EPA133a. Advanced Simulation
Assignment 1. Data Quality Issues for Data-Driven Simulation

Introduction

As presented in the Bangladesh case introduction in the class, we want to run simulations with accurate road data, river data, rail data, bridge data, and economic data for assessing the criticality and vulnerability of Bangladesh's transport infrastructure as well as testing the robustness of envisioned policy measures. Several data sources have been identified and made available:

- **RMMS** – a raw road dataset from the Ministry of Transport for all N (National), R (Regional), and Z (Zila) roads in Bangladesh. The dataset is stored in a ZIP file and consists of 7 files per road (where Rid stands for the road id, e.g. N1 or Z1041):
    - Rid.detail.htm: overview file with summary data
    - Rid.divisions.htm: file with information which divisions the road covers
    - Rid.lrps.htm: file with detailed data about locations, bridges, signs, crossings, etc.
    - Rid.traffic.htm: file with counted or estimated traffic per type
    - Rid.txt: overview file with the summary road information and original sources
    - Rid.widths.txt: raw width file as taken from the Flash web-page for the road
    - Rid.widths.processed.txt: processed data with lanes between chainage points

- **BMMS** – a raw bridge dataset from the Ministry of Transport. The dataset is stored in a ZIP file and clustered by road name in separate folders. It consists of 4 files per bridge:
    - Rid.LRPid.bridgeid.bcs1.htm: overview information about the bridge with location, structure, and substructure of the bridge. This file can show the data is absent
    - Rid.LRPid.bridgeid.bcs2.htm: survey data file; ~~not important for the simulation~~
    - Rid.LRPid.bridgeid.bcs3.htm: location, condition and maintenance information about the bridge. If the bcs1 file is missing, this file contains the chainage on the road for its location estimation
    - Rid.LRPid.bridgeid.txt: summary file with processed information and original sources

The Rid.txt, Rid.widths.processed.txt, and Rid.LRPid.bridgeid.txt files are already processed at TU Delft, and can contain wrong or unhelpful assumptions about processing of the data. The .htm files, and the Rid.widths.txt file are raw files that come directly from the Ministry of Transport. They can contain data problems, but it is the best source for infrastructure we have.

In the Zip file or folder of the Java simulation program, several other files can be found:

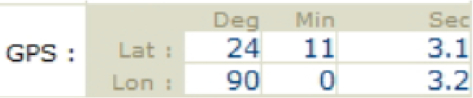
- **infrastructure/BMMS_Overview.xlsx**: a generated Excel file used by the simulation to draw the bridges. If you repair the bridge information, you have to create a file with the exact same name and column structure for the simulation to be able to read it. The columns are:
    - A: road – road name
    - B: km – the km point from the bcs1 (or bcs3) file
    - C: type – type of bridge; see one of the next sections
    - D: LRPname – the location reference point of the bridge
    - E: name – the bridge name if present, or '.' if unknown
    - F: length – the bridge length in m

