San Jose State University

SJSU ScholarWorks

Master's Theses Master's Theses and Graduate Research

Fall 2015

Seasonal Road Layout Design In Mountainous Terrain Using GIS With The Side Hill And Least Cost Path Methods

Emil Harold Brundage
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation
Brundage, Emil Harold, "Seasonal Road Layout Design In Mountainous Terrain Using GIS With The Side Hill And Least Cost Path Methods" (2015). *Master's Theses*. 4626.
DOI: https://doi.org/10.31979/etd.uud3-7eh8
https://scholarworks.sjsu.edu/etd_theses/4626

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

SEASONAL ROAD LAYOUT DESIGN IN MOUNTAINOUS TERRAIN USING GIS WITH THE SIDE HILL AND LEAST COST PATH METHODS

A Thesis

Presented to

The Faculty of the Department of Geography

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

by

Emil H. Brundage

December 2015

© 2015

Emil H. Brundage

ALL RIGHTS RESERVED

The Designated Thesis Committee Approves the Thesis Titled

SEASONAL ROAD LAYOUT DESIGN IN MOUNTAINOUS TERRAIN USING GIS
WITH THE SIDE HILL AND LEAST COST PATH METHODS
by

Emil H. Brundage

APPROVED FOR THE DEPARTMENT OF GEOGRAPHY

SAN JOSÉ STATE UNIVERSITY

December 2015

Dr. Gary Pereira	Department of Geography and Global Studies
Dr. Kathryn Davis	Department of Geography and Global Studies
Dr. Kerry Rohrmeier	Department of Geography and Global Studies

ABSTRACT

SEASONAL ROAD LAYOUT DESIGN IN MOUNTAINOUS TERRAIN USING GIS WITH THE SIDE HILL AND LEAST COST PATH METHODS

By Emil Harold Brundage

Seasonal road design in mountainous terrain consists of four main phases: route selection, field investigation, surveying, and analysis. The first phase, route selection, consists of two parts: selecting control points at strategic locations, and then determining potential routes between those points. Two geographic information system (GIS) geoprocessing automations were developed to aid a road planner in determining routes between control points. Both automations utilized Environmental Systems Research Institute's (ESRI) ArcGIS software package. The first method developed was the least cost path method, which makes use of ArcGIS's cost path tool to find a route between points following a gradual slope. The second automation was the side hill method, which utilized a variety of ArcGIS tools to maintain a uniform grade along the side of a hill between two points. The two methods were compared and contrasted based on control point locations. The least cost path method was determined to be preferable for main thoroughfares along flat valleys and ridge lines, while the side hill method was preferable for secondary roads that could be used to access steeper ground. It was concluded that the two methods can save time and increase accuracy of GIS road features for land managers planning new seasonal roads.

ACKNOWLEGEMENTS

I would like to thank the following people for their help and support: to Cassady Bill Vaughan and Ed Struffenegger, for their knowledge and insight into road design and layout; to Richard Taketa, for teaching me so many of the skills I employed to create my GIS methods; to my thesis committee, Kathryn Davis, Gary Pereira, and Kerry Rohrmeier, and finally, to Mom and Dad, for their support on this project and so many others.

Table of Contents

List of Tables viii

List of Figures ix

Introduction 1

Road Design in Mountainous Terrain 1

Sediment and its Effects 2

Road Design Methods to Mitigate Sedimentation 5

Fish Bone Road Layout 7

Geographic Information Systems and Road Planning 8

Objectives 11

Methods 12

ESRI/ArcGIS 12

GIS Data 12

Python 13

Least Cost Method 14

Side Hill Method 20

Discussion 32

Implementation 32

Limitations 34

Design Alternatives 35

Conclusion 38

References...39

List of Tables

Table 1. Slope classification for the least cost path method 18

List of Figures

Figure 1. An example of fish bone road layout, with a primary road along a ridge line (red), and secondary roads branching out to the sides of hills (blue)..................8

Figure 2. GIS raster (purple and yellow shading) and vector (white contour lines and green polygon) data. 13

Figure 3. The side hill method script loaded into IDLE. 14

Figure 4. The USGS Earth Explorer web site, a source for DEM files. 16

Figure 5. Start and end point features placed on a digital elevation model raster. 17

Figure 6. The printed output of the least cost path method. 20

Figure 7. A start point (green dot), the point that intersects the end point's contour (blue triangle) and the line between the two that will be used to clip the contours. 25

Figure 8. Contours between start and end points (purple lines) used to generate a side hill line feature...27

Figure 9. The selected segment of the first contour, starting at the starting point, and its connecting line to the next contour (red line)..29

Figure 10. A series of contour lines and steps (red line). 30

Figure 11. The final line feature (red line) from start point to end point.........................31

Figure 12. An example of the side hill method output (blue lines) and the least cost path method (green lines) used together to create a road network. 33

Figure 13. An example of a side hill method line feature (A), and the same line after the *smooth line* tool has been applied to the feature with different tolerances (B and C)..36

Introduction

Road Design in Mountainous Terrain

New seasonal road design and implementation is an integral component of effective forest land management in mountainous terrain. A variety of situations can justify road construction, including the need to access timber for harvesting, the need to access newly acquired land, the need to join previously disconnected areas, and the need for new routes after old routes become unusable due to natural degradation.

Seasonal road layout and design is a process that includes four main phases, as described by the British Columbia Ministry of Forests (2002): route selection, field investigation, surveying, and analysis. Route selection generally takes place in an office environment so as to limit extensive field survey efforts through the elimination of unsuitable areas and identification of areas highly suitable for new roads. During the route selection phase, the road planner studies various information sources, including aerial imagery and topographic maps, boundary maps, areas of concern maps (showing unstable areas, culturally significant areas, etc.), and maps showing existing roads and waterways.

A road planner in the route selection phase first determines control points that will strongly influence road locations. Control points can include:

- Bluffs, benches, passes, saddles, and stream crossings
- Switchback locations
- Harvest landings

- Haul waste areas
- Man-made structures such as power lines, pipelines, and railways
- Existing access points

Once control points have been established, a road planner must determine the best routes between points. With other considerations such as ecologically sensitive areas and road networking in mind, a main goal in the route selection phase is finding the route with the consistently smallest incline, or grade, feasible (USDA Forest Service, 2015). Road grade is usually expressed as the quotient rise (y) over run (x), although a measure of degrees may also be used. Mild road grades are beneficial for a number of reasons, such as ecological stability and economics.

Sediment and its Effects

A primary concern for rangeland managers considering new road implementation is minimization of sediment runoff. Sediment is the fine-grained material broken down by natural processes, such as chemical and mechanical weathering, and erosion. Sediment may then be transported by wind movement, water, ice, or gravity. Sedimentation of waterways is a naturally occurring process. New road construction in mountainous terrain, however, can expose large amounts of settled sediment, previously held in place by vegetation, to these natural forces. These forces can increase sediment dispersion to degrees that otherwise would only rarely occur naturally. In such a situation, sediment often finds its way into creeks and streams, causing abnormally high concentrations of fine suspended particles.

Salmon are one species dependent on river environments for their anadromous lifecycle. These fish are affected by high sediment concentrations within those rivers throughout the extent of their lifecycle, starting when they hatch from eggs laid in freshwater rivers by the previous generation. Upon birth, young salmon swim downstream until they reach the ocean, where they spend most of their lives in a marine environment, only to return to riverine systems to spawn.

Salmon become an important species for maintaining ecological stability in the freshwater ecosystems they inhabit (Naiman, Bilby, Schindler, & Helfield, 2002). Salmon die and biodegrade after spawning, releasing the nutrient abundance they accumulated during time spent at sea. A wide range of predators feed on salmon during their spawning run, including bears, birds of prey, sea lions, otters, and seals.

Sedimentation of rivers and streams can negatively affect salmon in a number of ways during their journey upriver to spawn. Along the way, salmon must rest in pools formed by large debris, such as rocks and fallen logs. High sediment concentrations can fill in these pools, rendering them useless (Frissell, 1992). Knopp (1993) found that up to 42% of a river's pools can fill in with sediment in managed watersheds, which is significant when compared to unmanaged watersheds, where only 17% of rivers' pools were found to fill with sediment.

Another way in which salmon are negatively affected by river sedimentation is during the egg-laying period. Salmon construct nests, called redds, in which they lay their eggs. Redds, which can be 5 to 50 centimeters deep, are constructed along the riverbed (Evenson, 2001). Redds require clean water flowing through this gravelly environment to supply oxygen and prevent the accumulation of waste and debris (Silver, Warren, & Doudoroff, 1963; Daykin, 1965; Turnpenny & Williams, 1980). Salmon reduce the level of sediment around their redds as they construct them, clearing the area with their tails and body.

Once hatched, salmon enter their alevin phase, in which the young (around 2.5 centimeters in length) reside between small rocks on the river floor. The alevin sustain themselves and grow larger by consuming nutrients from their egg yolk sac. During this time, increasing sedimentation can significantly increase mortality rates (Nawa & Frissell, 1993).

