Final Project Report:
Finding a Least-Cost Path Through Selected Sample Sites at Joker's Hill

Mike Lackner
Student Number: ### ### ###

GGR 1907
Advanced GIS

December 9, 2004

Prof. Boyes

INTRODUCTION

The purpose of this project was to find random sample points for all clusters of an unsupervised classification and then determine the most time-efficient path through a cost surface, passing all sample points and returning to the starting point. While the statement of the problem sounds fairly simple, solving it turned out to be far less straightforward than expected. Essentially, there were three major tasks to complete—finding random sample points for the unsupervised classification, solving the so-called traveling salesman problem (TSP), and combining the TSP with moving over a cost layer.

It was originally planned to create a fully automated cartographic model with the Environmental Systems Research Institute (ESRI) ArcGIS 9 Model Builder. According to DeMers (2002, 6-8), cartographic modeling is said to be “at the core of GIS modeling” (DeMers, 2002, 6). Therefore, the goal was to build a cartographic model that automated all the parts of this project. However, the limitations of the ArcGIS 9 Model Builder, as well as my limited expertise and time constraints forced me to implement some of the modeling steps manually and perform some of the processes in other GIS software, such as ESRI’s ArcView 8.3 and ArcInfo Command Line. Deviations from the original project proposal to the final model will be explained in more detail in the methodology sections.

Spatial sampling is a part of almost every land cover classification and requires the collection of field points that need to be distributed within the categories to be classified across a study area. These categories or clusters can be derived through an unsupervised classification, which examines the pixels of a rasterized landscape image and groups them together into a pre-defined number of clusters that are based on the spectral values of the pixels (Lillesand and Kiefer 2000, 555-559). In order to find out what these spectral clusters correspond to in reality, in most cases, field visits need to be conducted to collect sample points that represent the individual clusters.

Extensive literature exists on different spatial sampling methods (Rogerson 2001, 58-59; Burrough and McDonnell 1998, 100-101; Jensen 1996, 249-250), but none of the examined material on spatial sampling mentions anything on how to effectively visit the sample points in the field once they are determined, in terms of covering all sample points in the least amount of time. To address this issue, the TSP needs to be included. Sarin et al (2004) define the TSP as finding a tour among a finite number of cities that visits each city once with the cost of travel between each pair of cities given, while minimizing the total travel cost. In the case of this project, the sample points replace the cities that need to be visited.

Adding more complexity to the project, the travel cost between points was not simply the shortest distance between them but the least-cost distance, based on a cost surface or friction layer that was derived from several datasets. Tomlin (1990, 143) defines a friction layer as a layer that expresses the travel cost of each pixel in that layer. Including a cost surface in raster format also meant that simply connecting the sample points through a network of lines did not solve the problem because the values of the raster cost surface needed to be considered for calculating the fastest way that covered all sample points.

It needs to be mentioned that the main focus of this project was to gain experience with creating GIS models and not to produce the most accurate results possible. Therefore, some of the parameters were estimated without verification, and some data were used only because no others were readily available although they might not have been appropriate. Nevertheless, the basic idea of the project can be implemented in other applications if the parameters are sufficiently tested and verified and if the data are appropriately chosen based on the problem to be solved. Details about unverified estimates and inappropriate data will be explained in the methodology section.

DATA & STUDY AREA

Joker's Hill, the study area for this project, is located approximately 25 kilometers north of Toronto, Ontario, and has a size of roughly 3.5 square kilometers. The area consists mostly of forest, open grassy areas, fields, and some water features, as well as roads, trails, and several buildings. Figure 1 shows a color orthophoto with a one-meter spatial resolution and the study area outlined in yellow.

