

Finding the Lowest Energy Pathway

ArcGIS 2

Cole Anderson

```
In [49]: import arcpy  
         #Fill In  
  
         #these are in UTM  
  
         #hashed out later
```

Imports (run EVERY time first)

```
In [50]: import requests  
         import json  
         import zipfile  
         import arcpy
```

Retrieve the data

```

In [ ]: # a function to search MN_Geospatial Commons for specific datasets
def downloader (search_query, result_num, resource_num):
    #####
    # the URL is the MNGC API location + search terms you want
    big_url = 'https://gisdata.mn.gov/api/3/action/package_search?q=' + search
    _query

    # Sends a request to the API for the set URL
    # API returns response object
    response = requests.get(big_url, verify = False)

    # response object need to be loaded as a JSON
    json_response = json.loads(response.content)

    # this digs into the first layer of the JSON to a list of results
    result_options = json_response['result']['results']

    #select the result
    chosen_result = result_options[result_num]

    #dig further to resources and select resource number
    resources_under_result= chosen_result['resources'][resource_num]

    # find the URL for that resource for retrieval
    chosen_resource = resources_under_result['url']
    print(chosen_resource)

    # send a request to the resource URL and get response object
    URL_request = requests.get(chosen_resource)

    #save this response object to a zipfile (because response is a ZIP)
    with open('filename.zip', 'wb') as f:
        f.write(URL_request.content)
        f.close()

    #extract the zipfile contents
    with zipfile.ZipFile("filename.zip","r") as zip_ref:
        zip_ref.extractall('C:\\\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Ra
w Data')

    #confirm completion
    print('Download and extraction complete. Check notebook folder')
    #####

    #execute function for DEM and RoadCenterline datasets
    downloader('us-mn-state-metrogis-trans-road-centerlines-gac',7,1)
    downloader('dataset/elev-dtm-30m-condpr-a',1,1)

```

Create Feature Dataset to work in

```
In [38]: #find the spatial reference from RoadCenterline
spatial_ref = arcpy.Describe('C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Raw Data\\RoadCenterline.shp').spatialReference

# create a new feature dataset inside the project GDB using spatial reference from above
arcpy.CreateFeatureDataset_management(r'StreetsProject.gdb',
'Networks2', spatial_ref)
```

Out[38]:

Output

StreetsProject.gdb\Networks2

Messages

Start Time: Thursday, April 22, 2021 9:14:26 PM

Succeeded at Thursday, April 22, 2021 9:14:26 PM (Elapsed Time: 0.12 seconds)

Bring the data into the GDB/Dataset

```
In [39]: # set workspace up a Level
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj'

#bring Roadlines into the GDB feature dataset
arcpy.FeatureClassToGeodatabase_conversion('Raw Data\\RoadCenterline.shp',
'Notebooks\\StreetsProject.gdb\\Networks2')

#bring DEM into the GDB (cannot bring inside FD)
arcpy.CopyRaster_management('Raw Data\\digital_terrain_model.gdb\\DTM30CONDPR_A',
'Notebooks\\StreetsProject.gdb\\DEM')
```

Out[39]:

Output

C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb\\DEM

Messages

Start Time: Thursday, April 22, 2021 9:15:27 PM

Building Pyramids...

Calculating Statistics...

Succeeded at Thursday, April 22, 2021 9:16:18 PM (Elapsed Time: 50.96 seconds)

Calculate elevations for RoadCenterlines from DEM

```
In [41]: #workspace reset
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb'

#variable for RoadCenterline Layer
roads = "Networks2\\RoadCenterline"

#find min and max elevation for roads from DEM
arcpy.AddSurfaceInformation_3d(roads, "DEM", "Z_MAX;Z_MIN", "LINEAR")
```

Out[41]:

Output

C:\Users\Cole\Documents\GitHub\GIS5572\SemProj\Notebooks\StreetsProject.gdb\Networks2\Rc

Messages

Start Time: Thursday, April 22, 2021 9:17:04 PM

Succeeded at Thursday, April 22, 2021 9:19:42 PM (Elapsed Time: 2 minutes 38 seconds)

Convert RoadCenterlines units from M to FT

```
In [43]: # reset workspace
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb\\Networks2'

#need to divide road lengths by 3.28 to convert m to ft.
arcpy.CalculateField_management("RoadCenterline", "LENG_FT", "!Shape_Length!*
3.28", "PYTHON3", field_type = 'DOUBLE')
```

Out[43]:

Output

a Layer object

Messages

Start Time: Thursday, April 22, 2021 9:20:58 PM

Adding LENG_FT to RoadCenterline...