After depleting their yolks, salmon begin their journey downstream towards the ocean. This is known as the fry stage, when their small size allows salmon to hide from predation between rocks of the gravelly river beds. Aquatic invertebrates also reside in this area and serve as the primary food source for salmon fry. Increased stream and river sedimentation fills in spaces between gravel with particles, removing the salmon fry's cover and food source (Suttle, Power, Levine, & McNeely, 2004).

Salmon are not the only species negatively affected by steam and river sedimentation. Newlon and Rabe (1977) found four factors that affect macroinvertebrate biodiversity: water temperature, gradient, suspended sediment, and substrate material. Lemly (1982) also found that macroinvertebrates are negatively affected by habitat reduction and habitat change brought about by sedimentation. Suspended sediment was found to cause a lowered respiration capacity in various insect species. Feeding, especially for those macroinvertibrates that primarily feed through filtering, was found to be negatively impacted due to suspended sediment.

Welsh and Ollivier (1998) found that amphibians were also negatively impacted by high sedimentation rates. Researchers studied five stream channels affected by high sediment loads from a nearby highway construction project after a heavy rain event. The three most abundant amphibians were studied: the Pacific giant salamander, the tailed frog, and the southern torrent salamander. For all three species, population densities in the amphibian's larval and adult stages were negatively impacted by increased sedimentation, and the relative amounts of gravel and cobble were the best predictors for the Pacific giant salamander abundance. Tailed frog abundance was also strongly associated with the relative amount of coarse substrates such as cobble. The southern torrent salamander's abundance rates followed the same pattern, with a decrease in abundance in streams containing high concentrations of fine sediment. The researchers attributed these findings to the lack of protected habitat, as well as diminished food sources.

Road Design Methods to Mitigate Sedimentation

Seasonal roadways have been found to be the highest anthropogenic contributor to stream sedimentation (Anderson & Potts, 1987; Beschta, 1978; Dunne & Dietrich, 1982; McChashion & Rice, 1983; Megahan & Kidd, 1972; Reid & Dunne, 1984; Rice, Tilley, & Datzman, 1979; Swanson & Dyrness, 1975). Proper road construction and maintenance, however, can drammatically reduce the amount of sediment that reaches waterways (Reid & Dunne, 1984). Some tools at a road designer's disposal to help mitigate sediment runoff from roads, as listed by the British Columbia Ministry of Forests (2002), are:

- Road materials, including rocking and matting
- Water bars
- Silt fencing, hay bales, and revegetation mats
- Rock check dams or rip rap
- Sediment catchment basins
- Roadside revegetation
- Culverts
- Streamflow diversion and impoundment

The need for increased sediment mitigation is directly correlated to road grade. The amount of sediment transported due to erosion as a result of water runoff increases as grade increases. Roadway channeling occurs on unpaved roadways when sediment is carried off of the road by water movement during and after rain events. Channeling is dependent on three factors: drainage area, soil type, and grade (Montgomery, 1994). Since soil type in relation to sediment runoff potential can often be consistent throughout a region, drainage area and grade are the main factors to consider in sediment mitigation. As the grade of the road increases, drainage and the distance water travels along the road must be reduced to compensate. Reducing the distance water travels may be achieved through the mitigation methods listed above, such as the application of road materials and water bars. These methods reduce the distance water travels, which reduces the water's kinetic energy. These measures reduce channeling potential and minimize water’s capacity to transport sediment.

Since steeper roads require increased mitigation, road planners have not only an environmental but also a financial incentive to plan roads with gradual slopes (Bowers & Garland, 2012). Steep roads also come with the external cost of reduced traffic speed. If a road is being planned for harvesting use, this traffic reduction can increase operation times, which in turn increases operational costs.

Fish Bone Road Layout

A common method of road layout is the fish bone design. The fish bone design consists of a main roadway along a ridge or valley intended to carry most traffic. Secondary rib roads branch off from the main road and allow for access to hillsides and other main roadways. The fish bone method allows for the majority of traffic to be routed along flat, wide stretches of land, with minimal impact on steeper areas.

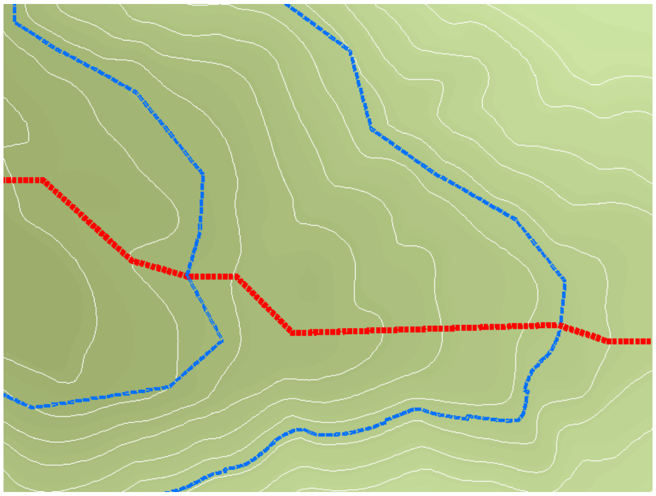


Figure 1. An example of fish bone road layout, with a primary road along a ridge line (red), and secondary roads branching out to the sides of hills (blue).

Geographic Information Systems and Road Planning

Creating geographic information system (GIS) features of potential roads is helpful in road building operations, as it tells the planner which areas are accessible and which are not, as well as the distance of road required for an operation. The incorporation of GIS technologies into road design has continued to expand, but there are no modern tools available to aid planners with specific layout locations during the route selection phase. Gumus, Acar, and Toksoy (2007) developed a GIS method for evaluating road layouts based on area boundary, timber volume, available infrastructure, topographic structures, existing roads, water resources, forest structure, and local work opportunities. These data sets were combined to form an environmental impact assessment map, which indicated the extent of environmental effects of new roads in a given area. Such GIS analysis would be beneficial to a road planner for choosing control points, but would not aid in the placement of roads between control points.

Stuckelberger, Heinimann, Chung, and Ulber (2006) evaluated road location models based on construction and maintenance costs, negative ecological effects, and cable yarding landing suitability. Newnham (1995) developed a model that creates a road network based on harvesting priority raster data. His method generates harvest blocks with straight lines between points that indicate the optimum road layout. Newnham's model may be most helpful to forest land managers deciding on road control points. Picard, Gazull, and Freycon (2006) compared multiple target numerical optimization techniques that generated road networks based on building costs. Like Newnham's model, Picard's research would aid in route selection, but offers only a general location for road placement. Finally, Jusoff (2008) developed a GIS-based decision support system that optimized sustainable timber harvesting in the selection of new forest road locations. Jusoff's findings, like those of other researchers, could prove useful to road planning in certain situations, but may not be broadly applicable to general land managers.

For a time, PEGGER, developed by Rogers and Schiess (2001), was perhaps the most accessible GIS tool to aid in the creation of road locations between control points. PEGGER is an ArcView GIS extension that emphasizes a simple, intuitive user interface. The software makes use of contour data. To use the software, a user initially inputs a desired grade, a starting location, and the desired road direction. PEGGER then generates lines from the starting location to consecutive contour lines at the indicated grade and in the desired direction. With each click a user creates a new line between contours. The software allows for grade increase or grade decrease at any time. A user can estimate the grade needed to reach an end point, and quickly start again if the estimated grade proves insufficient by repeating the process until a desired road location is achieved. While PEGGER's design allows for simple, wide-ranging uses by road planners, the software is out of date. PEGGER is compatible with Environmental Systems Research Institute (ESRI) ArcGIS ArcView 3.2, but cannot be installed on the current version of ESRI ArcGIS. The lack of an updated version for this ten year old software severely limits is current application.

Without GIS tools easily available for automated creation of roads between control points, a planner must create features manually. For roads along ridges and valleys, creating the road line features is time consuming. The planner creates lines along crests and valleys with the aid of contour lines. The manual process is often repetitive, leading to mistakes in placement and features not lining up properly along the flattest path. Roads manually placed along the sides of slopes can also suffer from precision errors. Maintaining a constant slope involves mathematical calculations that inform the planner how long a road segment should be between contours. Small distance errors compounded over several segments can lead to a poor outcome and force the designer to make multiple attempts at a single feature.

Objectives

The objective of this thesis is to provide a modern software tool that increases precision and efficiency of preliminary route selection in mountainous terrain. The tool needs to complement other methods of preliminary road planning, such as making use of aerial photography, paper-based maps, and other GIS data and analysis. The tool should be accessible and compatible with widely used GIS software. Finally, this tool should yield the best location for a road as its output, with the most gradual slope.

Methods

ESRI/ArcGIS

ArcGIS, developed by ESRI, is currently the most popular GIS software available. A study published by the ARC Advisory Research Group (ESRI, 2015) reported that ESRI holds a 43% share of the GIS market while the next-largest developer holds only 11%. ESRI, established in 1969, produces a variety of GIS software packages, with its main product offerings being ArcGIS for Desktop, ArcGIS Online, and ArcGIS for server.

