActivitySim model to the latest and greatest ActivitySim version

Thank You! Any Questions?

Guy Rousseau
Modeling Manager / Data Administrator
Atlanta Regional Commission, Atlanta, Georgia
grousseau@atlantaregional.org