  - G: condition – the condition from A (good) to D (bad)
  - H: structureNr – the official bridge nr in the database
  - I: roadName – the name of the road
  - J: chainage – the chainage from the bcs3 file (should be the same as column B)
  - K: width – the width of the bridge in m if known
  - L: constructionYear – the construction year of the bridge if known
  - M: spans – the number of spans [arcs] of the bridge if known
  - N: zone – the administrative zone in the country
  - O: circle – the administrative circle within the zone
  - P: division – the administrative division within the circle
  - Q: sub-division – the administrative sub-division within the division
  - R: latitude – latitude as retrieved, calculated, or estimated
  - S: longitude – longitude as retrieved, calculated, or estimated
  - T: estimatedLoc – information where the lat/lon comes from in the processing by TU Delft:
    - bcs1: taken from the bcs1 file
    - bcs1_zerosec: taken from the bcs1-file but seconds were missing and therefore assumed to be zero
    - road_precise: taken from the LRP in bcs3, which had an exact match with an LRP in the road database, from which the lat/lon coordinates were taken
    - road_chainage: taken from the chainage in bcs3, which had an exact match (plus or minus 10 m) in the road database, from which the lat/lon coordinates were taken
    - road_interpolate: taken from the chainage in bcs3, for which the lat/lon coordinates were interpolated fractionally based on the two nearest LRPs in the road database
    - error: no match was found with any of the above methods.
- **infrastructure/_roads.tsv**: a tab-separated text-file with processed information from the RMMS dataset. It contains a first line with explanation, and the following tab-separated data starting from line 2, with one road per line. This is also a file you can re-create after repair:
  - road name: the official name of the road
  - lrp_1: the name of the first LRP of the road
  - lat_1: the latitude of lrp_1
  - lon_1: the longitude of lrp_1
  - ...
  - lrp_n: the name of the last LRP of the road
  - lat_n: the latitude of lrp_n
  - lon_n: the longitude of lrp_n
- **infrastructure/water**: contains ESRI shape files (GIS data with the .shp file as the main data containing file) and Excel files about the 53 most important waterways in Bangladesh. This information is used to draw the red waterway lines on the map. Waterways are in 4 classes, as explained in the file **infrastructure/water/WaterwayTypes.xlsx**.
- **gis**: the gis folder contains the ESRI shape files for drawing the background information and maps. The map.xml file controls which shape files are rendered on the map, in which colors, and in which way. If you know xml, you can edit the **gis/map.xml** file, and obtain different results for the background maps.

    - **osm:** the osm sub-folder contains the OpenStreetMap files in ESRI shape-file format to render roads, rivers, railroads, buildings, etc.
    - **osm-countries**: the osm-countries sub-folder contains the OpenStreetMap files in ESRI shape-file format to render the contours of neighboring countries
    - **gadm**: the gadm folder contains ESRI shape files to render the administrative areas in Bangladesh on different levels (level 0 to level 3; zones to subdivisions).
    - **wfp**: the wfp folder coming from the World Food Program contains ESRI shape files to render a number of the rivers with their appropriate widths.
- **infrastructure/Roads_InfoAboutEachLRP.csv**: a comma-separated text-file with processed information from the RMMS dataset. It contains information about one LRP per line.

Lat/Lon information

In the BCS1-file, latitude and longitude are given in degrees, minutes, and seconds:



		Deg	Min	Sec
GPS :	Lat :	24	11	3.1
	Lon :	90	0	3.2

These can be calculated into the decimal latitude and longitude by (deg + min/60 + sec/3600). Note that the longitude is the x-coordinate on the map, and latitude is the y-coordinate.

Chainage and LRP numbers

In the case of roads or other linear infrastructure, a chainage (derived from Gunter's Chain - 1 chain is equal to 66 feet or 100 links) will be established, often to correspond with the center line of the road or pipeline. During construction, structures would then be located in terms of chainage, offset and elevation. Offset is said to be "left" or "right" relative to someone standing on the chainage line who is looking in the direction of increasing chainage[1].

The LRP number starts with LRPS (Start) for every road, and ends with LRPE (end). In principle, the LRP numbers contain 3 digits and indicate the km number from the start. When multiple objects are identified in the kilometer, reference points are indicated with suffices a, b, c, etc. This also holds for LRPS, which can have ids such as LRPSa (reference to an object location within the first km of the road).