GIS Data

There are two main categories of GIS data: vector and raster. Vector data represent information using points, lines, and polygons. Vector data are most useful for representations of features that have distinct boundaries, such as political boundaries or road networks. In the ArcGIS environment, points, lines, and polygons are grouped into feature classes, with each point, line, or polygon being referred to as a feature. Raster data consist of a cellular matrix organized into rows and columns. Each cell has a value assigned to it, representing a measurement within its area. Some examples of raster data are digital pictures, scanned maps, aerial photography, and satellite imagery. Both raster and vector data can have information related to each feature. These data are stored in an attribute table.

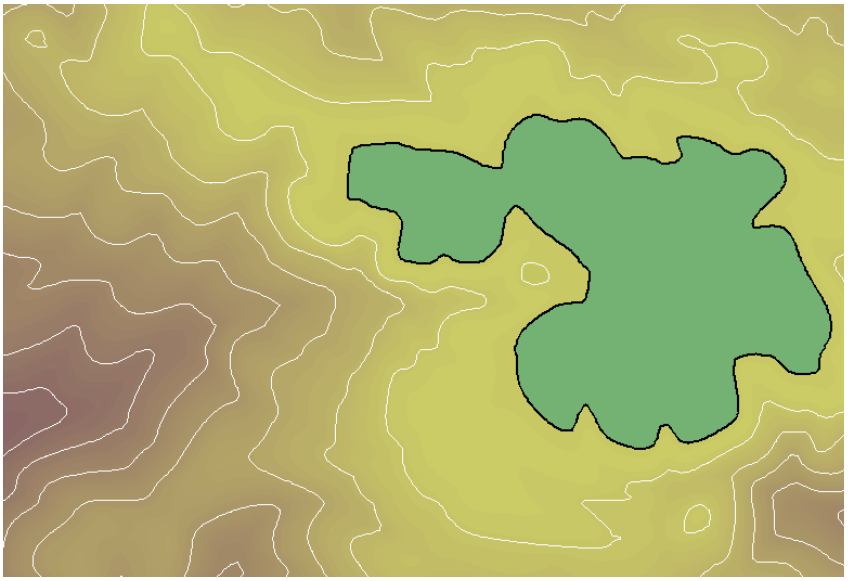


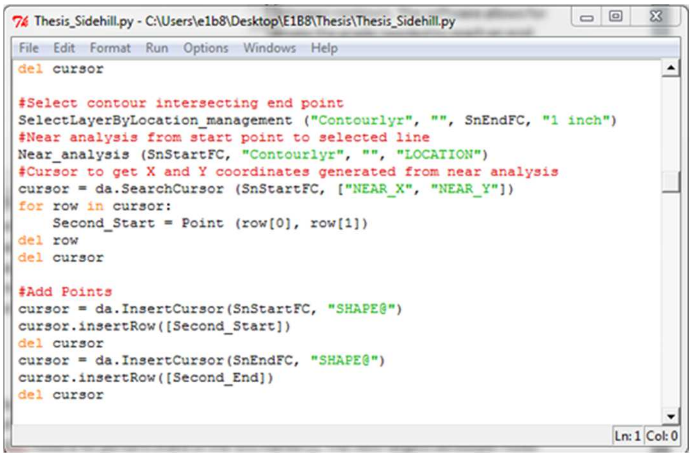
Figure 2. GIS raster (purple and yellow shading) and vector (white contour lines and green polygon) data.

Python

Python is a popular programming language that emphasizes concise methods. Python generally requires fewer lines of code than other languages. As of October 2015, Python was the fifth most popular programming language in use, with its popularity continuing to rise (TIOBE, 2015). Python functionality was introduced into ArcGIS in May 2004 with the release of ArcGIS 9.0. ArcPy is ESRI's Python module and allows ArcGIS functionality to be easily incorporated into a Python script. Road prediction automations for this thesis were developed with ArcGIS 10.1, which has Python 2.7 and ArcPy as part of its standard installation.

Least Cost Method

Two methods were created for predicting road placement. The first method makes use of the ArcGIS least cost analysis tool. Least cost analysis determines the least costly avenue between two points where the cost of an area is evaluated from a raster data set with cell values equaling each cell's cost. The tool evaluates the eight neighbors of a raster cell and determines which neighboring cell will have the least accumulated cost. This process repeats until the start and ends points are connected.


Thesis_Sidehill.py - C:\Users\e1b8\Desktop\E1B8\Thesis\Thesis_Sidehill.py
File Edit Format Run Options Windows Help
Ln: 1 Col: 0

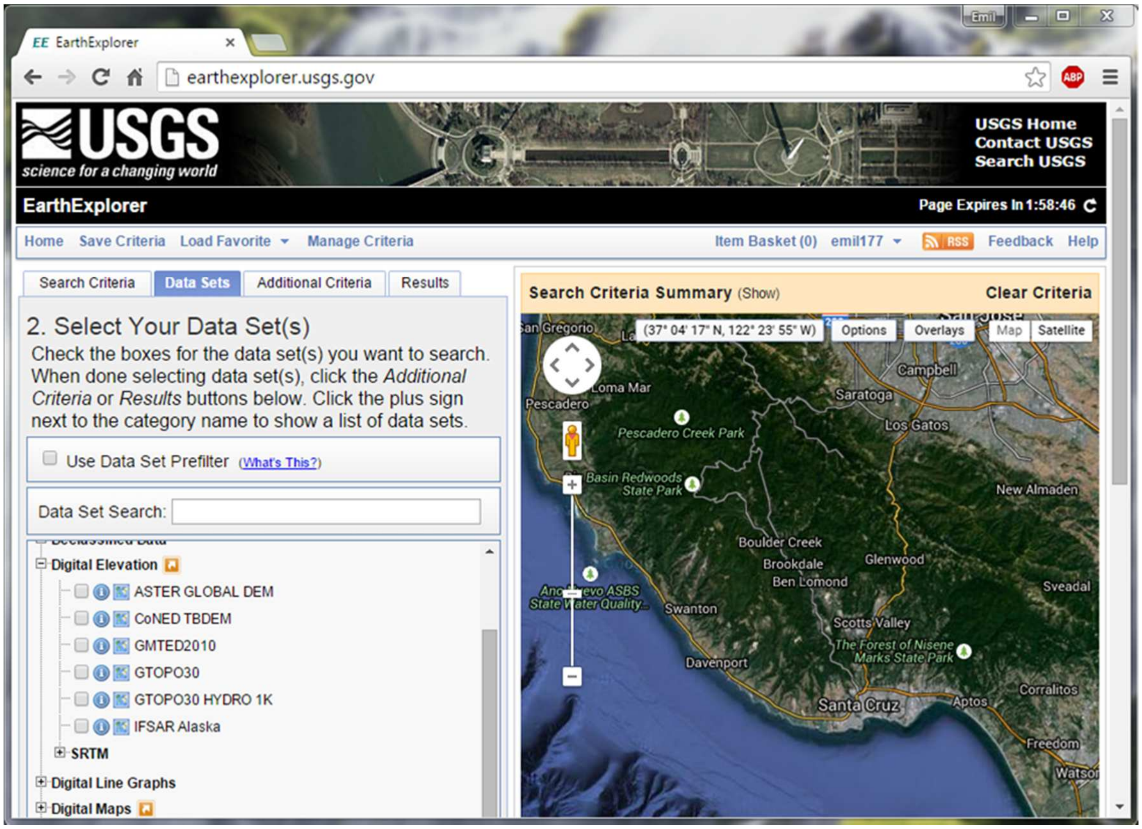
```
del cursor

#Select contour intersecting end point
SelectLayerByLocation_management ("Contourlyr", "", SnEndFC, "1 inch")
#Near analysis from start point to selected line
Near_analysis (SnStartFC, "Contourlyr", "", "LOCATION")
#Cursor to get X and Y coordinates generated from near analysis
cursor = da.SearchCursor (SnStartFC, ["NEAR_X", "NEAR_Y"])
for row in cursor:
    Second_Start = Point (row[0], row[1])
del row
del cursor

#Add Points
cursor = da.InsertCursor(SnStartFC, "SHAPE@")
cursor.insertRow([Second_Start])
del cursor
cursor = da.InsertCursor(SnEndFC, "SHAPE@")
cursor.insertRow([Second_End])
del cursor
```

Figure 3. The side hill method script loaded into IDLE.

The least cost tool is a Python file, which is a text file that has the file extension .py. Python files are simple to work with, and can be read by any Python integrated development environment (IDE). ArcGIS comes with an IDE named IDLE, which consists of a Python shell for testing code and tracking script outputs, as well as a window that lets the user create, view, and edit Python files. The window color-codes lines of code for easier readability. While working within an IDE is preferable because of the interface, any text editor can be used when working with a Python file. Python files can be incorporated into an ArcGIS toolbox as a custom tool.