Succeeded at Thursday, April 22, 2021 9:21:15 PM (Elapsed Time: 17.03 seconds)

Find the slope

```
In [44]: #reset workspace
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb\\Networks2'

# calculate the slope of roads now that units are the same
arcpy.CalculateField_management("RoadCenterline", "Slope",
                                '(!Z_MAX!-!Z_MIN!)/!LENG_FT!', "PYTHON3", field_type = 'DOUBLE')
```

Out[44]:

Output

a Layer object

Messages

Start Time: Thursday, April 22, 2021 9:21:48 PM

Adding Slope to RoadCenterline...

Succeeded at Thursday, April 22, 2021 9:22:06 PM (Elapsed Time: 18.09 seconds)

Find the energy cost on the roadlines layer

```
In [45]: #Find the Energy score: Length * slope score

# 30' hill takes >2x energy vs. 15' hill? NO, =
arcpy.CalculateField_management("RoadCenterline", "E_Score",
                                '!Slope!*!LENG_FT!', "PYTHON3", field_type = 'DOUBLE')
```

Out[45]:

Output

a Layer object

Messages

Start Time: Thursday, April 22, 2021 9:22:31 PM

Adding E_Score to RoadCenterline...

Succeeded at Thursday, April 22, 2021 9:22:54 PM (Elapsed Time: 22.87 seconds)

Create Network Dataset

```
In [12]: #reset workspace
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb'
#arcpy.CopyFeatures_management("Networks2\\RoadCenterLine", "Networks2\\AllRoads")

# create a new network dataset from the RoadCenterLine Layer
#(that now contains a few calculated attributes)
arcpy.na.CreateNetworkDataset(r"Networks2",
                             "All_ND", "RoadCenterline",
                             "ELEVATION_FIELDS")

#build the network so that it exists
arcpy.na.BuildNetwork('Networks2\\All_ND')
```

```
-----
ExecuteError                                Traceback (most recent call last)
In [12]:
Line 9:      "ELEVATION_FIELDS")

File C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\na.py, in CreateNetworkDataset:
Line 4822: raise e

File C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\na.py, in CreateNetworkDataset:
Line 4819: retval = convertArcObjectToPythonObject(gp.CreateNetworkDataset_n
a(*gp_fixargs((feature_dataset, out_name, source_feature_class_names, elevati
on_model), True)))

File C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\geoprocessing\_base.py
, in <lambda>:
Line 511:     return lambda *args: val(*gp_fixargs(args, True))

ExecuteError: ERROR 030222: The network dataset cannot be created from the gi
ven parameters.
A network dataset with the specified name already exists.
Failed to execute (CreateNetworkDataset).

-----
```

Create TravelMode (manually override)

```
In [120]: #Current Travel Mode Features

#Travel Mode Name: Walking Test
#Impedence = Energy
#Distance = meter

arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb'

# make a manual travel mode inside the network to call later

nd_travel_modes = arcpy.nax.GetTravelModes('Networks2\\All_ND')

#select travel mode
travel_mode = nd_travel_modes["Walking Test"]

#called Walking Test
#arcpy.na.BuildNetwork('Networks2\\ALL_ND')
```

```
In [ ]: '''from ARCPY documentation

If a template is not specified or a value of None is used,
a blank TravelMode object will be created,
and the values of all properties must be explicitly set before using
the travel mode in a network analysis.

attributeParameters =

Lists the parameterized attributes used by the travel mode.
The property is a dictionary.
The dictionary key is a two-value tuple consisting of the attribute name and the parameter name.
The value for each item in the dictionary is the parameter value.
An empty dictionary means the travel mode uses the current default parameters of the network dataset.

Parameterized network attributes are used to model some dynamic aspect of an attributes value.
For example, a tunnel with a height restriction of 12 feet can be modeled using a parameter. A vehicles height in feet can be specified as the attribute parameter value.
If the vehicle is taller than 12 feet, this restriction will evaluate to True, thereby restricting travel through the tunnel.
Similarly, a bridge could have a parameter to specify a weight restriction.
'''
```

Create ND_Layer

```
In [121]: #reset workspace
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb'

#NDS is the new network dataset
NDS = r'Networks2\\All_ND'

#ND_layer is only temporary to allow faster processing, not saved to ROM
ND_layer = 'Working'

#input_stops = 'Networks2\\SourceModel'
#output_routes = r'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb\\Result'

#Create a new network dataset layer from the NDS for faster processing
arcpy.nax.MakeNetworkDatasetLayer(NDS, ND_layer)
```

Out[121]:

Output

a Layer object

Messages

Start Time: Thursday, April 29, 2021 1:58:58 PM

Succeeded at Thursday, April 29, 2021 1:58:58 PM (Elapsed Time: 0.06 seconds)

Everything works great up to here

Create Inputs Layer

```
In [122]: #reset workspace
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb'

#find spatial ref for OG RoadCenterline
spatial_ref = arcpy.Describe("Networks2\\RoadCenterline").spatialReference

#create a new point feature class using the spatial reference
arcpy.CreateFeatureclass_management("Networks2", "SourceModel", "POINT",
                                     spatial_reference = spatial_ref)
```

