Route Optimization with Suitability Modelling (Lab 2 - Part 2)

GIS 5571: ArcGIS I
University of Minnesota

Luke Zaruba
October 25, 2022

In [1]:

```
import arcpy
import requests
import numpy as np

import os
import glob
import shutil
```

In [2]:

```
from arcpy import env
env.scratchWorkspace = env.scratchFolder
env.workspace = env.scratchWorkspace
```

Elevation Data Prep

In [3]:

```
def downloadLAZ(county, tile):
    # Set up URL
    county = county.lower()
    base_url = "https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/CTY/laz/"
    base_url = base_url.replace("CTY", county)

    # Download Tile
    tile_url = base_url + tile + ".laz"

    resp = requests.get(tile_url, stream = True)

    # Write Tile to File
    laz_name = f"./laz/{tile}.laz"

    with open(laz_name, "wb") as laz:
        laz.write(resp.content)
        print(f"Download complete for tile {tile}")


def convertLAZtoLAS(laz):
    # Create Variables
    sr = 'PROJCS["NAD_1983_UTM_Zone_15N",GEOGCS["GCS_North_American_1983",DATUM["D_North_American_1983",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Greenwich"
    out_dir = "./las"

    # Convert
    arcpy.conversion.ConvertLas(laz, out_dir, "1.4", "6", "NO_COMPRESSION", "REARRANGE_POINTS", None, "ALL_FILES", sr)
    print(f"{laz} successfully converted.")
```

In [4]:

```
# Create Lists of Tiles Needed
olmsted_tiles= ['4342-30-59', '4342-30-60', '4342-30-61', '4342-30-62', '4342-31-59', '4342-31-60', '4342-31-61', '4342-31-62']
wabasha_tiles = ['4342-28-59', '4342-28-60', '4342-28-61', '4342-28-62', '4342-29-59', '4342-29-60', '4342-29-61', '4342-29-62']
winona_tiles = ['4342-28-63', '4342-29-63', '4342-30-63', '4342-31-63']

# Iteratively run the downloadLAZ Function
for i in olmsted_tiles:
    downloadLAZ("olmsted", i)
for i in wabasha_tiles:
    downloadLAZ("wabasha", i)
for i in winona_tiles:
    downloadLAZ("winona", i)
```

```
Download complete for tile 4342-30-59
Download complete for tile 4342-30-60
Download complete for tile 4342-30-61
Download complete for tile 4342-30-62
Download complete for tile 4342-31-59
Download complete for tile 4342-31-60
Download complete for tile 4342-31-61
Download complete for tile 4342-31-62
Download complete for tile 4342-28-59
Download complete for tile 4342-28-60
Download complete for tile 4342-28-61
Download complete for tile 4342-28-62
Download complete for tile 4342-29-59
Download complete for tile 4342-29-60
Download complete for tile 4342-29-61
Download complete for tile 4342-29-62
Download complete for tile 4342-28-63
Download complete for tile 4342-29-63
Download complete for tile 4342-30-63
Download complete for tile 4342-31-63
```

In [5]:

```
# Create Combined Lists of All Tile Names
combined_list = olmsted_tiles + wabasha_tiles + winona_tiles

# Iteratively Convert LAZ to LAS
for i in combined_list:
    laz = f"./laz/{i}.laz"
    convertLAZtoLAS(laz)
```

```
./laz/4342-30-59.laz successfully converted.
./laz/4342-30-60.laz successfully converted.
./laz/4342-30-61.laz successfully converted.
./laz/4342-30-62.laz successfully converted.
./laz/4342-31-59.laz successfully converted.
./laz/4342-31-60.laz successfully converted.
./laz/4342-31-61.laz successfully converted.
./laz/4342-31-62.laz successfully converted.
./laz/4342-28-59.laz successfully converted.
./laz/4342-28-60.laz successfully converted.
./laz/4342-28-61.laz successfully converted.
./laz/4342-28-62.laz successfully converted.
./laz/4342-29-59.laz successfully converted.
./laz/4342-29-60.laz successfully converted.
./laz/4342-29-61.laz successfully converted.
./laz/4342-29-62.laz successfully converted.
./laz/4342-28-63.laz successfully converted.
./laz/4342-29-63.laz successfully converted.
./laz/4342-30-63.laz successfully converted.
./laz/4342-31-63.laz successfully converted.
```