Five inputs are required for the least cost method. The first four inputs are full paths to the required data sources. An easy way to access these paths is ArcGIS's ArcCatalog software. ArcCatalog is ESRI's file explorer software that is designed around accessing and manipulating GIS data. The desired files can be navigated to using ArcCatalog's catalog tree, and the full path can be copied from ArcCatalog's location box.

The first input is a digital elevation model (DEM) data set. DEMs are widely-available data layers and can be either raster-based cell grids or vector-based triangular irregular networks. The method developed for the least cost analysis requires raster DEMs, in which each cell value is the average elevation of the cell area. There are a variety of methods for generating DEMs, including satellite remote sensing, land surveying, and contour map interpolation. DEMs are available online using United States Geological Survey's (USGS) Earth Explorer website (Figure 4), GeoCommunity's DEM archive, the National Center for Environmental Information DEM Discovery Portal, and other websites easily found in a web search. These sites often make use of interactive maps that allow for easy selection of DEM data for a specific location, and also offer multiple resolution choices (cell size). A 10 meter resolution raster DEM was used to test the least cost method.



2. Select Your Data Set(s)

Check the boxes for the data set(s) you want to search. When done selecting data set(s), click the *Additional Criteria* or *Results* buttons below. Click the plus sign next to the category name to show a list of data sets.

□ Use Data Set Prefilter (What's This?)

Data Set Search:

- Digital Elevation
  - □ ASTER GLOBAL DEM
  - □ CoNED TBDEM
  - □ GMTED2010
  - □ GTOPO30
  - □ GTOPO30 HYDRO 1K
  - □ IFSAR Alaska
  - SRTM
- Digital Line Graphs
- Digital Maps

Figure 4. The USGS Earth Explorer web site, a source for DEM files.

The second and third inputs for the least cost method are two point feature classes. Each feature class should have a single point representing each end point of the new road. If either feature class has more than one point feature, the first point feature in the feature class will be used. The two point feature classes can be generated through geoprocessing with ArcGIS. A new feature class can be created for each, and a point can be placed using an editing session within ArcGIS. The feature classes can also be the end product of previous geoprocessing procedures, or files downloaded from the internet or other sources.

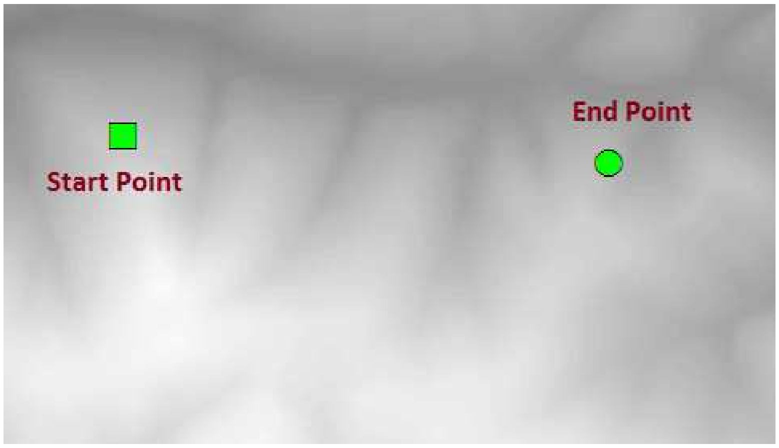


Figure 5. Start and end point features placed on a digital elevation model raster.

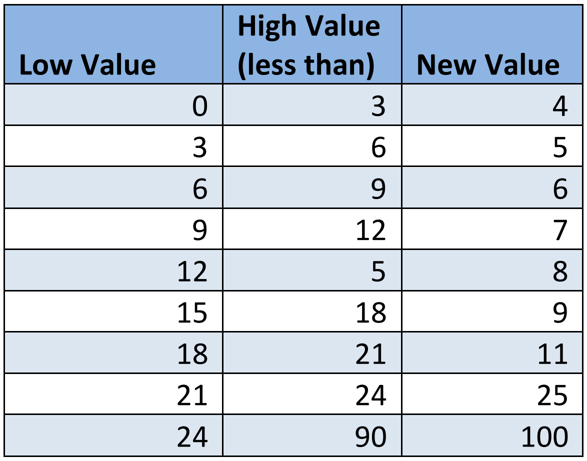
The final two inputs for the least cost method are the geodatabase for the tool's output as well as the output feature class's name. The output of the tool is a line feature class with a single feature representing the best location for a new road as determined by the least cost analysis.

Once inputs are defined by the user, the automation can start. The road line feature is generated automatically based on inputs by means of a number of geoprocessing steps. The first step of the least cost automation is importing the required modules. Python modules allow for extended functionality beyond Python's core methods. The least cost automation imports ArcPy to allow for ArcGIS's functionality, as well as the time module, which is used to report on how long the script ran for, and the *os* module, which makes working with file paths easier and more efficient.

After modules are imported, geoprocessing of the inputs begins. The first step in the code converts the input DEM to a slope raster. The least cost path method uses slope as the input cost, as it is generally desirable to place roads in locations with the most gradual (least) slope. The *slope* tool converts the raster into a slope, measured in degrees.

Once the input DEM has been converted to a slope raster, the slope raster is reclassified based on value categories with the *reclass* tool. The value categories were determined through a trial-and-error method, with tests being run on a number of terrains. The values are divided into multiples of three for one to 24 and given a relatively low new value, with the smallest slopes given the lowest new values. At a slope of 21 to 24, a large increase is given to the new value, from 11 to 25. Values above 24 are given a new value of 100, to strongly discourage the least cost analysis from selecting areas with these steep slope values.

Table 1. Slope classification for the least cost path method



Low Value	High Value (less than)	New Value
0	3	4
3	6	5
6	9	6
9	12	7
12	5	8
15	18	9
18	21	11
21	24	25
24	90	100

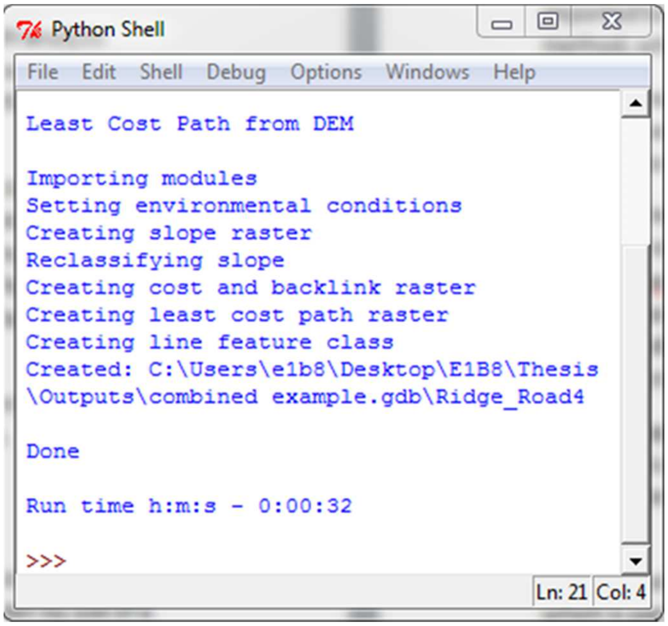
The next process of the least cost analysis is the generation of a cost distance raster. A cost distance raster is one of the two inputs required for a least cost calculation.

ArcGIS's *cost distance* tool is available through the spatial analyst extension. The tool calculates the least accumulative cost distance for each cell to the source point over the reclassified slope raster. The cost distance tool also produces a back link raster. A back link raster has cells with values between 0 and 8, with each value representing the direction along the least accumulative cost. This raster is used to guide the least cost from cell to cell.

After the generation of a cost distance raster comes the least cost analysis with the *cost distance* tool. The least cost analysis incorporates the starting point, the cost distance raster, and then the backlink raster to generate a least cost raster. The raster consists of a connected line of raster cells indicating the most efficient path based on the criteria provided.

With the output least cost generated, a few more processes are run before the script completes. The next process is converting the least cost raster to a line feature class. This conversion is done with use of the *raster to polyline* tool. The raster to polyline tool converts the path of cells from the least cost output and creates a polyline feature class. The polyline feature class contains a single feature: a polyline that follows the raster least cost. This feature class is the final output that can be used to plan the location of a road from the starting point to the ending point provided.

Finally, the script initiates a series of deletes for intermediate data with use of a Python list and a Python *for* loop. After this clean up step, the total run time is calculated and printed on screen.



```
Least Cost Path from DEM

Importing modules
Setting environmental conditions
Creating slope raster
Reclassifying slope
Creating cost and backlink raster
Creating least cost path raster
Creating line feature class
Created: C:\Users\e1b8\Desktop\E1B8\Thesis
\Outputs\combined example.gdb\Ridge_Road4

Done

Run time h:m:s - 0:00:32

>>>
```

Figure 6. The printed output of the least cost path method.

