Lab Report

Title: Lab 3 - Part 1

Notice: Dr. Bryan Runck

Author: Mattie Gisselbeck

Date: December 2, 2022

Repository: https://github.com/mgisselbeck/GIS5571

Time Spent: 15 hours

Abstract

The objective of this analysis is to compare three different weighting approaches to generating a cost surface for determining optimal paths for Dory, within Dory's preferences. The data was sourced from Minnesota Geospatial Commons and was scraped through an ETL in ArcGIS Pro via a Python notebook. Based on Dory's preferences, the objective is to find a path that avoids farm fields, water bodies without a bridge, and has a gradual slope. The results are shown in the figures below (see Figure 2 through Figure 7). The data flow diagram above (Figure 1) shows all the variables and commands I applied in finding an optimal route for Dory. The results could be qualitatively verified by using the 'ArcGIS Pro – Topographic' by visually comparing the route with the hill shade of the map. In this lab, I was able to build off Lab 2 with creating an ETL and building a cost path analysis model. The objectives of this lab helped me to gain practical applications of how I would create multiple cost path analyses through ArcPy or an open-source package.

Problem Statement

The objective of this analysis is to compare three different weighting approaches to generating a cost surface for determining optimal paths for Dory, within Dory's preferences. Dory's preferences are: (1) Dory prefers to not walk through any farm fields, (2) she doesn't like crossing water bodies if there isn't a bridge (though sometimes she doesn't mind if she's wearing her waders), and (3) prefers a path with the most gradual slope (Runck, 2022).

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	County Boundaries in Minnesota	Wabasha, Winona, and Olmstead County	.shp (Vector)	N/A	Minnesota Geospatial Commons	ETL
2	2012 Assessed Streams	Assessed Streams for Minnesota in 2012	.shp (Vector)	N/A	Minnesota Geospatial Commons	ETL
3	NCLD 2019 Land Cover, Minnesota	Land Cover Classification for Minnesota	TIFF (Raster)	Land Cover Classification	Minnesota Geospatial Commons	ETL
4	Minnesota Digital Elevation Model - 30 Meter Resolution	Elevation (Wabasha, Winona, and Olmsted County)	TIFF (Raster)	Elevation	Minnesota Geospatial Commons	ETL
5	Roads, Minnesota, 2012	Road Centerlines for all Public Roads in Minnesota	.shp (Vector)	N/A	Minnesota Geospatial Commons	ETL

Table 1. Required Data

Input Data

The table below is a collection of data from the Minnesota Geospatial Commons. Data was scraped through an ETL in ArcGIS Pro via a Python notebook. All the data described below will be used in a cost path analysis to find three optimal routes for Dory.

#	Title	Purpose in Analysis	Link to Source
1	County Boundaries in Minnesota	To be used in the cost surface equation to find optimal routes for Dory	Minnesota Geospatial Commons
2	2012 Assessed Streams	To be used in the cost surface equation to find optimal routes for Dory	Minnesota Geospatial Commons
3	NCLD 2019 Land Cover, Minnesota	To be used in the cost surface equation to find optimal routes for Dory	Minnesota Geospatial Commons
4	Minnesota Digital Elevation Model - 30 Meter Resolution	To calculate the slope and add the output into the cost surface analysis to find optimal routes for Dory	Minnesota Geospatial Commons

5	Roads, Minnesota, 2012	To be used in the cost surface equation to find optimal routes for Dory	[Minnesota Geospatial Commons]

Table 2. Input Data

Methods

Based on Dory's preferences, the objective is to find three paths that avoid farm fields, water bodies without a bridge, and have a gradual slope. The starting point, Dory's Farm, is 44.127985, -92.148796 and end point, picnic area in Whitewater State Park, is 44.054852, -92.045780.

