CostSurfaceAnalysis

November 29, 2022

1 Cost Surface Analysis Operations

```
[6]: import arcpy
     import requests
     import io
     import zipfile
```

1.1 Display Dory's Farm and North Picnic Area in ArcGIS Pro from CSV Table to XY to Point Tool

```
[2]: arcpy.management.XYTableToPoint(r"C:\Users\Alexander Danielson\Desktop\Fall␣
      ↪2022Spring2023\ArcGIS I\Lab3\Lab3\DoryNPATable.csv", r"C:\Users\Alexander␣
      ↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
      ↪gdb\DoryNPATable_XYTableToPoint", "Long", "Lat", None,␣
      ↪'GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.
      ↪257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]];-400␣
      ↪-400 1000000000;-100000 10000;-100000 10000;8.98315284119521E-09;0.001;0.001;
      ↪IsHighPrecision')
```

```
[2]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
     I\\Lab3\\Lab3\\Lab3.gdb\\DoryNPATable_XYTableToPoint'>
```

1.2 Form Optimal Route For Dory's Given Extent of Area Using Data From MNGEO and Performing Raster Analysis

1.3 Extract County Data from MNGEO (including MN NLCD, DEM, Streams, Impervious Surfuces/Roads) to Create Optimal Route

```
[3]: CountyBoundsLink = "https://resources.gisdata.mn.gov/pub/gdrs/data/pub/
      ↪us_mn_state_dnr/bdry_counties_in_minnesota/shp_bdry_counties_in_minnesota.
      ↪zip"
```

```
[4]: CountyBoundsLink
```

```
[4]: 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/bdry_countie
     s_in_minnesota/shp_bdry_counties_in_minnesota.zip'
```

```
[7]: OutputSource = requests.post(CountyBoundsLink)
```

```
[8]: Sources = OutputSource.content
```

```
[9]: Zipp = zipfile.ZipFile(io.BytesIO(Sources))
```

```
[10]: Zipp.extractall(r'C:\Users\Alexander Danielson\Desktop\Fall␣
       ↪2022Spring2023\ArcGIS I\Lab3\Lab3')
```

```
[11]: CountyNLCDLink = "https://resources.gisdata.mn.gov/pub/gdrs/data/pub/
       ↪us_mn_state_dnr/biota_landcover_nlcd_mn_2019/
       ↪tif_biota_landcover_nlcd_mn_2019.zip"
```

```
[12]: CountyNLCDLink
```

```
[12]: 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/biota_landco
      ver_nlcd_mn_2019/tif_biota_landcover_nlcd_mn_2019.zip'
```

```
[13]: OutputNLCDSource = requests.post(CountyNLCDLink)
```

```
[14]: SourcesNLCD = OutputNLCDSource.content
```

```
[15]: ZippNLCD = zipfile.ZipFile(io.BytesIO(SourcesNLCD))
```

```
[16]: ZippNLCD.extractall(r'C:\Users\Alexander Danielson\Desktop\Fall␣
       ↪2022Spring2023\ArcGIS I\Lab3\Lab3')
```

```
[17]: CountyDEMLink = 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/
       ↪us_mn_state_dnr/elev_30m_digital_elevation_model/
       ↪fgdb_elev_30m_digital_elevation_model.zip'
```

```
[18]: CountyDEMLink
```

```
[18]: 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/elev_30m_dig
      ital_elevation_model/fgdb_elev_30m_digital_elevation_model.zip'
```

```
[19]: OutputDEMSource = requests.post(CountyDEMLink)
```

```
[20]: SourcesDEM = OutputDEMSource.content
```

```
[21]: ZippDEM = zipfile.ZipFile(io.BytesIO(SourcesDEM))
```

```
[22]: ZippDEM.extractall(r'C:\Users\Alexander Danielson\Desktop\Fall␣
       ↪2022Spring2023\ArcGIS I\Lab3\Lab3')
```

```
[23]:
```

```
CountyStreamLink = 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/
↪us_mn_state_dnr/water_strahler_stream_order/shp_water_strahler_stream_order.
↪zip'
```

```
[24]: CountyStreamLink
```

```
[24]: 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/water_strahl
      er_stream_order/shp_water_strahler_stream_order.zip'
```

```
[25]: OutputStreamSource = requests.post(CountyStreamLink)
```

```
[26]: SourcesStream = OutputStreamSource.content
```

```
[27]: ZippStream = zipfile.ZipFile(io.BytesIO(SourcesStream))
```

```
[28]: ZippStream.extractall(r'C:\Users\Alexander Danielson\Desktop\Fall␣
       ↪2022Spring2023\ArcGIS I\Lab3\Lab3')
```

```
[29]: MNRoads = 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dot/
       ↪trans_roads_mndot_tis/shp_trans_roads_mndot_tis.zip'
```

```
[30]: MNRoads
```

```
[30]: 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dot/trans_roads_
      mndot_tis/shp_trans_roads_mndot_tis.zip'
```

```
[31]: OutputMNRoads = requests.post(MNRoads)
```

```
[32]: SourcesMNRoads = OutputMNRoads.content
```

```
[33]: ZipRoads = zipfile.ZipFile(io.BytesIO(SourcesMNRoads))
```

```
[34]: ZipRoads.extractall(r'C:\Users\Alexander Danielson\Desktop\Fall␣
       ↪2022Spring2023\ArcGIS I\Lab3\Lab3')
```