Side Hill Method

Quite often, forest land managers want to build a road along the side of a hill. The least cost method cannot be used effectively for this type of road design. The least cost method analyzes a terrain by use of a DEM raster and actively avoids steep areas, regardless of the desired direction of travel. A different method is needed for the design of side hill roads.

As discussed previously, maintaining a constant grade is one of the chief concerns when planning a road along the side of a hill. An automation was created that produces a line between two points on the side of a hill that maintains a constant grade from its start to its end. This is accomplished by converting the DEM to contours, selecting out the contours between the two points, and then selecting out a portion of each contour equal to the distance of the contour divided by the number of contours. In this fashion, the final created line is a series of steps along the contour grade of relatively equal distances, thus maintaining a near-constant grade.

As with the least cost tool, a Python script is used to determine a potential road between two points along the side of a hill. The side hill script is very similar to a text file, and is easy to execute for anyone who can work with GIS software. Six inputs are required for the side hill script. The first is the full path to the digital elevation model raster for the area where the new road feature will be generated. The second and third inputs are beginning and end point feature classes. Both point feature classes must contain a single point feature of one end of the new road. These points represent where the road will start and end. The fourth and fifth inputs are the output geodatabase that will house the feature class produced by the script, as well as the name of the new feature class. Finally, the sixth input is the contour interval used in the analysis. This input is a number that will be used by the script to create contours in whatever unit of measurement the input digital elevation model uses. The smaller the contour number input, the more contours the script will have to work with. Thus, the smaller the contour interval, the more the script has to process and the slower it will run. However, more contours also allow for a smoother transition for the line created along a hill slope.

Modules are imported after user inputs are defined. Again, the ArcPy, time, and os modules are imported to extend Python's functionality. The *sys* module is also imported to make use of its *exit* method, which can be called to exit the script at any time.

The next lines of code in the side hill script are a check to ensure that the input start and end points each only contain a single feature. The script makes this check through a conditional *if* statement, as well as ArcPy's *get count* function. The get count function returns the number of features within a feature class. If either of the two point feature classes do not contain a single point, the code prints a warning indicating that the start and end point feature classes must contain a single feature, and then exits the script.

ArcPy extensions and environmental conditions need to be set to complete the side hill method. ArcGIS's capabilities can be broadened with extensions that provide more geoprocessing tools. To use these tools, an extension must first be checked out that allows access to its functions. The side hill code makes use of several functions in the spatial analyst extension. When working with raster data, as the side hill method does, the environmental variable extent needs to be set so that the script knows the geographic area to work with as it processes. The extent covered by the input DEM raster is used as the environmental extent setting in the side hill script.

The script converts the DEM input into a contour line feature class with the *contour* function, which is part of the spatial analysis extension. Contours are line features that represent an equal elevation across a given area. Contour intervals are determined by the user's input. A feature layer is then created from the contour feature class. A feature layer allows for extended functionality for a feature class, including filtering the features based on certain criteria, as well as the ability to select particular features from the entire feature class.

After creating the contours feature class and feature layer, two copies of the beginning and end point feature classes are copied into memory. With use of the ArcPy module, geospatial data can be stored in a computer's memory. Storing data in memory means data does not have to be written on the hard drive. Geoprocessing with feature classes that are stored in memory often allow for faster processing time. Data that remains in memory after a script completes is automatically deleted. Storing data in memory is generally good practice for any intermediate data generated by a geoprocessing script.

Once copies of the start and end points are made, the copies are snapped to the nearest contour with use of ArcPy's *snap* function. For a point feature class, the snap function moves each point feature contained within the point feature class to the closest point intersecting the target feature class. The snap can be set to snap to the end of a feature (for line features only), the vertex of a feature, or the edge of a feature. In order to snap to the closest point in the contour feature class, the snap tool is set to snap to the edge of the contour features. Generally, contours will be spaced close together. Due to the close proximity of each contour to its neighbor, the snap function will only move the points a short distance. The points are snapped to the contours to allow for spatially relating the points and the contour lines to each other.

Only certain contours are relevant to the generation of the side hill road feature. The next steps in the script extract the contour features used by the side hill automation from the entire DEM-generated contour feature class.

The first step of extracting the relevant contour sections involves ArcPy's *select layer by location* function. Selection is important when one wishes to limit the scope of a geoprocess to a subset number of features within a feature class. The select layer by location function selects features based on their spatial relationship with other features. In the side hill script, the contour intersecting (touching) the snapped start point is selected.

After selecting the contour feature intersecting the starting point, a *near analysis* is performed between the selected contour and the end point. A near analysis is generally used to determine which features are nearest to other features. The tool is used in the side hill script to determine the coordinates of the closest point of the selected contour (the contour intersecting the start point) to the end point. The near analysis function adds two new fields to the input feature class's (the end point feature class) attribute table. The function adds a *NEAR_X* and a *NEAR_Y* field to the point's attribute table, and populates it with the *x* and *y* coordinates for the point along the selected contour nearest to the end point. The script then accesses this information by use of a *search cursor*. The search cursor iterates through an attribute table's rows and returns the specified field's data.

Once coordinates are generated and stored with the near analysis tool and subsequent search cursor, a point feature is created from the coordinates. This point feature is added to one of the copies of the end point feature classes stored in memory. The addition of the point is accomplished with an *insert cursor*. An insert cursor is similar to a search cursor in that it accesses the feature class's attribute table, but an insert cursor allows for the insertion of a new feature into a feature class's rows.

After inserting the point into the end point feature class, a line feature class is created between the two points by use of the *points to line* function. Since the input point feature class only contains two points, a single line feature is created in the new line feature class. This line runs from the end point to the nearest point along the contour intersecting the starting point.

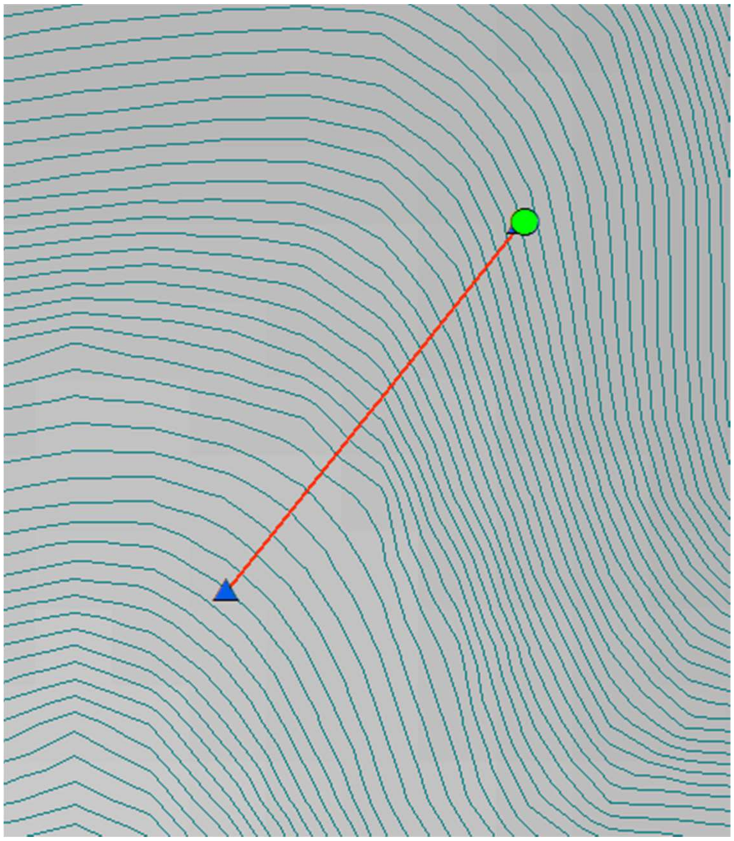


Figure 7. A start point (green dot), the point that intersects the end point's contour (blue triangle) and the line between the two that will be used to clip the contours.

A point feature class is then generated that contains points for each location where the generated line intersects the contours. These points are generated with the *intersect* function. The intersect function analyzes how features in various feature classes spatially intersect each other and can create a variety of outputs depending on the types of inputs used: points, lines, or polygons. In this case, two line feature class inputs produce a point feature class output where the line features in the two feature classes intersect.

The points from the intersection of the generated line and the contours are used to split the contour features at the points. This is done with the *split line at point* function. The process of creating a line from the end point to the nearest contour intersecting the start point, and then splitting the contours by that line, is mirrored with the start point and the contour intersecting the end point. This process creates a feature class of the contours, split at the lines between the start and end lines. The contour features between the start and end lines are then selected by use of a select by location. First, the lines intersecting the start line are selected. Next a subset selection is made on the contours that intersect the end line. What remain selected are the contours between the start and end lines. These contour features are saved into their own feature class, with use of the *feature class to feature class* function.