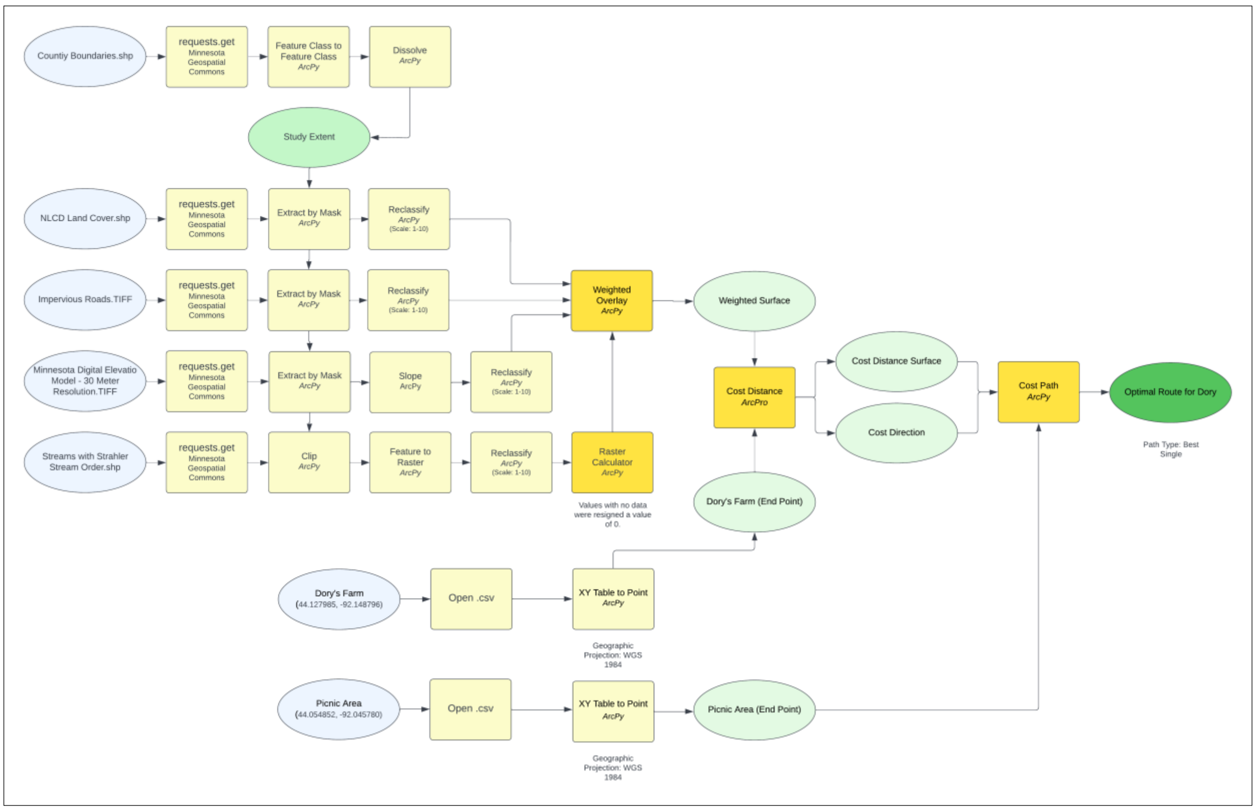

Figure 1. Cost Path Analysis.

Part 1.1: Import Packages and Request Data from Minnesota Geospatial Commons

(See Lab 2 – Part 1 Python Notebook)

Part 1.2: Create a Study Extent

To create the study extent, I used 'Feature Class to Feature Class' to apply a SQL expression that selects only Winona, Olmsted, and Wabasha and create a new feature class.

```
# Create Study Extent (Feature Class to Feature Class)
arcpy.conversion.FeatureClassToFeatureClass("mn_county_boundaries",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb", "StudyExtent", "CTY_NAME = 'Wabasha'
Or CTY_NAME = 'Winona' Or CTY_NAME = 'Olmsted'", 'AREA "AREA" true true false 19 Double 0
0,First,#,mn_county_boundaries,AREA,-1,-1;PERIMETER "PERIMETER" true true false 19 Double 0
0,First,#,mn_county_boundaries,PERIMETER,-1,-1;CTYONLY_ "CTYONLY_" true true false 19 Double 0

# Dissolve County Boundaries
arcpy.management.Dissolve("StudyExtent",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\StudyExtent_Dissolve", None, None,
"MULTI_PART", "UNSPLIT_LINES")
```

Part 1.3: Impervious Routes

The impervious roads were extracted by mask to match the study extent. Reclassification of impervious roads was executed by using the reclassify command (See Table 3).

```
# Extract by Mask
Extract_Roads = arcpy.sa.ExtractByMask("NLCD_2019_Land_Cover_Impervious_Descriptor.tif",
"StudyExtent_Dissolve");
Extract_Roads.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Extract_Roads")

# Reclassify Roads (Scale: 1-10) (See Table 3)
arcpy.ddd.Reclassify("Extract_Roads", "Class_Name", "Unclassified 10;'Primary road' 1;'Secondary road'
1;'Tertiary road' 1;'Non-road non-energy impervious' 2;'LCMAP impervious' 3;'Wind turbines' 7",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_Roads", "DATA")
```

Value	New
Unclassified	10
Primary	1
Secondary	1
Tertiary	1
Non-road Impervious	2
LCMAP Impervious	3
Wind Turbines	7

Table 3. Reclassification for Impervious Roads for Cost Path A & Cost Path C.