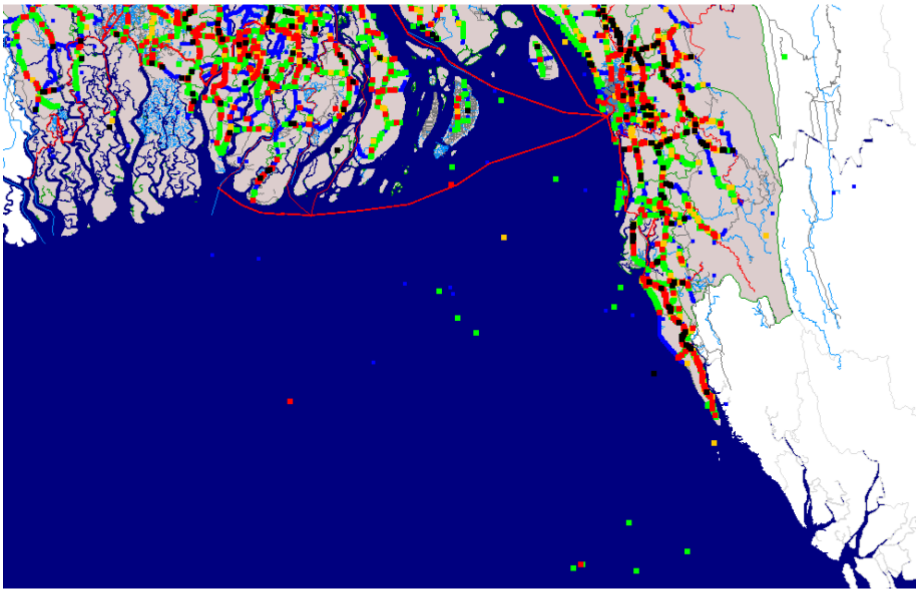
Note that when the road is maintained, e.g. straightened or redirected around a town, the chainage changes. It is always difficult to determine whether the chainage is consistently changed for all objects that relate to the road. The LRP numbers (originally containing the chainage in whole kms) do not change. Therefore, for longer roads, an LRP number of e.g., 100 can belong to a chainage of 96 or 104, or even a number that deviates more…

Data Quality Issues

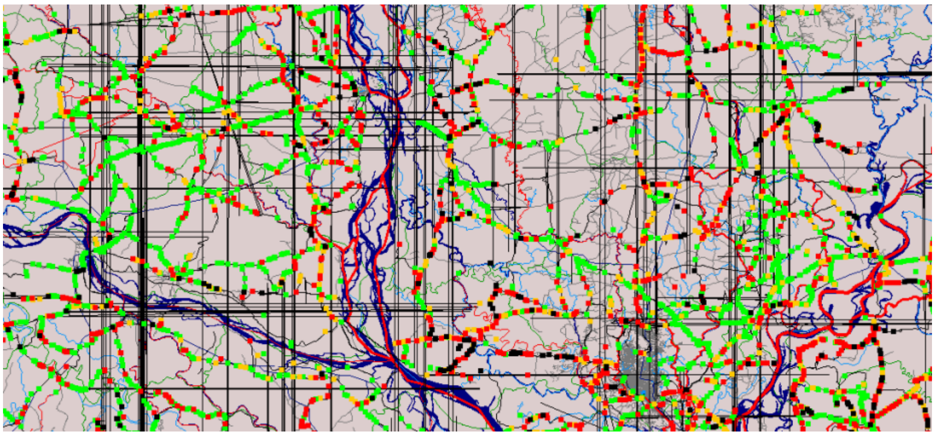
There are quite some issues with the data, as can be seen on the map.

The bridge information from **WBSIM/infrastructure/BMMS_Overview.xlsx** leads to bridges that are not on roads, in odd locations in the ocean, and near Svalbard east of Greenland. Some are plain errors in the bcs1 through bcs3 datasets, others are the result of wrong assumptions to estimate the location by combining it with the road database.

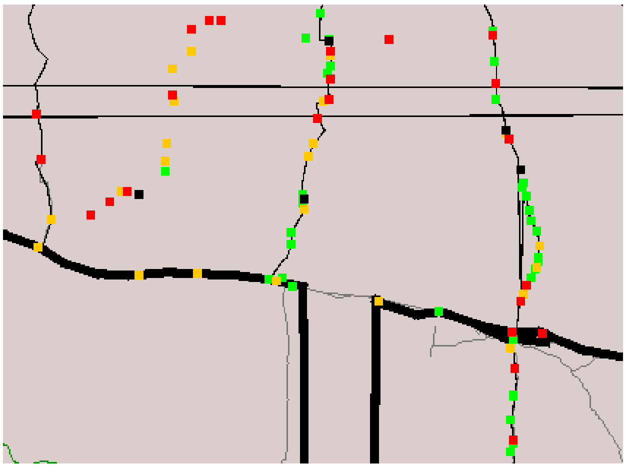
[1] https://en.wikipedia.org/wiki/Construction_surveying