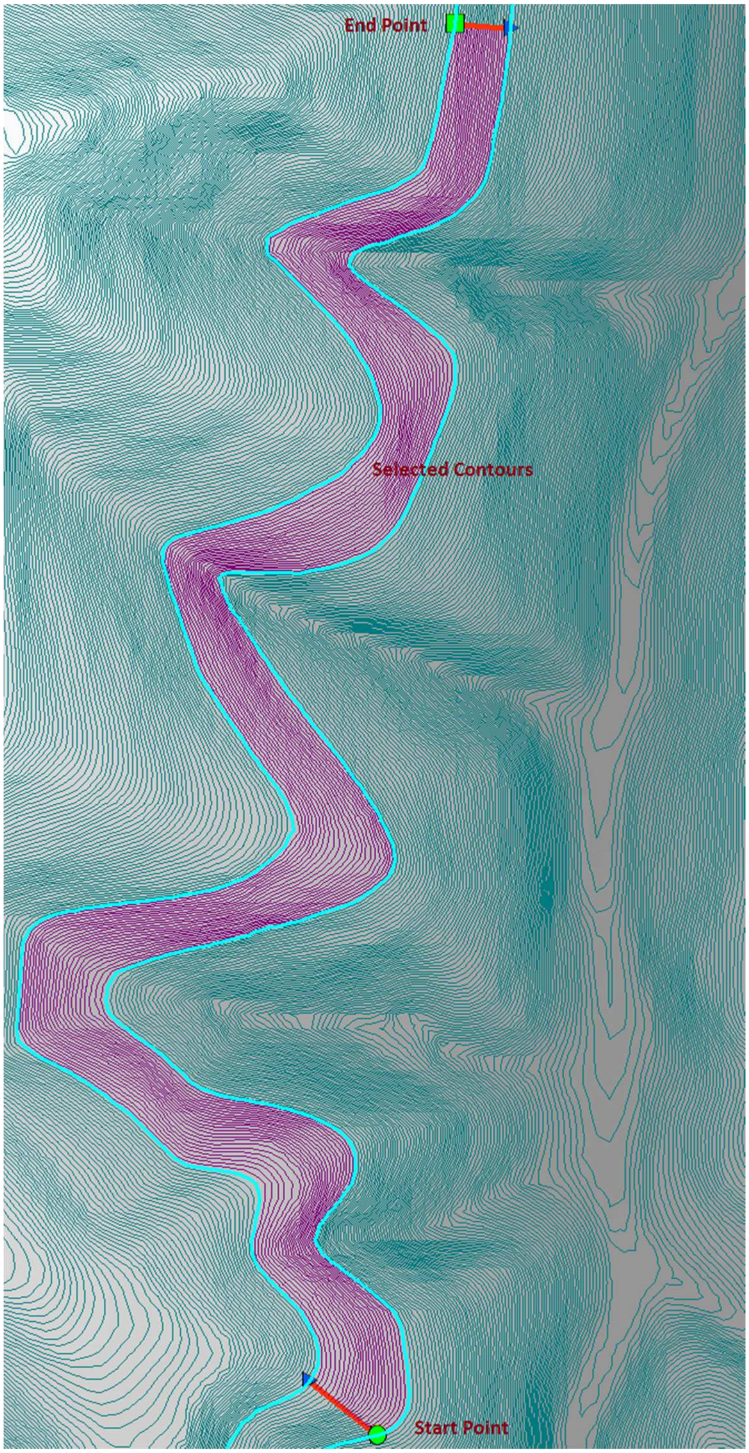


Figure 8. Contours between start and end points (purple lines) used to generate a side hill line feature.

With the relevant contours in their own feature class, the number of contours is determined by use of the get count function described previously. The first contour, which intersects the starting point, is then selected. The length of the line is determined through a search cursor. The length of the contour is then divided by the number of contours to determine the length of selection. Line selection is accomplished by dividing the contour feature by its vertices into multiple features. Each line feature can contain any number of vertices. Dividing a line feature into multiple features by its vertices creates a series of smaller features. Since the contour feature was generated with the contour function from a DEM, the contour has a large number of vertices and divides into many smaller features. The division is performed with use of the *split line* tool, which divides lines by their vertices. The lines are then split twice more. Midpoints to each line feature are created with the *feature to point* tool. The lines are split by their midpoints, with use of the split line at point tool. The remaining lines are now no more than a few feet in length.

The script then iterates through a number of select by locations with use of a Python *while* loop. The script selects the first line feature, intersecting the snapped start point. The script then continues to select connected line features until all the line features selected add up to the desired distance.

Once the proper distance is selected, the script creates a line between the starting contour and the next contour. This is done with use of the *feature vertices to points* method, which creates points at both ends of the selected contour line. The starting point is selected by a select by location and deleted. This leaves a single point at the end of the selected contour. A near analysis is performed to find the coordinates of the nearest point on the next contour. This point is added to the contour end point and a line is created. This line is the connector line from the first contour to the second. This line is combined with the selected contour features and merged into one feature class with use of the *merge* function. The merge function combines features from multiple feature classes into a single feature class.

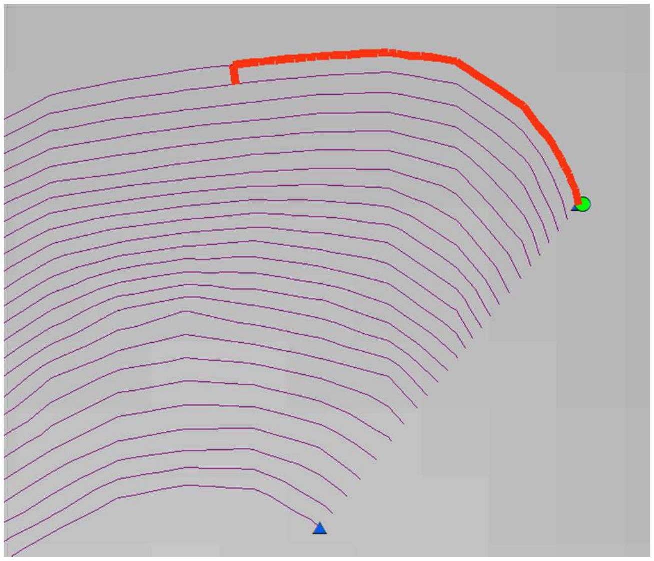


Figure 9. The selected segment of the first contour, starting at the starting point, and its connecting line to the next contour (red line).

The process for the first contour is then repeated for each additional contour between the start and end points. The length of the output line is determined by dividing the length of the contour by the number of contours between the two points. Each intermediate contour is then divided at its vertices, and divided twice more at its midpoints, to create lines of no more than a few feet in length. A series of selections are again performed until the length of the selected lines is equal to the desired length. An end point is created at the end of this line, a near analysis is performed to determine the point along the next contour to start at, a line is generated between the two points, and all lines generated are merged together. This process continues until the final contour is reached. Once the final contour is reached, it is divided at the appropriate point and the final section is merged with the other generated sections.

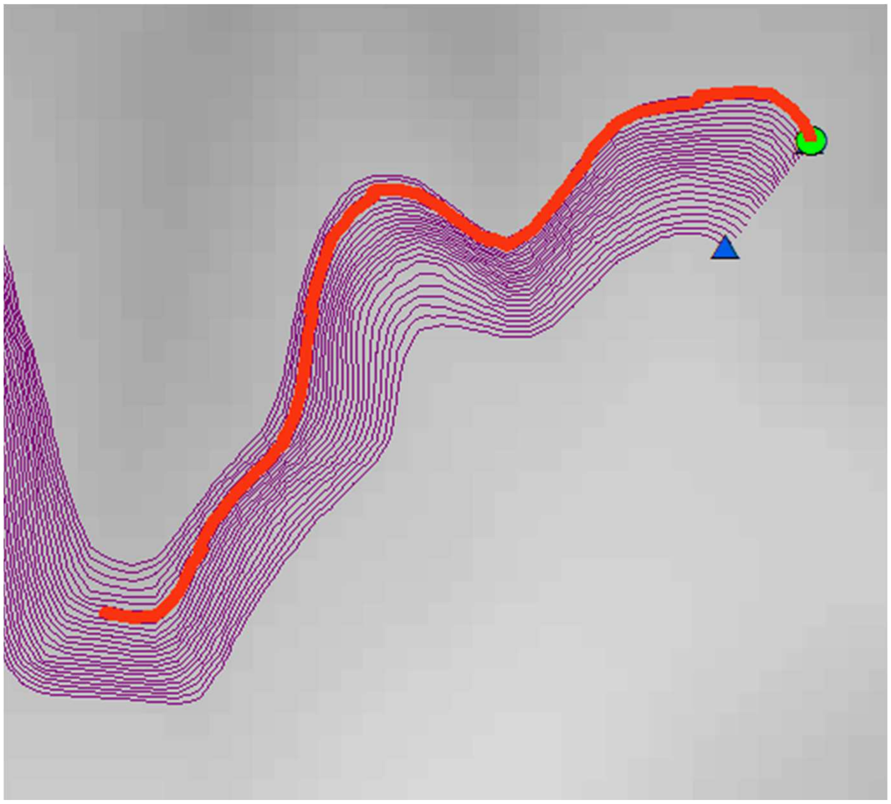


Figure 10. A series of contour lines and steps (red line).