1.4 Dissolve Optimal Counties and Geoprocess Other Ancillary Data Then Transform Features to Reclassify for Weights

1.4.1 Dissolve Operation for Optimal Study Area

```
[35]: arcpy.management.Dissolve("mn_county_boundaries selection", r"C:
       ↪\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
       ↪I\Lab3\Lab3\Lab3.gdb\DoryNPAStudyArea", None, None, "MULTI_PART",␣
       ↪"DISSOLVE_LINES", '')
```

```
[35]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab3\\Lab3\\Lab3.gdb\\DoryNPAStudyArea'>
```

1.4.2 Extract by Mask (NLCD) using Optimal Study Area

[36]:
```
out_raster = arcpy.sa.ExtractByMask("NLCD_2019_Land_Cover.tif",␣
 ↪"DoryNPAStudyArea", "INSIDE", '524966.6376 4853462.8394 637916.1448 4922619.
 ↪9426␣
 ↪PROJCS["NAD_1983_UTM_Zone_15N",GEOGCS["GCS_North_American_1983",DATUM["D_North_American_198
 ↪0,298.257222101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.
 ↪0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.
 ↪0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-93.
 ↪0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_Of_Origin",0.
 ↪0],UNIT["Meter",1.0]]'); out_raster.save(r"C:\Users\Alexander␣
 ↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
 ↪gdb\Extract_NLCD")
```

1.4.3 Clip (Roads) using Optimal Study Area

[39]:
```
arcpy.analysis.Clip("STREETS_LOAD", "DoryNPAStudyArea", r"C:\Users\Alexander␣
 ↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
 ↪gdb\StreetsClip", None)
```

[39]:
```
<Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
I\\Lab3\\Lab3\\Lab3.gdb\\StreetsClip'>
```

1.4.4 Extract by Mask (DEM) using Optimal Study Area

[37]:
```
out_raster = arcpy.sa.ExtractByMask("digital_elevation_model_30m",␣
 ↪"DoryNPAStudyArea", "INSIDE", '524966.6376 4853462.8394 637916.1448 4922619.
 ↪9426␣
 ↪PROJCS["NAD_1983_UTM_Zone_15N",GEOGCS["GCS_North_American_1983",DATUM["D_North_American_198
 ↪0,298.257222101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.
 ↪0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.
 ↪0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-93.
 ↪0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_Of_Origin",0.
 ↪0],UNIT["Meter",1.0]]'); out_raster.save(r"C:\Users\Alexander␣
 ↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
 ↪gdb\Extract_DEM")
```

1.4.5 Clip (Streams) using Optimal Study Area

[38]:
```
arcpy.analysis.Clip("streams_with_strahler_stream_order", "DoryNPAStudyArea",␣
 ↪r"C:\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
 ↪I\Lab3\Lab3\Lab3.gdb\StreamsClip", None)
```

[38]:
```
<Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
I\\Lab3\\Lab3\\Lab3.gdb\\StreamsClip'>
```

1.4.6 Create Slope out of DEM

```
[40]: arcpy.ddd.Slope("DEMArea", r"C:\Users\Alexander Danielson\Desktop\Fall␣
       ↪2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.gdb\SlopeDEM", "PERCENT_RISE", 1,␣
       ↪"PLANAR", "METER")
```

```
[40]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab3\\Lab3\\Lab3.gdb\\SlopeDEM'>
```

1.4.7 Buffer Road Features 100 Meters

```
[42]: arcpy.analysis.Buffer("StreetsClip", r"C:\Users\Alexander␣
       ↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
       ↪gdb\StreetsClip_Buffer100M", "100 Meters", "FULL", "ROUND", "NONE", None,␣
       ↪"PLANAR")
```

```
[42]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab3\\Lab3\\Lab3.gdb\\StreetsClip_Buffer100M'>
```