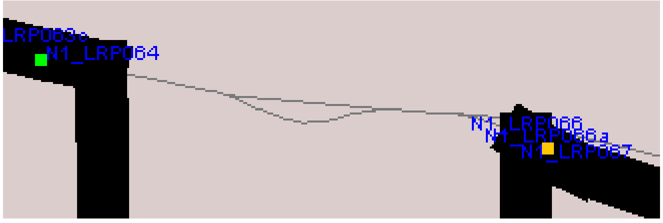
The road dataset **WBSIM/infrastructure/_roads.tsv** that is used as input to draw the N, R, and Z roads on the map as well as the LRPs also contains quite some errors. Possibly these errors have aggravated the location errors for the bridges when their locations were estimated based on the (wrong) road information.



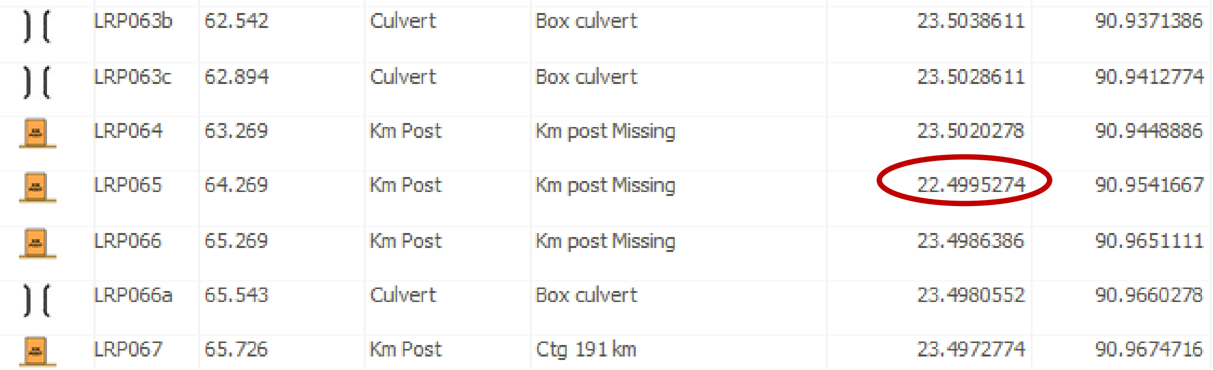
The above picture clearly shows that the road information is not perfect. Let's look at two examples:



The gap in the N1 road points to a coordinate that is a few hundred km south. It is between LRP64 and LRP66:



In the original database file RMMS/N1.lrps.htm we see:



)(	LRP063b	62.542	Culvert	Box culvert	23.5038611	90.9371386
)(	LRP063c	62.894	Culvert	Box culvert	23.5028611	90.9412774
▣	LRP064	63.269	Km Post	Km post Missing	23.5020278	90.9448886
▣	LRP065	64.269	Km Post	Km post Missing	22.4995274	90.9541667
▣	LRP066	65.269	Km Post	Km post Missing	23.4986386	90.9651111
)(	LRP066a	65.543	Culvert	Box culvert	23.4980552	90.9660278
▣	LRP067	65.726	Km Post	Ctg 191 km	23.4972774	90.9674716

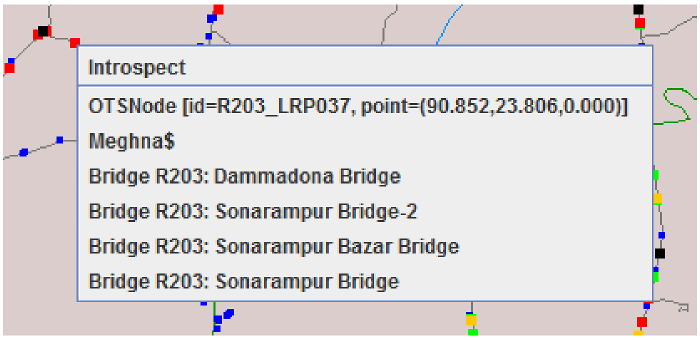
The 22.499 value is clearly out of line between the 23.49 and 23.50 values. Probably a typo.