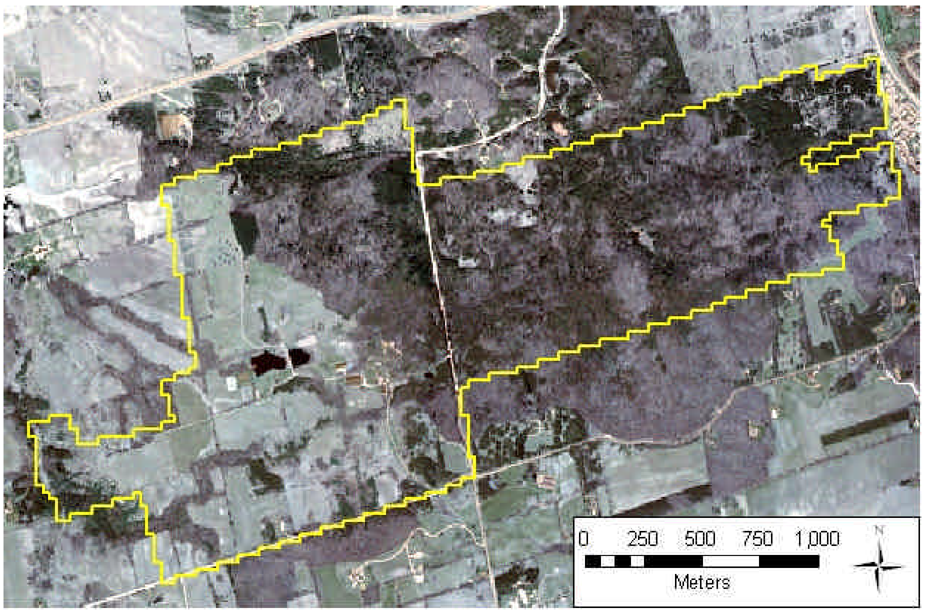

Figure 1: One-meter resolution orthophoto of the Joker's Hill area with the study area outlined in yellow.

The orthophoto was not used for any calculations during the modeling process, but it served as a reference for modifying some of the input files that did not cover the whole extent of the study area.

Before the modeling process, all input layers had to be pre-processed. All layers were projected to a Transverse Mercator projection (UTM Zone 17, NAD 83). The original vector file of the Joker’s Hill boundary was rasterized to a spatial resolution of 30 meters to match the spatial resolution of the unsupervised classification. This raster defined the extent of the study area to which all other layers were then reduced or expanded. The stair-stepped characteristic of the study–area outline in figure 1 is explained by this conversion of the Joker’s Hill boundary to a raster and then reconverting it to a vector file. Table 1 lists all the original datasets used for the project and briefly describes how they were modified for the modeling process.

Table 1: Project data, including data type, spatial resolution (if applicable), and modifications.

Description	Type	Spatial resolution	Modifications
Joker's Hill boundary	Shapefile	N/A	Rasterized to 30m and reconverted to vector.
Landsat TM imagery of Joker’s Hill area	Grid	30 meters	Performed unsupervised classification, clipped to study area.
Roads	Shapefile	N/A	Rasterized to 30m, clipped to study area.
Trails	Shapefile	N/A	Rasterized to 30m, added pixels to cover whole study area.
Forestry Land cover	Shapefile	N/A	Collapsed classes, rasterized to 30m, added pixels to cover whole extent of study area.

The Landsat TM image was used in ERDAS Imagine to produce an unsupervised classification that yielded ten clusters. Using ArcToolbox, the image was imported as a grid into ArcMap, where it was clipped to the extent of the study area (figure 2).

The roads, trails, and forestry land cover shapefiles were all used as inputs for the cost layer. The roads shapefile was converted to a 30-meter grid and clipped to the study area, resulting in a very grainy and unrealistic representation of the roads layer (figure 3). This representation of roads is fairly unrealistic, but it is sufficient enough for this project because the goal was just to find a procedure that can be tailored and applied to other projects.

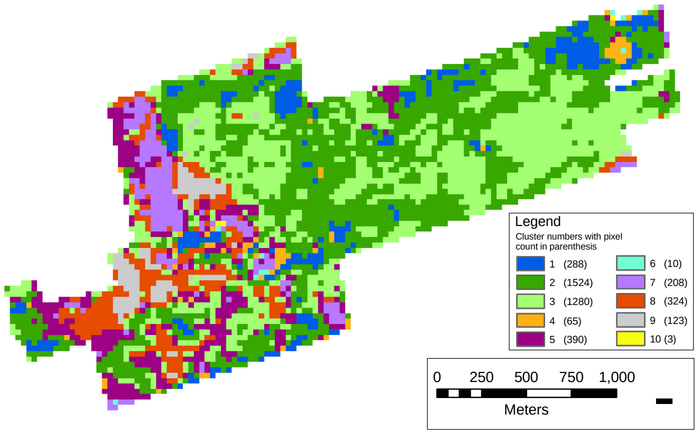

Figure 2: Unsupervised classification with 10 clusters.