1.4.8 Erase Riverways Where Intersection of 100 Meter Road Buffers Concide

```
[ ]: arcpy.analysis.Erase("StreamsClip", "StreetsClip_Buffer100M", r"C:
      ↪\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
      ↪I\Lab3\Lab3\Lab3.gdb\StreetsClip_Buffer100M_Erase", None)
```

1.4.9 Buffer Clipped River Feature and Convert to Raster for being Recognized in Weights then Use in Raster Calculator (In following steps)

```
[43]: arcpy.analysis.Buffer("StreetsClip_Buffer100M_Erase", r"C:\Users\Alexander␣
       ↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
       ↪gdb\StreetsClip_Buffer100_Buffer", "10 Meters", "FULL", "ROUND", "NONE",␣
       ↪None, "PLANAR")
```

```
[43]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab3\\Lab3\\Lab3.gdb\\StreetsClip_Buffer100_Buffer'>
```

```
[44]: arcpy.conversion.FeatureToRaster("StreetsClip_Buffer100M_Erase", "FW_ID", r"C:
       ↪\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
       ↪I\Lab3\Lab3\Lab3.gdb\BufferedRiversVector", 276.583641600005)
```

```
[44]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab3\\Lab3\\Lab3.gdb\\BufferedRiversVector'>
```

1.5 Reclassifcation

1.5.1 Relcassify NLCD for Land Use

```
[2]: arcpy.ddd.Reclassify("NLCDArea", "Value", "11 15.200000 1;15.200000 19.400000 2;
      ↪19.400000 23.600000 3;23.600000 27.800000 4;27.800000 32 5;32 36.200000 6;36.
      ↪200000 40.400000 7;40.400000 44.600000 8;44.600000 48.800000 9;48.800000 53␣
      ↪10;53 57.200000 11;57.200000 61.400000 12;61.400000 65.600000 13;65.600000␣
      ↪69.800000 14;69.800000 74 15;74 78.200000 16;78.200000 82.400000 17;82.
      ↪400000 86.600000 18;86.600000 90.800000 19;90.800000 95 20", r"C:
      ↪\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
      ↪I\Lab3\Lab3\Lab3.gdb\Reclass_NLCD20classes", "DATA")
```

```
[2]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
     I\\Lab3\\Lab3\\Lab3.gdb\\Reclass_NLCD20classes'>
```

1.5.2 Relcassify Slope

```
[1]: arcpy.ddd.Reclassify("SlopeDEM", "VALUE", "0 26.673990 1;26.673990 53.347980 2;
      ↪53.347980 80.021970 3;80.021970 106.695959 4;106.695959 133.369949 5;133.
      ↪369949 160.043939 6;160.043939 186.717929 7;186.717929 213.391919 8;213.
      ↪391919 240.065909 9;240.065909 266.739899 10;266.739899 293.413889 11;293.
      ↪413889 320.087878 12;320.087878 346.761868 13;346.761868 373.435858 14;373.
      ↪435858 400.109848 15;400.109848 426.783838 16;426.783838 453.457828 17;453.
      ↪457828 480.131818 18;480.131818 506.805807 19;506.805807 533.479797 20", r"C:
      ↪\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
      ↪I\Lab3\Lab3\Lab3.gdb\Reclass_Slope20classes", "DATA")
```

```
[1]: <Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
     I\\Lab3\\Lab3\\Lab3.gdb\\Reclass_Slope20classes'>
```

1.5.3 Reclassify Streams for Study Area 1 = Rivers and 0 = Study Area

```
[49]: output_raster = arcpy.ia.RasterCalculator(' Con(IsNull(␣
       ↪"BufferedRiversVector"),0, "BufferedRiversVector")'); output_raster.save(r"C:
       ↪\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
       ↪I\Lab3\Lab3\Lab3.gdb\RasterCalStreams01")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
In  [49]:
Line 1:     output_raster = arcpy.ia.RasterCalculator(' Con(IsNull(␣
 ↪"BufferedRiversVector"),0, "BufferedRiversVector")'); output_raster.save(r"C:
 ↪\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
 ↪gdb\RasterCalStreams01")

TypeError: RasterCalculator() missing 2 required positional arguments:␣
 ↪'input_names' and 'expression'
```

```
---------------------------------------------------------------------------
```

2 Create Buffers For Dory's Farm and North Picnic Area and Seperate Buffers

2.1 Create Buffers For Dory's Farm and North Picnic Area and Seperate Buffers into Own Feature Class then Convert to Raster