Out[122]:

Output

C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb\\Networks2\\Sc

Messages

Start Time: Thursday, April 29, 2021 1:59:14 PM

Succeeded at Thursday, April 29, 2021 1:59:16 PM (Elapsed Time: 1.25 seconds)

Add Facilities/Stops to Inputs Layer

```
In [123]: # Note: this is AFTER travel modes so don't have to redo every time if executi
ng lineraly
#reset workspace
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\N
otebooks\\StreetsProject.gdb\\Networks2'

# Select new point feature class
feature_class_source = "SourceModel"

start = arcpy.Point(479979.19, 4976268.77)
end = arcpy.Point(486014.32, 4975660.82)

#use a new cursor to insert the coordinates
cursor = arcpy.da.InsertCursor(feature_class_source, "SHAPE@XY")
cursor.insertRow([start])
cursor.insertRow([end])
del cursor
```

Initialize Route Solver

```
In [124]: ## Instantiate a Route solver object
route = arcpy.nax.Route(ND_layer)
```

Set Route Solver properties

```
In [125]: #issue doesn't seem to be here, the GUI does the same thing

nd_travel_modes = arcpy.nax.GetTravelModes(ND_layer)
travel_mode = nd_travel_modes["Walking Test"]
route.travelMode = travel_mode

#set network properties
route.timeUnits = arcpy.nax.TimeUnits.Minutes
route.accumulateAttributeNames = ["Energy"]
#route.directionsDistanceUnits =
#route.distanceUnits =
#route.networkDataSource =
#route.returnDirections =
#route.searchTolerance = 5000
#route.searchToleranceUnits = arcpy.nax.DistanceUnits.Meters
```

Load Inputs Layer

```
In [126]: #reset workspace
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\N
otebooks\\StreetsProject.gdb\\Networks2'

route.load(arcpy.nax.RouteInputDataType.Stops, 'SourceModel')
#output_path = 'Networks2\\OutputRoute'
```

Solve the Route Solver

```
In [127]: #execute the route solve
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\N
otebooks\\StreetsProject.gdb'

output_path = 'Networks2\\Least_E_Route'

result = route.solve()

#error checker
if result.solveSucceeded:
    result.export(arcpy.nax.RouteOutputDataType.Routes, output_path)
else:
    print("Solved failed")
    print(result.solverMessages(arcpy.nax.MessageSeverity.All))
```

```
In [128]: #run to delete start/end points layer, SourceModel
arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\N
otebooks\\StreetsProject.gdb\\Networks2'
arcpy.management.Delete(r"SourceModel")
```

Out[128]:

Output

true

Messages

Start Time: Thursday, April 29, 2021 1:59:44 PM

Succeeded at Thursday, April 29, 2021 1:59:44 PM (Elapsed Time: 0.03 seconds)

Deprecated Model/OLD

```

In [ ]: #deprecated

#selected manually in ArcPro: GrandAveTest roads set

arcpy.CopyFeatures_management("RoadCenterline", "Networks2\\AllRoads")
arcpy.na.CreateNetworkDataset(r"Networks",
                             "All_ND", "AllRoads",
                             "ELEVATION_FIELDS")

arcpy.na.BuildNetwork('Networks\\Test_ND')
arcpy.na.BuildNetwork('Networks\\All_ND')

#deprecated*****

#NAX module
## Source Settings
    ## Vertical connect.
## Travel Attributes
    ## Travel Modes: create one for walking, one for wheelchair
    ## Costs: Energy_Cost = E_score (assign at different quadratic?)
        ## wheelchair energy cost increases faster,
        ## but has a lower slope limit than person (.12% slope vs .3% slope),
        ## and slightly higher power limit (say, 450 W vs 400 W)
    ## Restrictions: avoid energy cost, high
## Directions
    ## Support Directions: checked
    ## Field mapping: full name to ST_name

arcpy.na.BuildNetwork('Networks\\Test_ND')
arcpy.na.BuildNetwork('Networks\\All_ND')

#deprecated*****

def routelayers(network, output, mode):
    arcpy.env.workspace = 'C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Notebooks\\StreetsProject.gdb\\Networks'

    #create a route analysis layer
    result_object = arcpy.na.MakeRouteAnalysisLayer(network, output, mode, "PRESERVE_BOTH")

    #grab the route layer object from the result object layer
    layer_object = result_object.getOutput(0)

    # add Locations from SourceModel to the route layer object as Stops
    arcpy.na.AddLocations(layer_object, "Stops", "SourceModel")

    #some network modify commands take pl

    arcpy.na.Solve(layer_object)
    layer_object.saveACopy("C:\\Users\\Cole\\Documents\\GitHub\\GIS5572\\SemProj\\Output\\"+output)

```

```
routelayers("All_ND", "Walking", "Walking")  
routelayers("All_ND", "Wheelchair", "Wheelchair")
```