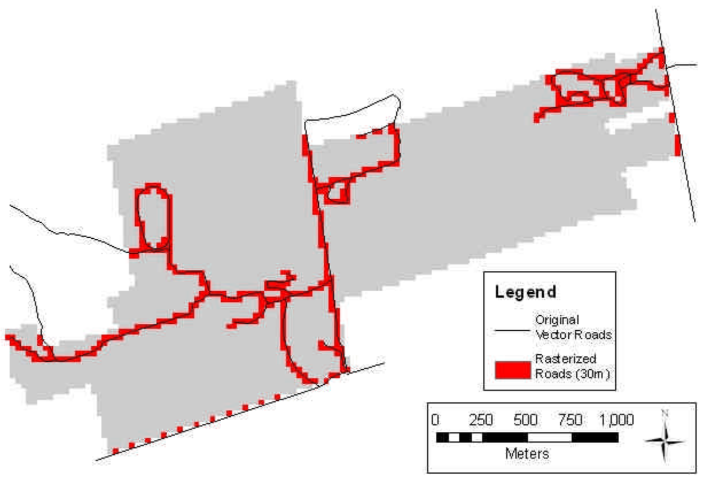

Figure 3: Rasterized roads and original roads, partly outside the study area for better illustration.

A similar unrealistic representation resulted from converting the trails shapefile to a 30-meter grid (figure 4). The rasterized trails layer did not cover the whole study area, so any missing pixels were added based on visual orthophoto interpretation. Many of the trails that are separated in the shapefile are connected in the grid, which creates an unrealistic view, but, as with the roads, it fits the needs for the modeling process.

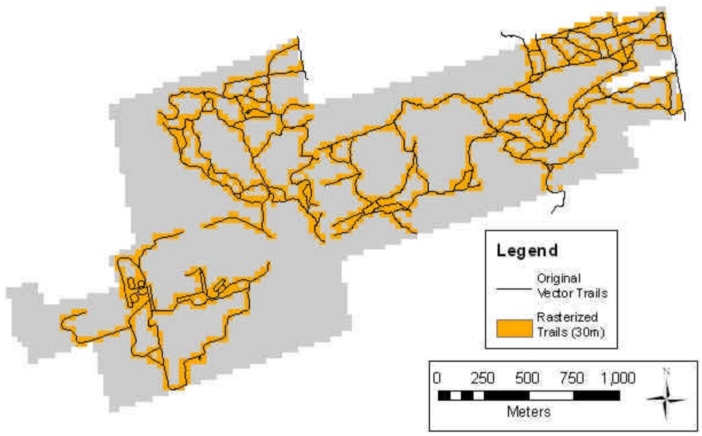

Figure 4: Rasterized trails with original trails superimposed.

The forestry land cover shapefile, created by a former University-of-Toronto graduate student and provided for assignments for this course as well as for this project, was simplified by collapsing several classes in this layer. This was done because the collapsed classes all would have received the same cost value, which will be explained in more detail in the methodology section. The reason why a land cover layer was chosen as an input to eventually find sample points for creating another land cover classification was adding more complexity to the model by including another cost layer. The original forestry land cover shape file consisted of 12 forest classes that were collapsed into only one forest class. Water and bog were combined to water/swamp; barren/scattered and field were combined to grass/field, and the building class remained the same. Because the forestry land cover layer did not cover the entire study area, the rasterized version (figure 5) was modified by adding any missing pixels to the file, based on visual interpretation of the orthophoto, as it was also done with the trails layer.

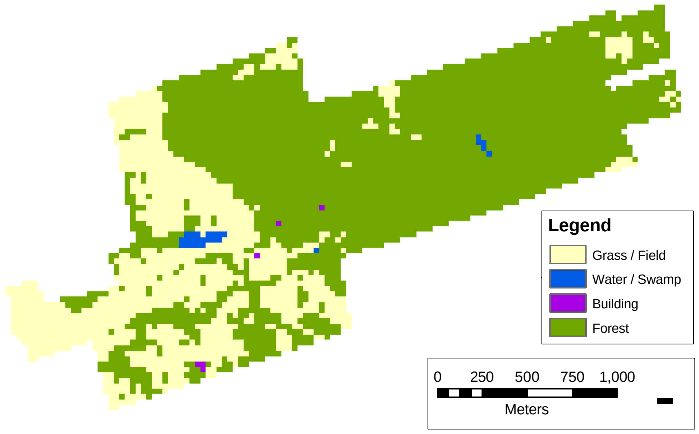

Figure 5: Modified rasterized forestry land cover layer, showing the 4 classes used as cost variables.

METHODOLOGY

Overview: Five-Part Process

As mentioned in the introduction, I originally planned to do all the modeling parts of the project in the ArcGIS 9 Model Builder. Since this was not possible, some parts were performed in other software, as illustrated in table 2.

Table 2: Sequential outline of project tasks, including the software used for each part.