Part 1.4: Digital Elevation Model (DEM)

The 'Extract by Mask' command was used to clip the DEM into the study extent. To calculate the slope, the 'Slope' tool was ran. The slope was reclassified by using the 'Reclassify' command (See Table 4).

```
# Extract by Mask
Extract_DEM =
arcpy.sa.ExtractByMask(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\elev_30m_digital_elevation_mo
del.gdb\digital_elevation_model_30m", "StudyExtent_Dissolve");
Extract_DEM.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Extract_DEM")

# Calculate Slope
arcpy.ddd.Slope("Extract_DEM",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\DEM_Slope", "PERCENT_RISE", 1,
"PLANAR", "METER")
```

```
# Reclassify Slope (Scale: 1-10) (See Table 4)
arcpy.ddd.Reclassify("DEM_Slope", "VALUE", "0 3 1;3 6 2;6 10 3;10 15 4;15 20 8;20 25 9;25 30 10;30 40 10;40
60 10;60 100 10;100 1000 10",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_DEM_2", "DATA")
```

Start	End	New
0	3	1
3	6	2
6	12	3
12	15	4
15	18	8
18	21	9
21	24	10
24	27	10
27	30	10
30	533.47	10

Table 4. Reclassification for DEM Slope for Cost Path A.

Part 1.5: NLCD

The NLCD was extracted by 'Extract by Mask' command to fit the study extent and then reclassified (See Table 5).

```
# Extract by Mask
Extract_NLCD =
arcpy.sa.ExtractByMask(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\tif_biota_landcover_nlcd_mn_20
```

```
19\NLCD_2019_Land_Cover.tif", "StudyExtent_Dissolve");
Extract_NLCD.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Extract_NLCD")

# Reclassify NLCD
arcpy.ddd.Reclassify("Extract_NLCD", "NLCD_Land", "'Open Water' 10;'Developed, Open Space' 2;'Developed, Low Intensity' 2;'Developed, Medium Intensity' 2;'Developed, High Intensity' 2;'Barren Land' 5;'Deciduous Forest' 7;'Evergreen Forest' 7;'Mixed Forest' 7;Shrub/Scrub 7;Herbaceous 7;Hay/Pasture 9;'Cultivated Crops' 9;'Woody Wetlands' 10;'Emergent Herbaceous Wetlands' 10", r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_NLCD", "DATA")
```

Value	New
Open Water	10
Developed, Open Space	1
Developed, Low Intensity	2
Developed, Medium Intensity	2
Developed, High	2
Barren Land	5
Deciduous Forest	7
Evergreen Forest	7
Mixed Forest	9
Shrub/Scrub	9
Herbaceous	9
Hay/Pasture	9
Cultivated Crops	10

Table 5. Reclassification for NLCD for Cost Path A.

Part 1.6: Streams with Strahler Stream Order

Streams with Strahler Stream Order were clipped to the study extent by using the “Clip” tool and then converted into a raster. The reason behind converting the feature to a raster is so we can use it in the raster calculator and weighted overlay. The streams were reclassified using the ‘Reclassify’ command. The ‘Raster Calculator’ command was implemented to convert the null values within the dataset to 1 using the SQL expression: Con(IsNull("Reclass_Streams"),1, "Reclass_Streams".

```
# Clip Streams
arcpy.analysis.Clip("streams_with_strahler_stream_order", "StudyExtent_Dissolve",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Streams_Clip", None)

# Feature to Raster
arcpy.conversion.FeatureToRaster("Streams_Clip", "SO_VALUE",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Feature_Stre1",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_DEM_Slope"







# Reclassify Streams
arcpy.ddd.Reclassify("Feature_Stre1", "Value", "1 1;2 2;3 3;4 7;5 8;6 9;8 10",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_Streams", "DATA")

# Raster Calculator (Note: Command only worked in Geoprocessing pane in ArcGIS Pro)
output_raster = arcpy.ia.RasterCalculator(' Con(IsNull("Reclass_Streams"),1, "Reclass_Streams")');
output_raster.save(r"c:\Users\gisse015\documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\RC_Streams")
```