Once all parts of the new line feature class are created, the features are dissolved into a single feature with use of the *dissolve* function. The dissolve function creates a single feature from multiple features, merging all of their line segments together. This dissolved line feature class is the final output of the side hill script.

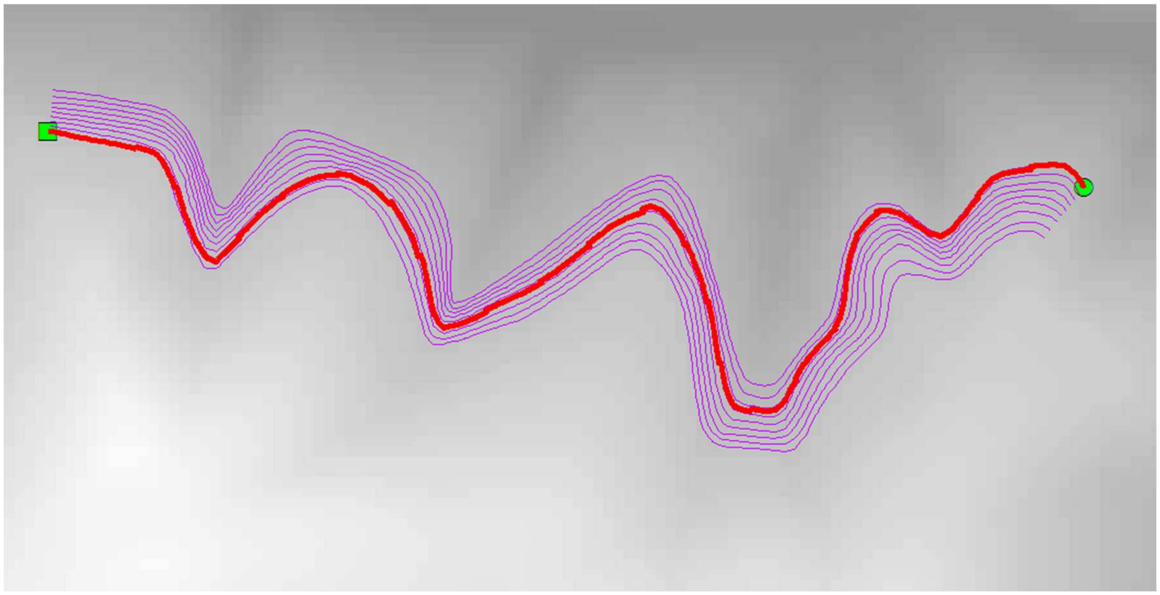


Figure 11. The final line feature (red line) from start point to end point.

Discussion

Implementation

The least cost path and side hill methods can be used a single time when a road between two control points is needed or repeatedly when a new road network is desired. To optimize the two methods, the planner must first determine which method best suits his or her needs. On steep ground, roads are often constructed along ridge lines and valleys. For such roads, the least cost method is the proper choice. The least cost method generates a smooth road line along the crest of a ridge or the flat of a valley. If the side hill road method is employed for a ridge or valley situation, the resulting feature is a jagged line that repeatedly rises and falls along the ridge or valley, which would be impractical and inefficient in a real-world scenario. The side hill method should be employed to design a road feature that runs along a slope. The side hill method creates a line feature that maintains its grade between its starting and ending locations. When the least cost path method is employed for starting and ending points on hill sides, the generated feature is generally a line that first runs from the starting point to the nearest flat ridge or valley, and then along the valley or ridge until it nears the ending point, where it makes a sharp turn and runs straight to the ending point. This design would be impossible to implement in a real-world situation due to the steepness of the grade.

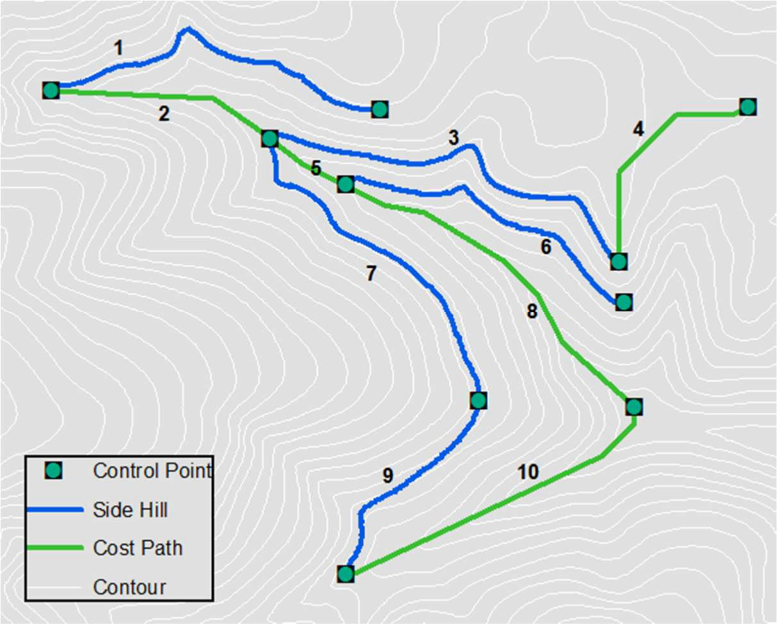


Figure 12. An example of the side hill method output (blue lines) and the least cost path method (green lines) used together to create a road network.

Figure 12 shows an example of how the two methods may be used to create a road network in steep terrain with the fish bone design. To achieve results such as those in Figure 12, the road designer would first choose control points based on terrain and other considerations, and create those points as GIS features. He or she would then pick the appropriate method to connect control points based on the topography between the points.

The least cost method is best used when main backbone roads along valleys or ridgelines are desired. These roads must follow slopes at their most gradual incline, such as along the top of a ridge. The least cost method evaluates the input DEM for specifically this circumstance. In Figure 12, the line features 2, 4, 5, 8, and 10 were generated with the least cost method.

The side hill method is best used when secondary rib roads branching off of main roads are desired, or whenever valleys and ridges prove too steep to traverse directly. In Figure 12, the line features 1, 3, 6, 7, and 9 were generated with the side hill method.

A road planner can create and evaluate road layouts in a timely manner with the least cost and side hill methods. The example in Figure 12 can be generated in about a half hour. Creating such line features manually, even with the aid of today's robust GIS software, would not generally be achievable in an equal amount of time if the same degree of precision is desired. Once a road layout is in place, the designer can evaluate the layout for factors such as length of road and expected grade. If needed, the planner can create and evaluate a number of layouts to determine which allows for the most efficient design. The planner can begin the field investigation phase of road implementation once a satisfactory layout is found.

Limitations

The least cost and side hill methods have limitations in their use. One limitation is the ability of the two tools to evaluate flat ground. Neither method is designed to work effectively in areas with especially shallow grade. Least cost line segment 4 in Figure 12, which is placed along such an area, is highly generalized, consisting of a horizontal line, a vertical line, and a middle line at a 45-degree angle between the two. In these cases of markedly flat ground, grade becomes a much less important factor in road placement. Other factors in these situations, such as natural obstacles, drainage, and soil type, outweigh the importance of grade for road implementation consideration.

These two methods can also cause problems with the inflection angles of road segments at control points. Roads in steep terrain, especially those that will be used by loaded logging trucks, cannot include sharp angles of inflection. In Figure 12, the angles created between lines 1 and 2, lines 3 and 4, and others are undesirable. The road designer will need to initially choose control points that allow for large angles of inflection and adjust the least cost and side hill method outputs accordingly after generation.

Design Alternatives

Currently, the methods exist as a stand-alone Python scripts. While this format for an automated Python GIS processes has strength in its simplicity, it is not without its issues. A basic understanding of Python coding is needed to use these scripts. To reference GIS data in the scripts, the user must type or paste the full path of the GIS data directly into the script. Insufficient understanding of code variables, syntax, and Python strings (text) may make this somewhat confusing to a user who has never worked with a Python script. These data paths must be surrounded by quotation marks, for example. They also must have an *r* before the string to indicate the string is a *raw* string. Raw strings ignore special characters, such as the backslashes. Having the entirety of the code exposed to the user may also be problematic to a new user, as a simple mistype within the body of code can cause the script to fail, and it may be impossible for the casual GIS user to rectify the error.

Incorporating the code as a Python tool into a custom toolbox for ArcGIS ArcMap is an option that addresses these issues. A Python tool can be set so that inputs are limited to specified values, so that no understanding of coding is required. The code itself is also not available for manipulation by the user. Creating a tool would present new challenges and limitations to the user, however, such as the need to install the toolbox into ArcGIS. Running a script within the ArcGIS environment increases the script's running time, and so reduces efficiency.