Part #	Task	Software used
Part 1	Generating Sample Points for the Unsupervised Classification Clusters	ArcMap
Part 2	Creating the Cost Surface	Model Builder
Part 3	Creating the Pixel Network	ArcMap, ArcInfo
Part 4	Assigning Cost Values to Network Segments	ArcMap
Part 5	Calculating the Least-Cost Path	ArcView

The detailed flowchart (figure 6) shows how the parts are interconnected. Each part will be explained in detail in the following sections.

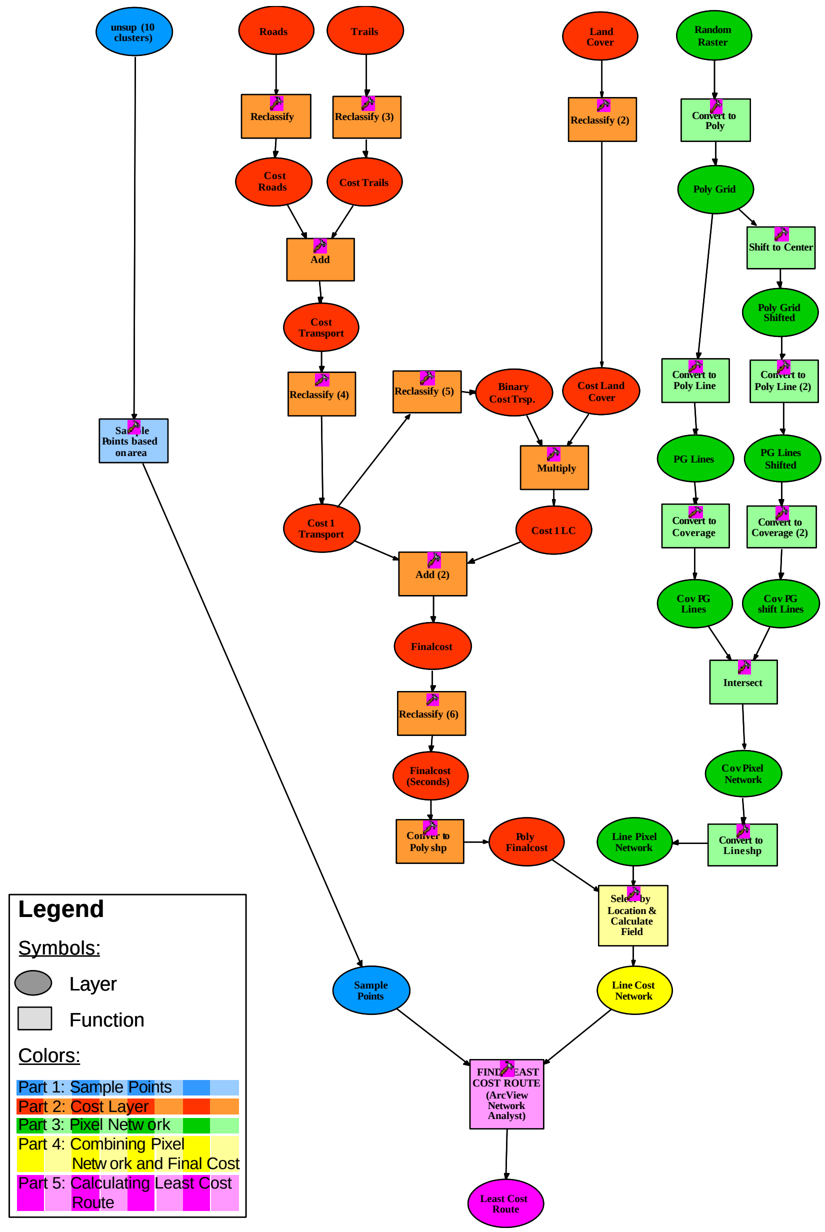

Figure 6: Detailed model flowchart with color-coded parts that were performed separately.

Part 1:Generating Sample Points for the Unsupervised Classification Clusters

When I started with the project, I wanted to randomly select five sample points within each of the ten clusters of the unsupervised classification. The points should also be at least 60 meters (2 Landsat TM pixels) away from each other and in different regions of the unsupervised clusters in order to establish an unbiased sample. After some feedback, rethinking and by looking at the unsupervised classification, I decided to determine the number of sample points for each unsupervised cluster based on their respective amount of pixels, which corresponds to their areas, rather than simply choosing five points per cluster. By looking at the pixel counts, I came up with a rule to determine the number of sample points to be one percent if the pixel count is greater than 1000. If the pixel count became less than 1000, at least ten points should be chosen in order to still have sufficiently enough points. However, if the clusters became too small I looked at their spatial distribution and decided the number of sample points based on what looked appropriate in terms of how connected the pixels of small clusters were. This led to a total of 87 sample points. Table 3 shows how these points were distributed among all clusters in the unsupervised classification.