Part 1.7: Dory’s Farm (Start Point) and Picnic Area (End Point)

```
# Dory's Farm (Start Point)
arcpy.management.XYTableToPoint("Start.csv",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Start_Point", "LONG", "LAT", None,
'GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PR
```

```
IMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]];-400 -400 1000000000;-100000 10000;-100000
10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision')

# Picnic Area (End Point)
arcpy.management.XYTableToPoint("End.csv",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\End_Point", "LONG", "LAT", None,
'GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PR
IMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]];-400 -400 1000000000;-100000 10000;-100000
10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision')

# Point to Raster - Dory's Farm (Start Point)
arcpy.conversion.PointToRaster("Start_Point", "OBJECTID",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Start_RasterPoint", "MOST_FREQUENT",
"NONE", None, "BUILD")

# Point to Raster - Picnic Area (End Point)
arcpy.conversion.PointToRaster("End_Point", "OBJECTID",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\End_RasterPoint", "MOST_FREQUENT",
"NONE", None, "BUILD")
```

Part 1.8: Cost Path A

```
# Weighted Overlay (Cost Surface)
Cost_Surface = arcpy.sa.WeightedOverlay("('Reclass_Streams_A' 25 'Value' (1 1; 2 2; 3 3; 7 8; 8 9; 9 10; 10 10;
NODATA NODATA); 'Reclass_DEM_Slope_A' 25 'Value' (2 2; 3 3; 4 4; 7 8; 8 9; 10 10; NODATA NODATA);
'Reclass_NLCD_A' 25 'Value' (2 2; 5 6; 7 8; 9 9; 10 10; NODATA NODATA); 'Reclass_Roads_A' 25 'Value' (1 1; 2
2; 3 3; 7 8; 10 10; NODATA NODATA));1 10 1");
Cost_Surface.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Cost_Surface_A")

# Cost Distance
```

```
Cost_Distance = arcpy.sa.CostDistance("DorysFarm", "out_raster", None,
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Direction_Raster", None, None, None, None,
''); Cost_Distance.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\CostDis")

# Cost Path (Best Single)
Cost_Path = arcpy.sa.CostPath("PicnicArea", "CostDis", "Direction_Raster", "BEST_SINGLE", "ID",
"INPUT_RANGE");
Cost_Path.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\CostPath")
```

Part 1.9: Cost Path B

Cost Path C includes reclassified roads (see Table 3), reclassified streams, reclassified NLCD land cover (See Table 7), and reclassified DEM slope (see Table 6). For this optimization, I chose to play around with inputting ‘random’ numbers to see what results I would get.

```
# Reclassify DEM Slope
arcpy.ddd.Reclassify("DEM_Slope", "VALUE", "0 12.552466 1;12.552466 31.381165 1;31.381165 54.394019
2;54.394019 81.591028 2;81.591028 108.788037 5;108.788037 133.892969 10;133.892969 156.905823
10;156.905823 179.918677 10;179.918677 209.207764 10;209.207764 263.601782 10;263.601782 533.479797
11", r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_DEM_2_OP", "DATA")

# Reclassify NLCD Land Cover
arcpy.ddd.Reclassify("Extract_NLCD", "NLCD_Land", "'Open Water' 10;'Developed, Open Space' 2;'Developed,
Low Intensity' 4;'Developed, Medium Intensity' 4;'Developed, High Intensity' 5;'Barren Land' 2;'Deciduous Forest'
6;'Evergreen Forest' 6;'Mixed Forest' 9;Shrub/Scrub 4;Herbaceous 5;Hay/Pasture 10;'Cultivated Crops' 10;'Woody
Wetlands' 7;'Emergent Herbaceous Wetlands' 8",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_NLCD_OP_2", "DATA")



# Cost Surface
Cost_Surface_B =
arcpy.sa.WeightedOverlay(r"('C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_NLCD
```

```
_OP_2' 25 'Value' (2 2; 4 4; 5 5; 6 6; 7 7; 8 8; 9 9; 10 10; NODATA NODATA);
'C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_DEM_2_OP' 25 'Value' (1 1; 2 2; 5 5;
10 10; 11 10; NODATA NODATA);
'C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Raster_C' 25 'Value' (1 1; 2 2; 3 3; 7 7; 8 8; 9
9; 10 10; NODATA NODATA);
'C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_Roads' 25 'Value' (1 1; 2 2; 3 3; 7 7;
10 10; NODATA NODATA));1 10 1");
Cost_Surface_B.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Weighte_Recl_2")

# Cost Distance
Cost_Distance_B = arcpy.sa.CostDistance("DorysFarm", "Weighte_Recl_2", None,
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Backlink_OP_2", None, None, None, None,
''); Cost_Distance_B.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\CostDis_OP_2")

# Cost Path
Cost_Path_B = arcpy.sa.CostPath("PicnicArea", "CostDis_OP_2", "Backlink_OP_2", "EACH_CELL", "ID",
"INPUT_RANGE");
Cost_Path_B.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\CostPat_Picn2")
```

Start	End	New
0	3	1
3	6	2
6	12	1
12	15	1
15	18	5
18	21	10
21	24	10
24	27	10
27	30	10
30	533.47	10

Table 6. Reclassification for DEM Slope for Cost Path B.