There is a second problem in the picture at the top of the page: the Z-road on the top-left side of the picture has not been drawn (should be a black line). The bridges are on the Z0142, which is the road in the top-left corner of the picture.

Many more inconsistencies can be found when studying the map.

Legend for the map

The Java application draws the LRPs as blue squares, and bridges as green (quality A), orange (quality B), red (quality C), or black (quality D) squares. You can click on bridges or LRPs to ask for their information. Alternatively, ids (labels) can be shown. Zoom in to see the labels separately, and hide layers if they obscure the information you want to see. The 53 most important rivers are drawn as red lines where the widest lines represent the most important rivers. Black lines indicate roads. The widest ones are N-roads, the smallest the Z (Zila) roads.



Introspect
OTSNode [id=R203_LRP037, point=(90.852,23.806,0.000)]
Meghna$
Bridge R203: Dammadona Bridge
Bridge R203: Sonarampur Bridge-2
Bridge R203: Sonarampur Bazar Bridge
Bridge R203: Sonarampur Bridge

When you want to inspect an element, click on it with the right mouse button, and you get a selection list. Click with the left mouse button to select something from the list. When you have too many elements visible on your screen, the selection list might not show. In that case, close a number of layers at the top of the layer list first.

Assignment and Deadline

The Java program is available on BrightSpace; the name of the Zip file is WBSIM_Lab1.zip. **You need to run the program with Java.** Run in a terminal window with the command: **java -jar wbsim.jar**

Study the inconsistencies in the map regarding the roads and the bridges. **Develop a good understanding of the data**. You need this understanding to complete this assignment and the assignments that follow. Analyse the data quality issues in a structured way. To do this meaningfully and strategically, check the Assignments (2, 3, and 4) you will get in the following weeks regarding how the data will be used at later stages. Identify strategies for how you can solve the data quality issues. Prioritise the issues with sound reasoning. Implement algorithms (as much as time allows) that can automatically solve the data quality issues without changing the original files. You can either base the data cleaning on the derived files **BMMS_Overview.xlsx** and **_roads.tsv,** or on the original datasets BMMS and RMMS, or both.

Given the limited amount of time available, it is not advised to use the shape-file information or other external information, e.g., from Google Maps. In addition, focus on the major data quality issues you found and limit the complexity of the algorithm you design. The cleaning procedure of the algorithm(s) shall **generate a new BMMS_Overview.xlsx** and **_roads.tsv** file that you can test with the Java wbsim.jar program.

For your convenience, a cleaned dataset as well as the used data cleaning algorithms and logs are provided on BrightSpace for inspiration. This is a 95% solution (where still some improvements can be made on individual roads, and for some of the bridges). But as the RMMS and BMMS datasets are inconsistent in several fundamental aspects, some inconsistencies will always remain.

For this assignment, you shall decide which data quality issues you want to solve with code implementation and design your own solution. You can use Python (preferred), R, Excel, MATLAB, C, C# or combinations, or whatever program(s) such as GIS packages you want to do the cleaning.

Hand in the "EPA133a-Gxx-A1" Zip file with your program, the cleaned data files generated by your group (if any) and a short report in pdf on the data quality issues (as presented in lecture 2.1) in the original dataset, prioritization of data issues, what data quality issues you chose to repair and how, and a short reflection. The report should have a length of about 6-10 pages (this means 3000-5000 words), excluding images and references. Upload the Zip file to Brightspace following the Submission Guidelines.

Time to spend and Support

There are 8 lab hours dedicated to completing the lab assignment. In addition, you are each expected to spend another **4 hours maximally** per person on carrying out the exercise. Don't overspend your hours and see how far you can get with data quality analysis in 12 hours total. You can already get a passing grade if you analyse, prioritise data quality issues, and present and discuss the conceptual strategies of how you can solve the issues. Of course, this also depends on the quality of your analysis and solutions. Divide the work well within your group, and make sure you use the available hours of all team members combined well, through good collaboration and communication.

The deadline for handing in the Zip file of Assignment 1 on Brightspace is Friday in week 2 at 18:00.

Only upload using the Assignment function, don't use the File Locker or email to hand in – we will base the grading on what you hand in as Assignment 1 on Brightspace.