[5]:
```
arcpy.analysis.Buffer("DoryNPATable_XYTableToPoint", r"C:\Users\Alexander␣
↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
↪gdb\DoryNPATable_XYTableT_Buffer", "100 Meters", "FULL", "ROUND", "NONE",␣
↪None, "PLANAR")
```

[5]:
```
<Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
I\\Lab3\\Lab3\\Lab3.gdb\\DoryNPATable_XYTableT_Buffer'>
```

[8]:
```
arcpy.conversion.FeatureToRaster("DorysFarm", "OBJECTID", r"C:\Users\Alexander␣
↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
↪gdb\DoryBuffersRaster", 0.000299576288000026)
```

[8]:
```
<Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
I\\Lab3\\Lab3\\Lab3.gdb\\DoryBuffersRaster'>
```

[9]:
```
arcpy.conversion.FeatureToRaster("NPA", "OBJECTID", r"C:\Users\Alexander␣
↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.
↪gdb\NPABuffersRaster", 0.000299576288000026)
```

[9]:
```
<Result 'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
I\\Lab3\\Lab3\\Lab3.gdb\\NPABuffersRaster'>
```

3 Input Reclassifed Rasters to Be Weighted and Create Cost Surface Analysis

3.1 Create Weighted Raster

[11]:
```
out_raster = arcpy.ia.WeightedSum("Reclass_NLCD20classes Value 1;ReclassDEM␣
↪Value 3"); out_raster.save(r"C:\Users\Alexander Danielson\Desktop\Fall␣
↪2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.gdb\Weighte_Recl3DEMHighNLCD20Class")
```

3.1.1 Use Raster Calculator For Multiplying All Rasters Together (Landcover, Roads, and Streams)

[106]:
```
output_raster = arcpy.ia.RasterCalculator('StreamOutput = ␣
↪Con(IsNull("ReClassStreams"), 0 , "ReClassStreams" )'); output_raster.
↪save(r"C:\Users\Alexander Danielson\Desktop\Fall 2022Springoutput_raster =␣
↪arcpy.ia.RasterCalculator('StreamOutput =  Con(IsNull("ReClassStreams"), 0 ,␣
↪"ReClassStreams" )'); output_raster.save(r"C:\Users\Alexander␣
↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab2\Lab2\Lab2.
↪gdb\RasterCalculatorStreams")2023\ArcGIS I\Lab2\Lab2\Lab2.
↪gdb\RasterCalculatorStreams")
```

```
---------------------------------------------------------------------------
SyntaxError                               Traceback (most recent call last)
File C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\ast.py, in␣
 ↪parse:
Line 50:    return compile(source, filename, mode, flags,

SyntaxError: invalid syntax (<string>, line 1)
---------------------------------------------------------------------------
```

3.2 Create Cost Surface (Distance) Raster

[12]:
```
out_distance_raster = arcpy.sa.CostDistance("NPA",␣
↪"Weighte_Recl3DEMHighNLCD20Class", None, None, None, None, None, None, '');␣
↪out_distance_raster.save(r"C:\Users\Alexander Danielson\Desktop\Fall␣
↪2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.gdb\CostDis_NLCDHigher4")
```

3.3 Create Back Link

[13]:
```
out_backlink_raster = arcpy.sa.CostBackLink("NPA",␣
↪"Weighte_Recl3DEMHighNLCD20Class", None, None, None, None, None, None, '');␣
↪out_backlink_raster.save(r"C:\Users\Alexander Danielson\Desktop\Fall␣
↪2022Spring2023\ArcGIS I\Lab3\Lab3\Lab3.gdb\CostBac_NLCD4")
```

3.4 Create Cost Surface Allocation Raster

[110]:
```
out_allocation_raster = arcpy.sa.CostAllocation("DisplayXYPathDoryFarmtoNPA",␣
↪"WeightedRaster", None, None, "OBJECTID", None, None, None, None, None,␣
↪None, ''); out_allocation_raster.save(r"C:\Users\Alexander␣
↪Danielson\Desktop\Fall 2022Spring2023\ArcGIS I\Lab2\Lab2\Lab2.
↪gdb\CosrSurfaceAllo")
```

3.5 Create Cost Path for Final Route

[14]:

```
out_raster = arcpy.sa.CostPath("NPA", "out_distance_raster4",␣
↪"out_backlink_raster4", "EACH_CELL", "ORIG_FID", "INPUT_RANGE"); out_raster.
↪save(r"C:\Users\Alexander Danielson\Desktop\Fall 2022Spring2023\ArcGIS␣
↪I\Lab3\Lab3\Lab3.gdb\CostPat_4")
```

[ ]: