AVDTA: Dynamic Traffic Assignment

Michael W. Levin

July 25, 2016

Table of Contents

Chapter 1

Project structure

1.1   Introduction

AVDTA is organized around *projects*. A project is best thought of as a specific scenario under study. For instance, a project might represent a certain demand scenario or network configuration for a city network or subnetwork. The AVDTA GUI contains methods to modify project options, such as link types or intersection controls. Data files can also be copied to and from Excel for easy modification.
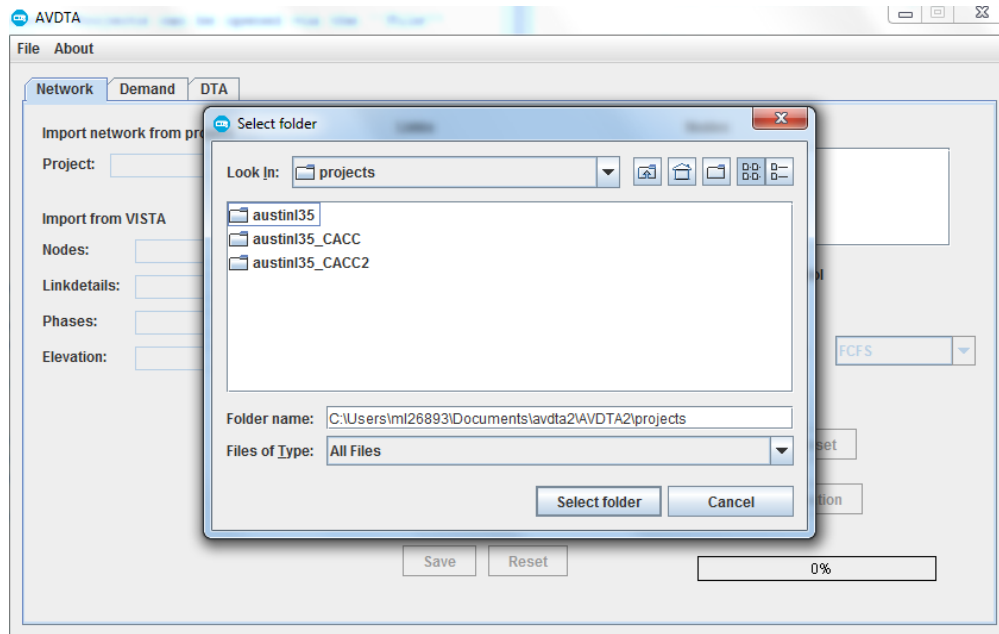
1.2   GUI

The AVDTA GUI is designed to interact with projects. Each instance of AVDTA can have a single project open at a time. Projects can be created and opened via the "File" menu.
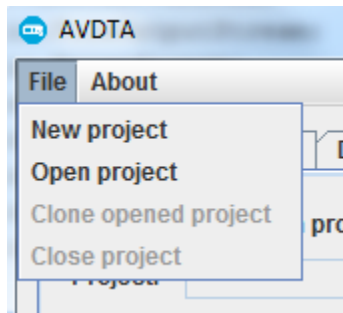
1.2.1   New project