Table 3: Clusters of the unsupervised classification, ordered by size, with their respective number of sample points.

Cluster Number	Pixel Count	Number of Sample Points
2	1524	15
3	1280	13
5	390	10
8	324	10
1	288	10
7	208	10
9	123	10
4	65	5
6	10	2
10	3	2

With the number of sample points established, they needed to be placed randomly across the different regions of the clusters, considering the pre-decided constraints. I used the region-group function to uniquely identify all the regions of the unsupervised classification and created a random raster, using the Raster Creation Toolbox in Spatial Analyst. That provided me with unique identifiers for each pixel in the study area as well as unique identifiers for each region of each cluster. Based on random numbers, generated from the attribute table of another random raster, I could then randomly choose the regions of each cluster and within each region, the desired number of points. However, I could not model these numbers to be at least two pixels apart from each other.

No solution could be found. Therefore, I created a new shapefile and manually placed the sample points in the regions that I had determined for each cluster, using the editing tools in ArcMap. Although this procedure violated an unbiased sampling selection, it was the only way that let me proceed with the other parts of the project. Figure 6 shows all 87 sample points and how they were spatially distributed across the clusters of the unsupervised classification.

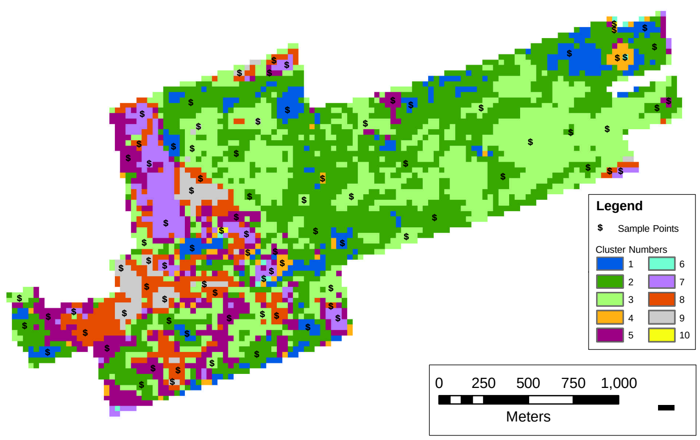

Figure 7: Spatial distribution of sample points within the clusters of the unsupervised classification.

Part 2: Creating the Cost Surface

The cost surface was based on the roads, trails, and land cover raster grids. It was assumed that the sample points would be visited on foot because the study area is small enough to walk through it in a reasonable amount of time, but moreover, the remote areas where some of the sample points were placed can only be reached on foot. Part 2 on the flowchart in figure 6 that represents this part of the model is the only one that was fully automated and executed in the ArcGIS 9 Model Builder. In the subsequent paragraphs, some of the layers will be referred to by the names to as they appear in the flow chart; these names will be put inside quotation marks for clarification.

As a first step, all grids were reclassified based on relative weights, in terms of how long it would take to cross a certain pixel. The idea was that higher weights were assigned to pixels that require longer time to be crossed. Pixels that represent roads received a weight of 1 and all pixels in the roads layer that were not roads received a weight of 8. In the trails layer, trail pixels received a weight of 2 and non-trail pixels a weight of 8. Using an overlay function, these two layers were then added to create unique values for each possible pixel combination between the roads and trails layers. The resulting layer “Cost 1 Transport” was then reclassified in a way that pixels that were identified as road in the roads layer received a value of 1. If the pixel was not road in the roads layer but trail in the trails layer, it received a value of 2, and if it was neither roads nor trail it received a value of 9. This reclassification ensured that road pixels had the lowest cost values, trials the second lowest, and other pixels higher cost values.

To establish the cost values of the remaining pixels, the land cover layer was reclassified in the following way. The grass/field received a weight of 3 because it was assumed it would take only a little longer to cross a grass/field pixel than it would take to cross a trail pixel.