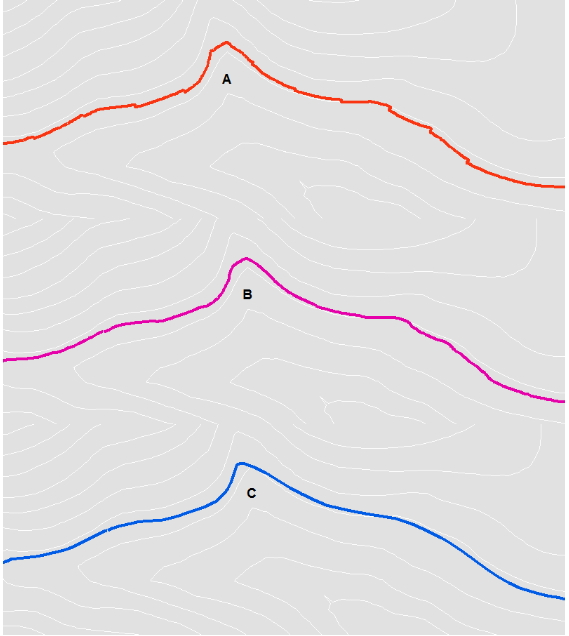


Figure 13. An example of a side hill method line feature (A), and the same line after the *smooth line* tool has been applied to the feature with different tolerances (B and C).

The side hill method generates features with sharp, near-90 degree angles between its contours that are noticeable at large scales. Smoothing these features would make for more realistic road segments. ArcPy has the *smooth line* tool that may rectify this situation. In Figure 13, smoothing with the smooth line tool has been performed to two extents. Line feature A represents the output of the side hill method with one meter contours and without smoothing applied. Sharp angels can be seen where the automation moved from one contour to the next. Line feature B is the output of the smooth line tool applied to line feature A, with a smoothing tolerance of 10 meters applied. Line feature C is the output of the smooth line tool applied to line feature A, with a smoothing tolerance of 100 meters applied. The larger the tolerance applied, the smoother the angles become. This improves the consistency of the grade on relatively straight segments of line, but can cause issues with rounded curves. The rounded curve above the letter C in Figure 13 can be seen to be sharper than in feature A, which is undesirable. Further experimentation and analysis with the smooth line tool may improve the jagged nature of the side hill method outputs, but if it is found to cause an excess of distortion the tradeoff may not be worth its implementation.

Conclusion

For land managers, intelligent road planning based on all available information and technologies is essential to maintaining ecological integrity. Degradation of waterways through sedimentation can have substantive negative impacts on the many species residing in those waterways. The aquatic impacts can, in turn, affect the terrestrial life tied to the rivers and streams. GIS is a powerful tool with widespread use by land managers today. The least cost path and side hill methods have been shown to quickly generate road layouts that can be compared and studied. The least cost path method is best utilized for creating a road feature along a ridge or in a valley, while the side hill method is best utilized for placing a road feature along the side of a slope. These time-saving automations improve road placement in the initial route selection phase, which in turn can lead to better designed, sounder, more ecologically stable road networks.

References

Anderson, B., & Potts, D.F. (1987). Suspended sediment and turbidity following road construction and logging in western Montana. *Journal of the American Water Resources Association, 23*, 681-690.

Beschta, R.L. (1978). Long-term patterns of sediment production following road construction and logging in the Oregon Coast Range. *Water Resources Research, 14*, 1011-1016.

Bowers, S., Garland, J.J. (2012). *Designing woodland roads*. Corvallis, OR: Oregon State College.

British Columbia Ministry of Forests. (2002). *Forest Road Engineering Guidebook* (2nd ed.). Victoria, B.C: British Columbia Government Publications.

Daykin P. (1965). Application of mass transport theory to the problem of respiration of fish eggs. *Journal of the Fisheries Research Board of Canada*, *22*, 159-171.

Dunne, T., & Dietrich, W. (1982) *Soil erosion and conservation in the tropics*. Madison, WI: American Society of Agronomy and Soil Science Society of America.

ESRI, (2015). *Independent Report Highlights ESRI as Leader in Global GIS Market*. Retrieved October 11, 2015 from http://www.ESRI.com/ESRI-news/releases/15-1qtr/independent-report-highlights-ESRI-as-leader-in-global-gis-market

Evenson, D. (2001). *Egg pocked depth and particle size composition with chinook salmon redds in the Trinity River, California* (Unpublished master's thesis). Humboldt State University, Arcata, CA.

Frissell, C. (1992). *Cumulative effects of land use on salmonid habitat on southwest Oregon streams* (Unpublished doctoral thesis). Oregon State University, Corvalis, OR.

Gumus, S., Acar, H.H., & Toksoy, D. (2007). Functional forest road network planning by consideration of environmental impact assessment for wood harvesting. *Environmental Monitoring and Assessment, 142*, 109-116.

Jusoff, K. (2008). Construction of New Forest Roads in Malaysia Using a GIS-Based Decision Support System. *Computer and Information Science, 1*(3), 48-59.

Knopp, C. (1993). *Testing Indices of Cold Water Fish Habitat*. Santa Rosa, CA. North Coast Regional Water Quality Control Board.

Lemly, A.D. (1982). Modification of benthic insect communities in polluted streams: combined effects of sedimentation and nutrient enrichment. *Hydrobiologia, 87*, 229-245.

McCashion, J.D., & Rice, R.M. (1983). Erosion on logging roads in northwestern California: how much is avoidable? *Journal of Forestry, 81*, 23-26.

Megahan, W.F., & Kidd, W.J. (1972). Effects of logging and logging roads on erosion and sediment deposition from steep terrain. *Journal of Forestry, 70*, 136-141.

Montgomery, D.R. (1994). Road surface drainage, channel initiation, and slope instability. *Water Resources Research, 30*, 1925-1932.

Naiman, R.J., Bilby, R.E., Schindler, D.E., & Helfield, J.M. (2002). Pacific salmon, nutrients, and the dynamics of freshwater and riparian ecosystems. *Ecosystems*, *5*, 399-417.

Nawa, R.K., & Frissell, C.A. (1993). Measuring scour and fill of gravel stream beds with scour chains and sliding bead monitors. *American J. of Fisheries Management*, *13*, 634-639.

Newlon, T.A., & Rabe, F.W. (1977). *Comparision of macroinvertebrate samplers and the relationship of environmental factors to biomass and diversity variability in a small watershed*. Moscow, ID. Idaho Water Resources Research Institute, University of Idaho.

Newnham, R.M. (1995). ROADPLAN: a tool for designing forest road networks. *Journal of Forest Engineering, 6*(2), 17-26.

Picarda, N., Gazull, L, & Freycon, V. (2006). Finding optimal routes for harvesting tree access. *International Journal of Forest Engineering, 17*(2), 35-50.

Reid, L.M., & Dunne, T. (1984). Sediment production from forest road surfaces. *Water Resources Research, 20*, 1753-1761.

Rice, R.M., Tilley, F.B., & Datzman, P.A. (1979). *A watershed's response to logging and roads: south fork of Casper Creek, California, 1967-1976.* Berkeley, CA. USDA Forest Service.

Rogers, L. & Schiess, P. (2001). *Pegger & Roadview – a new GIS tool to assist engineers in operations planning*. Paper presented at The International Mountain Logging and 11th Pacific Northwest Skyline Symposium, Seattle, WA.

Silver S.J., Warren, C.E., Doudoroff, P. (1963). Dissolved oxygen requirements of developing steelhead trout and chinook salmon embryos at different velocities. *Transactions of the American Fisheries Society, ;92*, 327-343.

Stuckelberger, J.A., Heinimann, H.R., Chung, W., & Ulber, M. (2006). *Automatic road-network planning for multiple objects*. Paper presented at Council of Forest Engineering Conference Proceedings: "Working Globally – Sharing Forest Engineering Challenges and Technologies Around the World, Coeur d'Alene, ID.

Suttle, K.B., Power, M.E., Levine, J.M., McNeely, C. (2004). How fine sediment in riverbeds impairs growth and survival of juvenile salmonids. *Ecological Applications, 14*, 969-974.

Swanson, F.J., & Dyrness, C.T. (1975). Impact of clear-cutting and road construction on soil erosion by landslides in the western Cascade Range, Oregon. *Geology, 3*, 393-396.

TIOBE, (2015) *TIOBE Index for October 2015*. Retrieved October 11, 2015 from http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Turnpenny A.W.H., & Williams R. (1980). Effects of sedimentation on the gravels of an industrial river system. *Journal of Fish Biology, 17,* 681-693.

USDA Forest Service. (2015). *Forest Road Construction and Maintenance.* Retrieved October 1st, 2015 from http://www.nrs.fs.fed.us/fmg/nfmg/docs/mn/roads.pdf

Welsh, H.H., Ollivier, L.M. (1998). Stream amphibians as indicators of ecosystem stress: a case study from California's redwoods. *Ecological Applications, 8*, 1118-1132.