To create a new project, you will first be asked to select the root folder. By default, this folder is `avdta/projects`, but it may be changed.
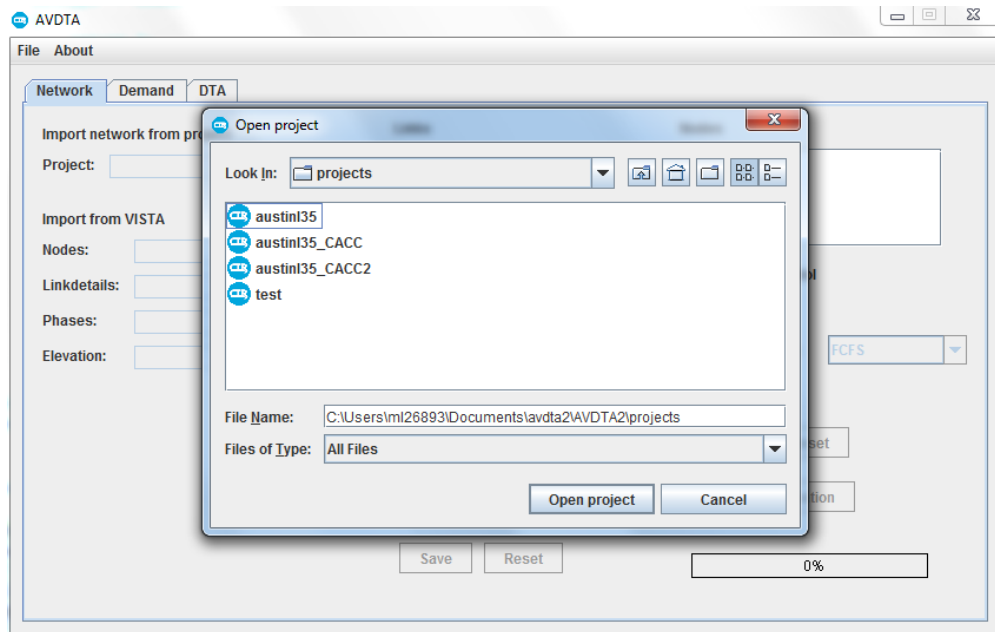
You will then be asked to enter the name of the new project. A project folder with the entered name will be created in the selected root folder.
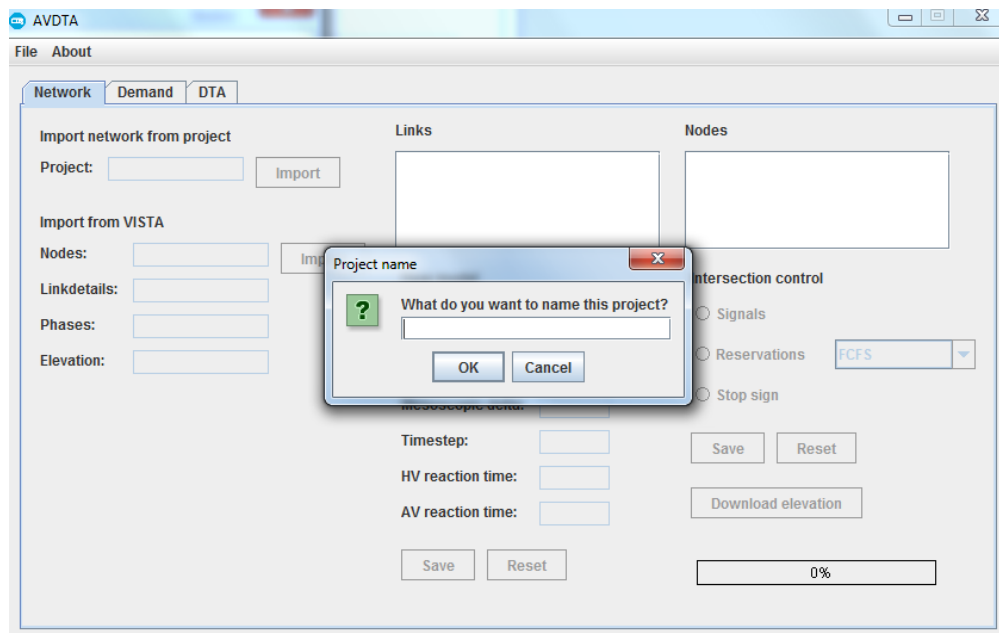
1.2.2 Open project

AVDTA project folders are shown with a special icon. Select a project folder and click "open project" to open it.

1.2.3 Clone opened project

If a project is open, the AVDTA GUI enables the option of creating a clone. Follow the instructions in Section 1.2.1 to choose the location and name of the clone. You can also create a copy of a project folder within the file system.

1.3 Files and folders

A project consists of a file folder containing several specific files and subfolders. The project folder, along with many of its files and subfolders, is automatically generated when a new project is created. However, many files will be empty, requiring the import of data from other projects or other sources. Note that removal or modification of these files outside of the AVDTA GUI could result in errors when loading the project. However, adding files or folders will not affect the project.

Projects contain several types of files. Text files (`.txt`) contain project inputs or outputs and are intended to be read or modified. However, if modifying these files, ensure that the format and units are correct. Text files are tab-delimited, and have a header indicating the data in each column. Text files may be copied to and from Excel.

Data files (`.dat`) are used to load the project and are not intended to be opened or modified. Similarly, files with other unknown extensions are not intended to be opened.

1.3.1 Files

1.3.1.1 `project.txt`

The `project.txt` file contains project properties used to load the project within AVDTA. The file consists of two columns: The file consists of two columns:

key	value

name   This is the project name that is displayed when AVDTA loads a project.

seed   This is the random number generator seed. If two projects have matching seeds, actions involving random numbers performed in the same order should have identical outputs.

type   This denotes the project type.