Building pixels received a weight of 4 because they might be less easily accessible than grass/field pixels and therefore require a little more time to cross them. Since it takes even more time to cross pixels of forest and water/swamp, these categories received weights of 5 and 9, respectively. In order to assign the values established in “Cost Land Cover,” “Cost 1 Transport” was used as a mask (“Binary Cost Trsp.”) with zeros for road and trail pixels and ones for all other pixels and then multiplied with “Cost Land Cover.” The resulting layer “Cost 1 LC” then had all the corresponding cost weights for the land cover classes and zeros for all road and trail pixels. By adding “Cost 1 Transport” and “Cost 1 LC,” the cost weights for all pixels in the study area were established.

A reclassification of the resulting layer “Finalcost” from the previous step was performed to assign actual time values to the cost weights. I determined these time values by walking 30 meters (the side of one pixel) in a hallway and measuring how many seconds it would take. The 15 seconds I measured served as the base value that I assigned to the road pixels because I concluded it would take about the same time to walk 30 meters along a road as well as in the hallway I walked upon. The cost values for trails, grass/fields, buildings, forest, and water/swamp were then just estimated because it would have been too time consuming to verify how long it would really take to cross a pixel for all the categories. The cost values for all categories are listed in table 4.

Table 4: Cost values (in seconds) for all categories in the “Finalcost (Seconds)” layer.

Class	Cost (in seconds per pixel)
Road	15
Trail	25
Grass/Field	35
Building	45
Forest	60
Water/Swamp	120

Since “Finalcost (seconds)” is a raster grid, but the subsequent parts of the model required vector files as inputs, the layer was converted to a polygon shapefile (figure 8).

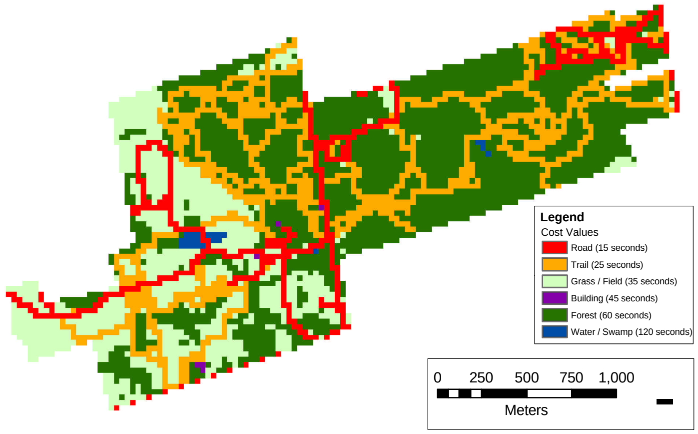

Figure 8: Cost values (in seconds) for all categories in “Poly Finalcost.”

Part 3: Creating the Pixel Network

After unsuccessful attempts to find the least-cost route between all the sample points with the distance functions in Spatial Analyst, I decided to try solving the problem with the Network Analyst extension in ArcView. The problem was that the cost layer was a raster grid converted to a vector polygon file, but the Network Analyst needed a vector line file to calculate a least-cost path through all sample points. Therefore, the goal was to create a “pixel network” that allowed for moving through the pixels in the cost layer through a vector network while also retaining the cost values.

Part 3 in the model flowchart (figure 6) illustrates step-by-step how the pixel network was created. First, a random raster that covered the study area was converted to a polygon shapefile. Using the ArcMap editing tools, a copied version of that output was then shifted 45 degrees in a diagonal direction, half way through a square that represented a 30-meter pixel. This shifted polygon layer was the basis of the pixel network to be created. In order to further prepare these files, they were converted to poly line shapefiles and then to coverages. In ArcInfo Grid, the arc attribute tables were built for both line coverages, and the arcs in the shifted coverage were selected before the two coverages were intersected to create all necessary nodes and vertices for the pixel network. This resulted in a coverage that only had the lines from the shifted line coverage, including vertices at the intersections of the original and shifted line coverages that were located exactly at the boundaries between pixels (figure 9).

