Network Analyst: Solving Large Transportation Analysis Problems
Deelesh Mandloi & Melinda Morang
Agenda

• Intro/background
• Techniques for solving large problems
• Python script walk-through (solving locally)
• Working with services
• Python script walk-through (solving with a service)
Code and slides:
http://esriurl.com/uc18pyslnp
Introduction
ArcGIS Network Analyst Extension for transportation analysis

Coverage
- Service Area

Optimization
- Location-Allocation
- Vehicle Routing Problem

Point-to-point routing
- Route
- Closest Facility
- Origin-Destination Cost Matrix
OD Cost Matrix

Calculates the **travel time** or **distance** between a **set of origins** and a **set of destinations**.

- Can use an impedance limit
- Can limit the number of destinations to find
What is a large problem?

• Can’t be solved in one calculation
  - Unreasonable calculation time
  - Memory limits
  - Service limits

• Large number of inputs

• Large number of outputs
  - number of origins x number of destinations
  - $1000 \times 1000 = 1,000,000$

Examples

• Calculate drive time for all patients to all medical clinics within 100 miles
• Calculate the network distance from every parcel to every other parcel
Today’s goal

Solve large OD Cost Matrix

• Any number of origins and destinations
• Using local data or a service
• With or without a time/distance limit
• Output a single feature class
What is a local solve and a service solve?

- **Local solve**
  - Solving a network analysis on your computer using a network dataset stored on your disk.
  - Solve uses computing resources of your computer.

- **Service solve**
  - Solving a network analysis using a GIS web service.
  - Solve uses computing resources of GIS Server.
Which product should I use?

**ArcGIS Pro**
- 64-bit
- Seamless integration with services
- Stores analysis results on disk

**ArcMap**
- 32-bit…runs out of memory easily
- For 64-bit capabilities, must use ArcGIS Server or the 64-bit Background Geoprocessing Extension
- Less simple to use services
Network Analyst workflow

Local solve and service solve
1. Make analysis layer
2. Add locations
3. Solve
4. Work with output
Techniques for solving large problems
How to optimize solving a large problem

- Reduce problem size
- Chunk data
- Spatially sort data
- Solve in parallel
- Pre-calculate location fields
- Use network dataset layer
Reduce problem size

- Find only the K nearest
- Use a time/distance limit
  - Use only destinations within a reasonable straight-line distance of origins
  - Select Layer By Location
  - Watch out for the case where none are selected
Chunk data

- Break up origins and destinations into chunks of reasonable size
- Iteratively solve each chunk
- Chunk size depends on service limits or memory limits
- Consider number of origins x number of destinations

```python
# Select the origins and destinations to process
origins_where_clause = "{} > {} And {} < {}".format(self.origins_oid_field_name, origins_criteria[0],
                                                        self.origins_oid_field_name, origins_criteria[1])
arcpy.management.MakeFeatureLayer(self.origins, self.input_origins_layer, origins_where_clause)
```
Spatially sort data

- Sort geoprocessing tool
- Sort by Shape field using Peano curve
- Requires Advanced license
All origins and all destinations

26 million origins, 220k destinations
Chunk of 1000 unsorted origins
Destinations within 100 miles of 1000 origins

214,449 destinations selected (98%)
Destinations within 100 miles of 1000 sorted origins

5,653 destinations selected (3%)
Solve in parallel

- concurrent.futures

- Multiprocessing: Spin up multiple processes and run solves on multiple cores
  - Better choice!
  - Works well with arcpy
  - Easy to run from standalone python
    - If running as a geoprocessing script tool, need to use subprocess module
  - Can’t write to same gdb from multiple processes
  - Can’t share NA layer across processes

- Multithreading: Use multiple threads in the same process
  - Not good for CPU-intensive problems
  - Does not work with arcpy
Pre-calculate location fields

• Define how a point snaps to the network
• Calculate them in advance if you’re using your points more than once
• Use field mapping in Add Locations to use existing location fields
• Only works for local data. Not for services.

SourceID, SourceOID, PosAlong, SideOfEdge
Use network dataset layer

- Opening from catalog path is slow
- Even slower for licensed data or data on UNC path
- Open once by making a Network Dataset Layer; then use the layer.
Outline of today’s code

Goal: Solve OD of any size using local data or a service, write it out to a single feature class

- Components:
  - Preprocessing
    - Sorting spatially
    - Calculate network locations
  - Solving
    - Chunking
    - Solving in parallel
  - Post-processing
    - Merging results
Network Analyst workflow

Local solve and service solve
1. Make analysis layer
2. Add locations
3. Solve
4. Work with output
# Compute CD cost matrix

```python
do_line_fcs = []
job_folders_to_delete = []

# Run on multiple processes or threads when solving large CDs:
if origins_count * destinations_count > inputs["max_od_size"]:
    max_workers = os.cpu_count() // 2
    pool = Futures.ProcessPoolExecutor
else:
    max_workers = (os.cpu_count() // 2) - 1
    pool = Futures.ProcessPoolExecutor

with pool(max_workers=max_workers) as executors:
    results = executors.map(solve_od_cost_matrix, inputs_iter, ranges)
    for result in results:
        if result["solveSucceeded"]:
            do_line_fcs.append(result["outputLines"])
            job_folders_to_delete.append(result["jobFolder"])
        else:
            logger.warning("solve failed for job id %s", result["jobId"])
            logger.debug(result["solveMessages"])
```

Let’s look at some code! (Local solve)
Working with services
Which do I pick?

**Local network dataset**
- Have your own street data
- Have network analyst extension
- Do not have ArcGIS Enterprise or ArcGIS Online service credits
- Ok to run the analysis only on the CPU cores available on your machine

**ArcGIS Online service**
- Do not have your own street data
- Do not have network analyst extension
- Have ArcGIS Online service credits
- Ok to run the analysis using up to 4 concurrent processes
- Ok to run the analysis with small service limits (eg. 1000 origins and 1000 destinations)

**ArcGIS Enterprise service**
- Have your own street data
- Have network analyst extension (for Desktop and Server)
- Have ArcGIS Enterprise
- Want to run the analysis on lots of CPU cores
All about credits and service limits!

- Working with ArcGIS Online services requires service credits
  - Credit cost for each service

| Origin Destination Cost Matrix | 0.0005 credits per input origin and destination pair |

- Each service imposes a limit on the maximum problem size that can be solved in one request. For example, OD service only allows 1,000 origins and 1,000 destinations per request.
  - Service limits for each service
    - OD Cost Matrix service
    - Route service
    - Closest Facility service
    - Service Area service
    - Location-Allocation service
    - Vehicle Routing Problem service
How to publish your own routing services

- Use Publish Routing Services utility to publish your routing services to ArcGIS Enterprise

- Special considerations and settings for services when solving large problems
  - Change minimum and maximum service instances to be equal to number of physical CPU cores on your server
  - Set a high value for service usage timeout
  - Set a high value for maximum records returned by server
Let’s look at some code!

(Service solve)
Wrap-up

- Reduce problem size
- Chunk data
- Spatially sort data
- Solve in parallel
- Pre-calculate location fields
- Use network dataset layer

Code and slides:
http://esriurl.com/uc18pysInp
Please Take Our Survey on the App

Download the Esri Events app and find your event

Select the session you attended

Scroll down to find the feedback section

Complete answers and select “Submit”