REPORT

Extending the Burma Road
In 1937, Japan invaded China. By late 1938, the Republican government of Chiang Kai-shek had been driven far into the mountains of western China, and that country and its armies were nearly cut off from external aid. To support their allies, the British built the Burma Road over the Shan Plateau and the eastern reaches of the Himalaya Mountains to Kunming. Many thousands of tons of war material made its way over this road to support the Chinese. The Japanese invaded Burma in 1942 and ended Allied land supply to China until 1945.

It's now February of 1945. The Allies are in control of northern Burma once again, and the Japanese are retreating. Supplying China by land is once again possible. General “Vinegar” Joe Stilwell has asked you to identify a route for a new road into China to increase the flow of supplies using this newfangled “GRASS” thing on this newfangled “Computer” that you have invented. Stilwell was born in the 1800's, so he can use words like “newfangled” without irony. The road origin is at coordinates (438740,2952290) and the destination is at coordinates (578230,3204510).

For your report:
1. generate two alternative routes: in one, employ a river penalty friction surface, while in the second, employ a friction surface with increasingly high penalties for higher slopes.
2. Produce maps of the routes, along with estimates of the road lengths, in kilometers.
3. How many bridges over significant rivers will be required for each route? Which route would you recommend, and why?
4. Generate a cost corridor for your preferred route to give planners some ideas about similarly-difficult alternatives.
5. Identify any questions or concerns you might have about either of the routes.

INSTRUCTIONS

1. Set up in GRASS
Fire up the burma GRASS location from the earlier lab. Load the DEM and display it. View the stats with r.univar. All should look good, or at least familiar.

2. Create some friction cost surfaces
River crossings are to be avoided. To create a river map, develop a flow accumulation raster map. If you use r.watershed, be sure to set the negative values to positive afterwards. Then create a new map called rivers10k in which any cell with an accumulation larger than 10,000 gets a 1000, while anything else gets a 1. This is a cost surface in which non-river cells have a low and constant cost (1) and river cells have a really high cost (1000).

The second surface will discourage steeper slopes by assigning a higher cost to them. Yes, slope is built-in to r.walk, but there is no cost to traversing progressively steeper slopes - a slope of 10 degrees is treated the same as a slope of 80 degrees. Do this by generating a slope map (degrees). Then reclassify so that slopes less than 5 degrees get a 1, slopes 5-10 get a 5, slopes 10-20 get a 50, slopes
20-45 get a 100, and slopes above that get a 500. Finally, add this to rivers10k to create a new raster map called friction_slope. This includes slope plus river crossing costs.

3. Generate accumulated cost surfaces
Run r.walk (Raster → Terrain analysis → Cumulative movement costs). Click Knights move to 'off'. Usually we want it on, but if you have to fill the subsequent cost surface (see next step) it's better to do it this way. The input elevation map is burma. The cost map is rivers10k. Call the output burma_cost1. Input the start and end points. Leave the default settings and Hit Run. It will take a couple of seconds to process.

Do the same, but using friction_slope, and call the output burma_cost2.

4. Generate the least cost paths
Use r.drain for this (Raster → Terrain analysis → Least cost route or flow). For elevation raster map, enter burma_cost1. Call the output constant_path. On the Start tab, enter the coordinates of the destination (NOT the start!). Otherwise you will have a length-0 path! Then hit Run. This will take a couple of seconds to complete.

Do the same, but using burma_cost2 and calling the output slope_path, to calculate the second least-cost path.

Display slope_path and constant_path. Do they go all the way from the destination to the origin? If not, they hit a sink in the surface. To fix this, run r.fill.dir from the GRASS prompt. Tell it to fill the problem burma_costx surface. Name the output filled surface burma_costxf. Then run r.drain on this filled surface instead of the original. Redraw. Did that fix the problem?

5. Convert paths to vector lines
Once your paths connect the origin to destination, use r.to.vect to convert to vector (File → Map type conversions → Raster to vector). Make sure you are converting to lines. Then display it in the window, perhaps setting a good color and widening the pixel width.

6. Visualize and report on path characteristics
Create a map, maybe draping your routes across the elevation surface. Vectorize your rivers (using rivers10k, but turn the 1-values into nulls first) and display those, too, since river crossings are an important issue. Run v.report (Vector → Reports and statistics → Report topology by category) to figure out the length of your routes.

Extra! You can also display the slopes along your routes; this would enable you to identify portions of your route with slopes greater than 30 degrees, for example. Think about how you would do this in raster, using the path raster and a slope map, or a reclassified slope map (high, medium, low). If you have time, try it out.

7. Create a Cost Corridor map
Rerun r.walk on your preferred route, but reverse the begin and end coordinates. This will produce a new accumulated cost surface increasing out from your end coordinate. Add this surface to the one generated from the beginning point, and display. Are there decent alternative corridors to that followed by your main route?