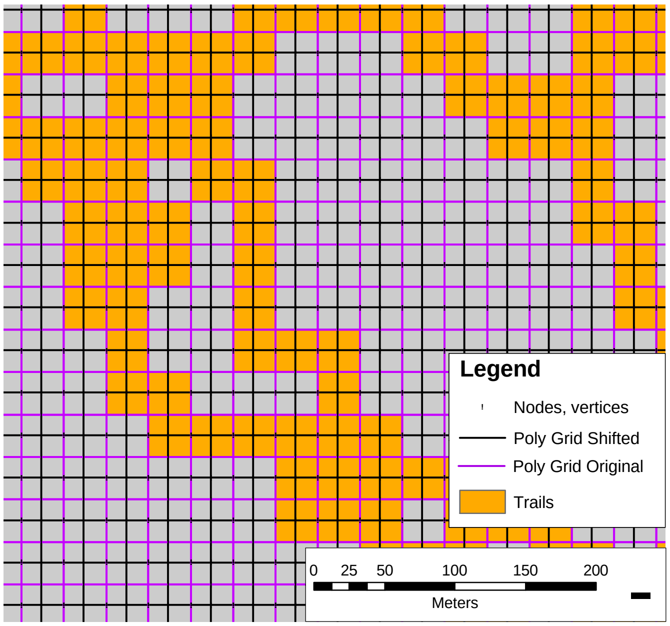

Figure 9: Shifted and original line coverages for pixel network with trails grid as background to illustrate the relation between line segments and pixels.

The shifted line coverage was then converted to a shapefile, named “Line Pixel Network,” and served as the network that had yet to be assigned the cost values. Each pixel contained four line segments, but only two segments needed to be traveled upon, in order to cross a 30-meter pixel in the cost layer.

Part 4: Assigning Cost Values to Network Segments

After adding a field named "Seconds" to the shapefile "Line Pixel Network," I used the Select By Location tool in ArcMap to select segments from the pixel network that were within a certain cost polygon in the "Poly Finalcost" shapefile. Figure 10 shows how the segments within the cost polygon of 15 seconds, representing roads, were selected.

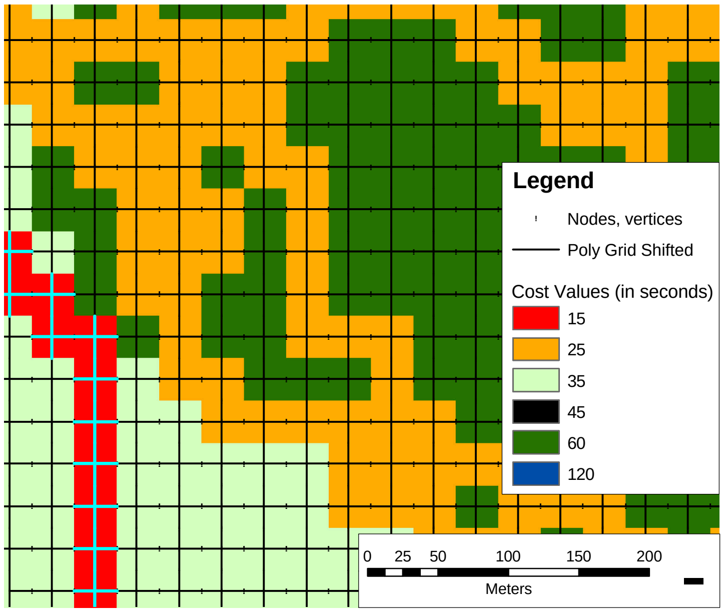

Figure 10: Cost values assigned to selected segments (in cyan) for roads (15 seconds) to the pixel network.

Having these segments selected, the "Seconds" field was calculated by assigning a value of half the cost that it takes to cross the corresponding cost pixel, which was 7.5 in the case of roads pixels. Half the cost was used because only two segments need to be crossed to cross a pixel. This process was repeated for all other cost polygons until all segments in the pixel network were assigned their respective cost values. The file was then called "Line Cost Network," as indicated in the flowchart (figure 6).

Part 5: Calculating the Least-Cost Path

The last part of the model was performed, using the Network Analyst extension in ArcView, and it combined parts 1 and 4. "Sample Points" and "Line Cost Network" were used as the inputs for the Find Best Route function in Network Analyst, which solves a TSP. By default the shortest path is calculated, so the cost field for "Line Cost Network" needed to be changed to "Seconds" to calculate the best route based on the time in seconds and not based on total length. A starting and end point for the route was manually placed with the Add Location button in Network Analyst. The field house at Joker's Hill, as interpreted on the orthophoto, was chosen to be that point because it seemed to be logical to start the route for the sample point visits at a place that can serve as a base station. With the addition of the Joker's Hill field house, 88 points needed to be visited in the least-time-consuming order. The path that was calculated by the Network Analyst with the cost layer as a background is shown in figure 11.