Value	New
Open Water	10
Developed, Open Space	2
Developed, Low Intensity	4
Developed, Medium Intensity	4
Developed, High	5
Barren Land	2
Deciduous Forest	6
Evergreen Forest	6
Mixed Forest	9
Shrub/Scrub	4
Herbaceous	5
Hay/Pasture	10
Cultivated Crops	10

Table 7. Reclassification for NLCD for Cost Path B.

Part 1.10: Cost Path C

Cost Path C includes reclassified roads (see Table 3), reclassified streams, reclassified NLCD land cover (See Table 8), and reclassified DEM slope (see Table 6). For this optimization, I chose to focus on creating stark difference in the NLCD land cover scale and used the reclassified DEM slope from Cost Path C.

```
# Reclassify NLCD Land Cover
arcpy.ddd.Reclassify("Extract_NLCD", "NLCD_Land", "'Open Water' 10;'Developed, Open Space' 3;'Developed,
Low Intensity' 3;'Developed, Medium Intensity' 3;'Developed, High Intensity' 3;'Barren Land' 4;'Deciduous Forest'
5;'Evergreen Forest' 5;'Mixed Forest' 6;Shrub/Scrub 5;Herbaceous 5;Hay/Pasture 10;'Cultivated Crops' 10;'Woody
Wetlands' 10;'Emergent Herbaceous Wetlands' 15",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_OP_3", "DATA")


# Cost Surface
```

```
Cost_Surface_C =
arcpy.sa.WeightedOverlay(r"('C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_OP_3'
34 'Value' (3 3; 4 4; 5 5; 6 6; 10 10; 15 10; NODATA NODATA);
'C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Raster_C' 33 'Value' (1 1; 2 2; 3 3; 7 8; 8 9; 9
10; 10 10; NODATA NODATA);
'C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Reclass_Roads' 33 'Value' (1 1; 2 2; 3 3; 7 7;
10 10; NODATA NODATA));1 10 1");



Cost_Surface_C.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Weighte_Recl5")


# Cost Distance
Cost_Distance_C = arcpy.sa.CostDistance("DorysFarm", "Weighte_Recl5", None,
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\BL_5", None, None, None, None, '');
Cost_Distance_C.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\CostDis_Dory8")


# Cost Path
Cost_Path_C = arcpy.sa.CostPath("PicnicArea",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\CostDis_Dory8",
r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\BL_5", "EACH_CELL",



"ID", "INPUT_RANGE");
Cost_Path_C.save(r"C:\Users\gisse015\Documents\ArcGIS\Projects\Lab2_2\Lab2_2.gdb\Cost_Path_3")
```

Value	New
Open Water	10
Developed, Open Space	3
Developed, Low Intensity	3
Developed, Medium Intensity	3
Developed, High	3

Barren Land	4
Deciduous Forest	5
Evergreen Forest	5
Mixed Forest	6
Shrub/Scrub	5
Herbaceous	5
Hay/Pasture	10
Cultivated Crops	10

Table 8. Reclassification for NLCD for Cost Path C.

Results

The results are shown in the figures below (see Figure 3). The main theme of the lab was creating three cost surfaces using different weighting approaches for route optimization. The data flow diagram above (Figure 2 & Figure 3) shows all the variables and commands I applied in finding three optimal routes for Dory.