1.3.1.2 `options.txt`

The `options.txt` file contains project parameters that define the loading and simulation of the project. The file consists of two columns:

key	value

Keys are case insensitive. Values are a string that could represent multiple data types, such as integers, floating-point numbers, or booleans ("true" or "false"). The options file is automatically generated with default values when a project is created.

av-reaction-time   This is used to determine the capacity and congested wave speed increase due to autonomous vehicles when using the multiclass CTM [7]. A typical value is 0.5 (s). Note that capacity and congested wave speed are scaled from the values in the `networks/links.txt` file.

hv-reaction-time   This is used to determine the baseline capacity and congested wave speed when using the multiclass CTM [7]. A typical value is 1 (s). Note that capacity and congested wave speed are scaled from the values in the `networks/links.txt` file.

dynamic-lane-reversal   If set to "true", DLR will be activated on DLR links. DLR is a specific type of CTM link, and the type of links can be set in the `networks/links.txt` file.

hvs-use-reservations   If set to "true", HVs will not avoid reservation-controlled intersections in their route choices. Reservations will use the legacy early method for intelligent traffic management [2], adapted to DNL [7] for HVs. Otherwise, HVs will avoid reservations in their route choices, passing through reservations only if no other route is available.

4

simulation-duration   This is the duration of the simulation, in seconds. The duration should be sufficiently longer than the demand departure times interval to allow all vehicles to exit the network.

simulation-mesoscopic-step   This is the time step used in simulation. For CTM, a typical value is 6 (s). For LTM, a typical value is 10 (s).

1.3.1.3 `dta.dat`

The `dta.dat` file marks the project as a DTA project, and should not be modified.

1.3.1.4 `paths.dat`

The `paths.dat` file contains a list of all known paths in the network. AVDTA does not generate a complete list of paths, but all paths that are created during DTA are saved here. The file is used when loading and saving assignments.

1.3.2   Folders

network folder   The `network` folder contains all network data files, discussed in Chapter ??.

results folder   The `results` folder is used to store and organize results files.

demand folder   The `demand` folder contains all demand data files, discussed in Chapter 3.

assignments folder   The `assignments` folder stores previous assignments for the project. Each assignment consists of a subfolder, usually named by the time at which the assignment was created. Previous assignments can be loaded through the AVDTA GUI.

Each assignment subfolder contains several files. The `log.txt` file contains a log of the DTA run that created the assignment. It includes results from each iteration of DTA as well as summary statistics for the assignment. The `vehicles.dat` file contains summary statistics in a more readable form for AVDTA, and then a list of vehicles and the path they are assigned to. The path ids correspond to a path in the `paths.dat` file. Do not modify the `vehicles.dat` file.

Chapter 2

Traffic network

2.1 Introduction

A *network* in AVDTA defines the

Chapter 3

Demand

Appendices

Appendix A

Abbreviations

Abbreviation	Definition
AV	autonomous vehicle
CV	connected vehicle
CTM	cell transmission model [3, 4]
DLR	dynamic lane reversal [5, 6]
DNL	dynamic network loading [1]
DTA	dynamic traffic assignment [1]
HV	conventional (human-driven) vehicle
GUI	graphical user interface
LTM	link transmission model [8, 9]

References

[1] CHIU, Y.-C., BOTTOM, J., MAHUT, M., PAZ, A., BALAKRISHNA, R., WALLER, T., AND HICKS, J. Dynamic traffic assignment: A primer. *Transportation Research E-Circular*, E-C153 (2011).

[2] CONDE BENTO, L., PARAFITA, R., SANTOS, S., AND NUNES, U. Intelligent traffic management at intersections: Legacy mode for vehicles not equipped with v2v and v2i communications. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on* (2013), IEEE, pp. 726–731.

[3] DAGANZO, C. F. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological 28*, 4 (1994), 269–287.

[4] DAGANZO, C. F. The cell transmission model, part ii: network traffic. *Transportation Research Part B: Methodological 29*, 2 (1995), 79–93.

[5] DUELL, M., LEVIN, M. W., BOYLES, S. D., AND WALLER, S. T. On system optimal dynamic lane reversal to improve traffic efficiency for autonomous vehicles. In *Transportation Research Board 95th Annual Meeting* (2016), no. 16-4922.

[6] LEVIN, M. W., AND BOYLES, S. D. A cell transmission model for dynamic lane reversal with autonomous vehicles. *Transportation Research Part C: Emerging Technologies 68* (2016), 126–143.

[7] LEVIN, M. W., AND BOYLES, S. D. A multiclass cell transmission model for shared human and autonomous vehicle roads. *Transportation Research Part C: Emerging Technologies 62* (2016), 103–116.

[8] YPERMAN, I. The link transmission model for dynamic network loading. *status: published* (2007).

[9] YPERMAN, I., LOGGHE, S., AND IMMERS, B. The link transmission model: An efficient implementation of the kinematic wave theory in traffic networks. In *Proceedings of the 10th EWGT Meeting* (2005).