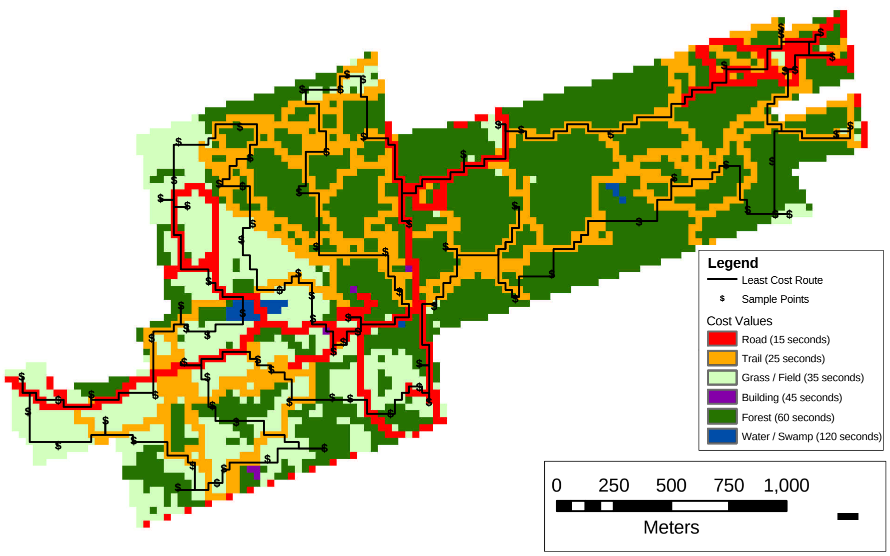

Figure 11: Least-cost route connecting all sample points at Joker's Hill.

RESULTS & DISCUSSION

The time it takes to visit all points and return to the origin came out to be 22,911.35 seconds (6 hours, 21 minutes, 51.35 seconds), as displayed in the attribute table of the shapefile “Least Cost Route.” Of course it needs to be considered that visiting sample points requires to stop at the points to be visited, so additional time needs to be added accordingly. This can vary from one project to another, depending on if there are any measurements or photographs to be taken and so on. Determining the time for such stops is beyond the scope of this project and will therefore not be discussed.

While the main goal of finding the least-cost route among sample points was found, there are still many things that can be improved in the model, given more time to develop and explore more efficient techniques. Generally, more parts of the model could be directly integrated into the Model Builder and automated. For example, some scripts could be written to truly randomly select the sample points and still have the originally planned distance constraint implemented. The cost layer could also be improved by verifying the times it takes to cross a certain cost pixel. Moreover, the spatial resolution of the cost layers could be improved to establish more realistic results. This could be achieved by resampling the cost layers to a resolution of one meter, so that even narrow trails are accurately represented. Of course, all other layers would have to be resampled to the same resolution as well, which could possibly lead to problems with computation time, especially when creating the pixel network with such a high resolution. This also leads to the question of how feasible the methodology with the pixel network would be in a larger study area and many more sample points. Only trial and error could give an answer to this question. Creating the pixel network and combining it with the cost layer maybe could have been done within the Model Maker if a way was found to shift the original vectorized random raster within the framework of the Model Maker.

Although the current model might have several limitations and unrealistic components, the general idea could likely be expanded and applied to different projects. For example, a larger area with more sample points could be traveled upon by car and on foot. The cost layers in such a scenario would then be a detailed road layer and costs could be associated with speed limits. Possible off-road sample points could again be associated with a trail layer and any other data that can be related to travel time. A problem might again be how realistic and feasible the model would be in terms of spatial resolution.

Throughout this project it was interesting to find out how easily one can reach a dead end that required redesigning the model that was thought to be as simple as the problem was stated. While it might be possible to accomplish the original goal of creating a fully automated model with the Model Builder with sophisticated scripts that possibly involve some programming, it is maybe often easier to resort to different software that does exactly that part that is needed in an efficient way. This may often be easier than trying to stick with only one software package that may need to be customized in many ways in order to solve the given problem.

REFERENCES:

Burrough, P.A. and R.A. McDonnell. 2004. Principles of geographic information systems. Oxford University Press. Oxford, UK.

DeMers, M.N. 2002. GIS modeling in raster. John Wiley & Sons, Inc. New York, NY.

Jensen, J.R. 1996. Introductory digital image processing: a remote sensing perspective. Prentice Hall. Upper Saddle River, NY.

Lillesand, T.M. and R.W Kiefer. 2000. Remote sensing and image interpretation. John Wiley & Sons, Inc. New York, NY.

Rogerson, P.A. 2001. Statistical methods for geography. Sage Publications Ltd. London, UK.

Sarin, S.C. et al. 2005. “New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints.” Operations Research Letters. Vol. 33, 1: 62-70.

Tomlin, C.D. 1990. Geographic information systems and cartographic modeling. Prentice Hall. Englewood Cliffs, NJ.