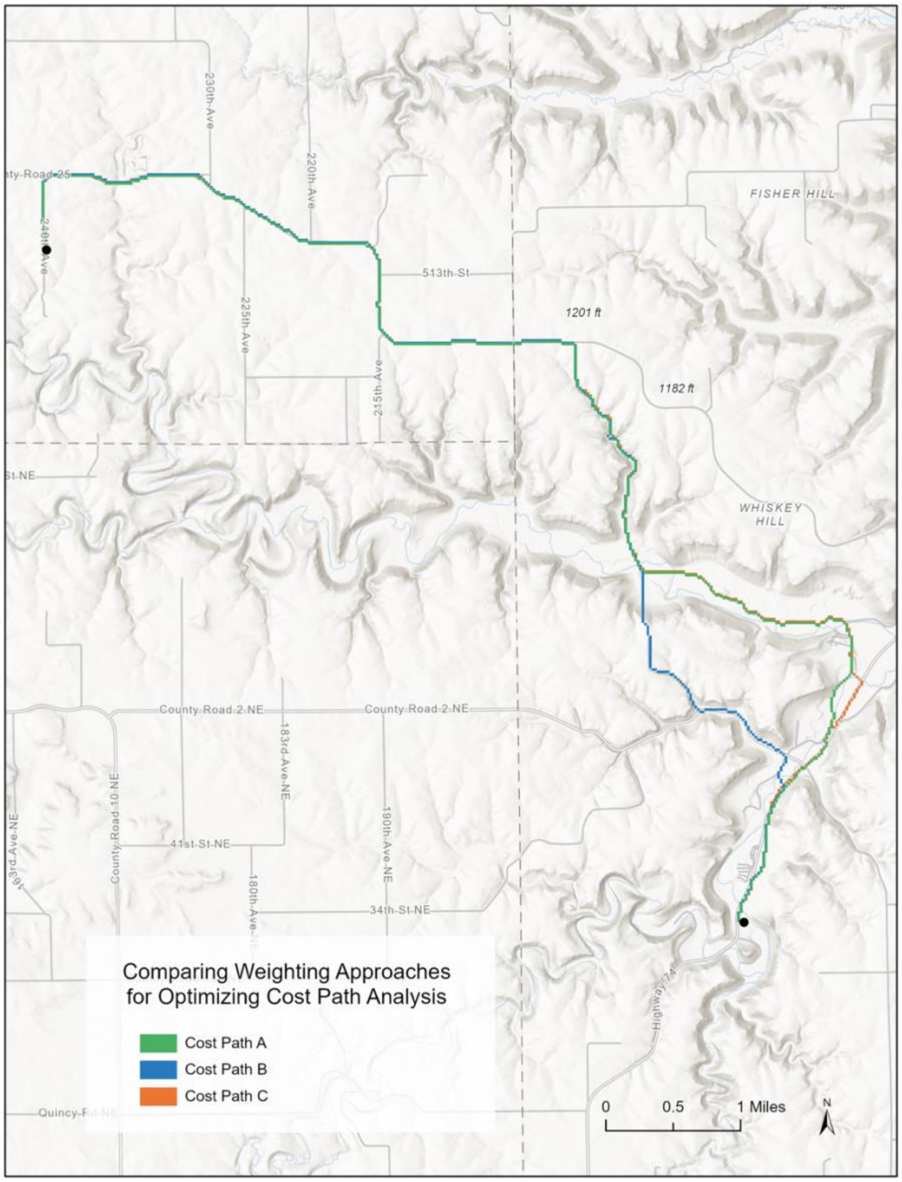

Figure 3. Comparing Weighting Approaches for Optimizing Cost Path Analysis.

Results Verification

The results could be verified by comparing the suggested routes by Apple Maps and Google Maps (shown in Figure 4 & Figure 5). The results (shown in Figure 4 through Figure 6) seem to show accuracy due to fact that the cost surface models are displaying as expected.

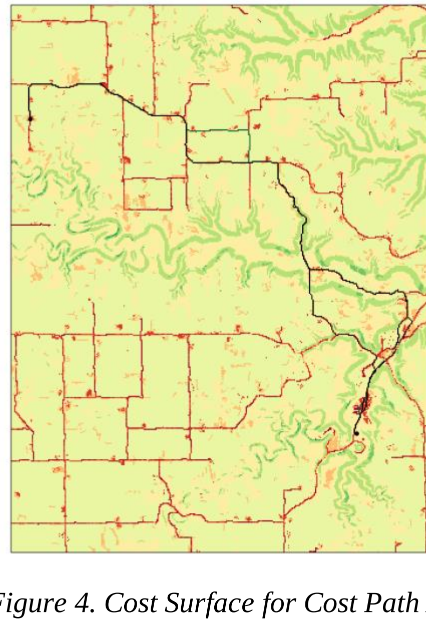

Figure 4. Cost Surface for Cost Path A.

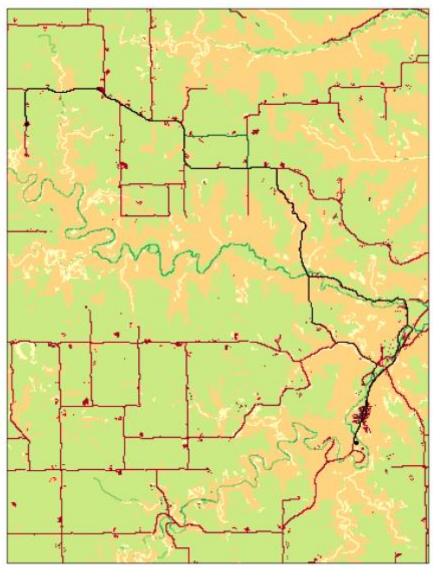

Figure 5. Cost Surface for Cost Path B.