In [6]:

```
# Create LASD
sr = 'PROJCS["NAD_1983_UTM_Zone_15N",GEOGCS["GCS_North_American_1983",DATUM["D_North_American_1983",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Greenwich",0.0
las_files = r"'C:\gitFiles\GIS5571\Lab2\Part 2\Lab2_Part2\las\4342-28-59.las';'C:\gitFiles\GIS5571\Lab2\Part 2\Lab2_Part2\las\4342-28-60.las';'C:\gitFiles\GIS5571\Lab2\
lasd_name = r"C:\gitFiles\GIS5571\Lab2\Part 2\Lab2_Part2\final_lasd.lasd"

lasd = arcpy.management.CreateLasDataset(las_files, lasd_name, "NO_RECURSION", None, sr, "COMPUTE_STATS", "ABSOLUTE_PATHS", "NO_FILES", "DEFAULT", None, "INTERSECTED_FI
```

In [7]:

```
# Convert LASD to DEM
dem_name = r"c:\gitFiles\GIS5571\Lab2\Part 2\lab2_part2\lab2_part2.gdb\dem_lm"

dem = arcpy.conversion.LasDatasetToRaster("final_lasd.lasd", dem_name, "ELEVATION", None, "FLOAT", "CELLSIZE", 1, 1)
```

In [8]:

```
# Calculate Slope
slope_name = r"C:\gitFiles\GIS5571\Lab2\Part 2\Lab2_Part2\Lab2_Part2.gdb\slope"

slope = arcpy.sa.Slope("dem_lm", "DEGREE", 1, "PLANAR", "METER")

slope.save(slope_name)
```

Landcover Data Prep

In [9]:

```
# Create AOI
bndry = arcpy.ddd.RasterDomain(slope, "aoi", "POLYGON")

# Clip Landcover to AOI
out_lc = r"C:\gitFiles\GIS5571\Lab2\Part 2\Lab2_Part2\Lab2_Part2.gdb\landcover_aoi"

lc_aoi = arcpy.management.Clip("landcover_impervious_statewide2013_v2.tif", "564958.63 4875662.12 577613.63 4889678.12", out_lc, bndry, "255", "ClippingGeometry", "MAIN
```

Create Model

In [10]:

```
# Reclassify Landcover
lc_reclass_v1 = arcpy.sa.Reclassify("landcover_aoi", "Value", "1 100 5;101 2;102 2;103 2;104 3;105 2;106 3;107 3;108 4;109 1;110 1", "DATA")
```

In [11]:

```
# Reclassify Slope
slope_reclass_v1 = arcpy.sa.Reclassify("slope", "VALUE", "0 2.084853 5;2.084853 3.822231 4;3.822231 6.602035 3;6.602035 18.416204 2;18.416204 88.606262 1", "DATA")
```

In [12]:

```
# Create Several Cost Surfaces to Test Different Model Weights
for i in np.arange(0.1, 1.0, 0.1):
    # Set Weights
    slope_w_rnd = round(i, 1)
    landcover_w_rnd = round((1 - i), 1)

    # Calculate Cost and Save as New Raster
    cost = ((((arcpy.Raster(slope_reclass_v1) * slope_w_rnd) + (arcpy.Raster(lc_reclass_v1) * landcover_w_rnd)) * -1) + 6)

    # Create Cost Path
    cPath_name = fr"cPath_{str(slope_w_rnd)[2:3]}s_{str(landcover_w_rnd)[2:3]}lc"
    cPath = arcpy.sa.OptimalRegionConnections("origin_destination", cPath_name, in_cost_raster = cost)
```

Cleanup

In [13]:

```
# Final Cleanup and Deleting Temp Files
os.chdir("C:\gitFiles\GIS5571\Lab2\Part 2\Lab2_Part2")
del_list = glob.glob("tmp*")

#for i in del_list:
    #shutil.rmtree(i)
```