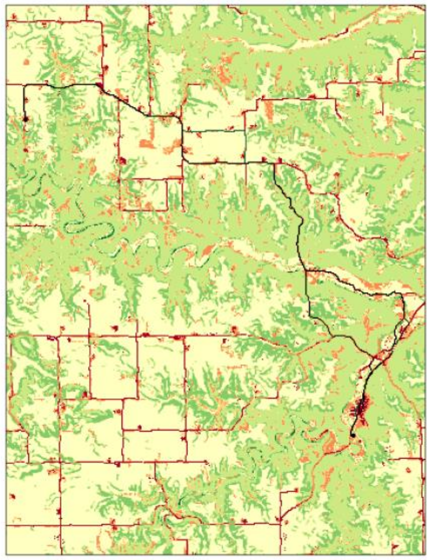

Figure 6. Cost Surface for Cost Path C.

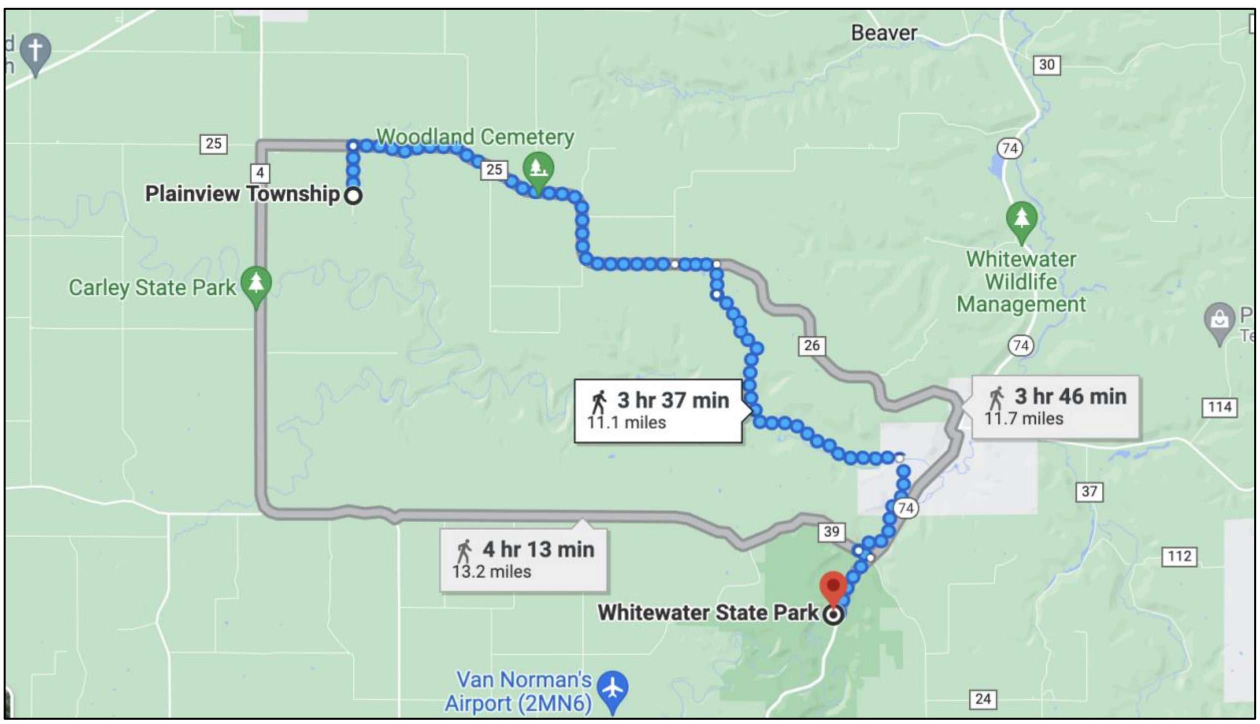

Figure 7. Suggested Routes by Google Maps.

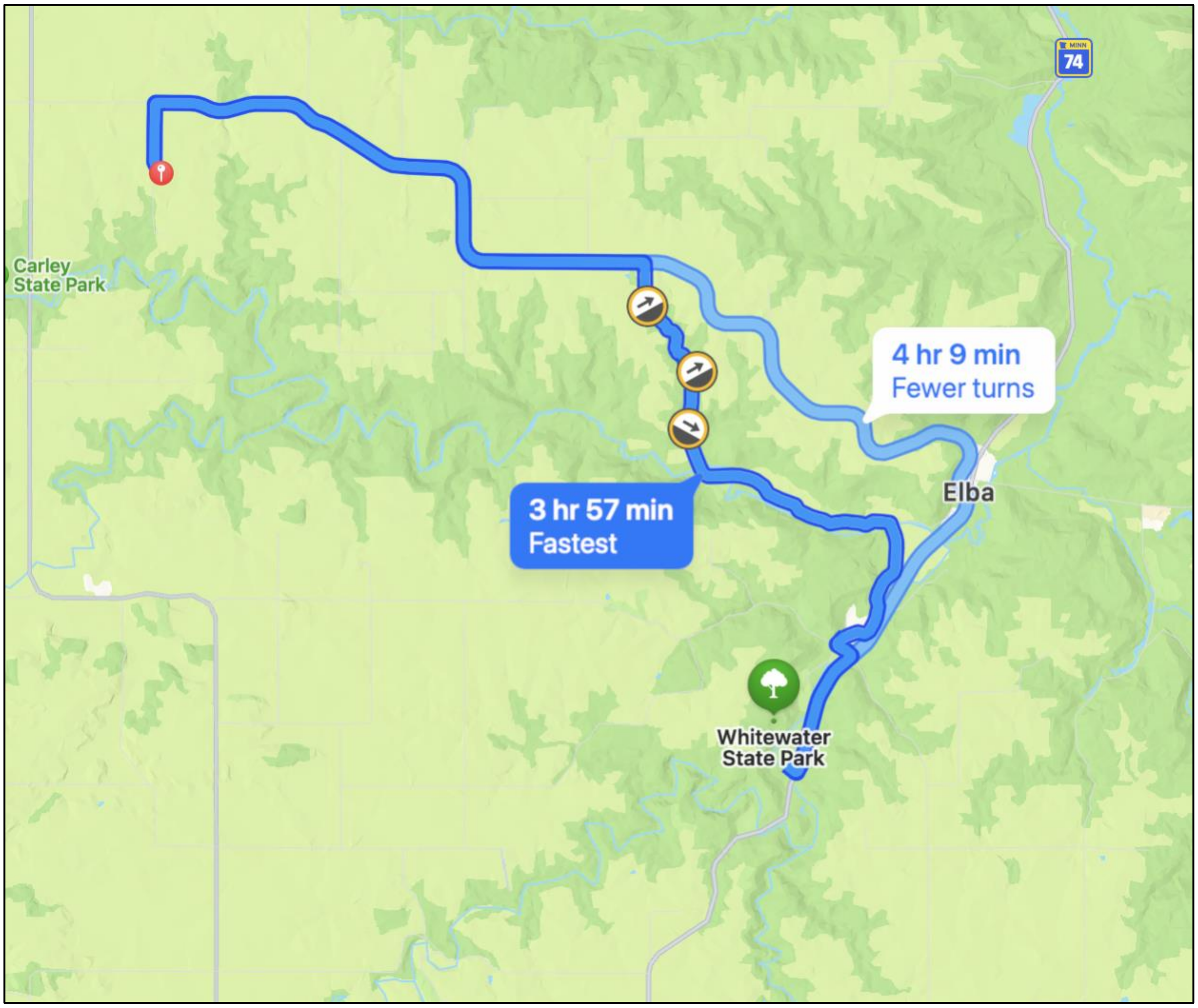

Figure 8. Suggested Routes by Apple Maps.

Discussion and Conclusion

In this lab, I was able to build off Lab 2 with creating an ETL and building a cost path analysis model. The objectives of this lab helped me to gain practical applications of how I would create a cost path analysis through ArcPy or an open-source package. The biggest roadblock was how computationally intensive it was for my computer. In the future, I hope to apply what I learned in this lab to learn how to learn how to code using packages like numpy and rasterio.

References

Runck, B. 2022. GIS 5571: Lab 2. https://github.com/mgisselbeck/GIS5571/blob/main/Lab2/Lab2_Instructions.pdf

Self-score

Category	Description	Points Possible	Score
Structural Elements	All elements of a lab report are included **(2 points each)**: Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score	28	**28**
Clarity of Content	Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**.	24	**24**
Reproducibility	Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified.	28	**27**
Verification	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**.	20	**20**
		100	**99**