Vehicle Grid Integration Planning: User Guide and Technical Overview

This repository contains Python code developed for the research paper: "Vehicle grid integration planning tool: Novel approach in case of Tokyo."

Citation

If you use this code, please cite our paper:

[Reza Nadimi, Mika Goto, "Vehicle grid integration planning tool: Novel approach in case of Tokyo", Applied Energy, Volume 399, 2025, https://doi.org/10.1016/j.apenergy.2025.126509.]

Implementation Procedure

Figure 1 presents the structure of the Vehicle Grid Integration (VGI) Planning Tool, comprising eight Python scripts and a configuration folder containing 16 Excel files. Users can modify these input files to customize simulation scenarios. The tool is designed to be data-driven and modular, allowing flexibility in vehicle fleet size, temporal resolution, and modeling assumptions.

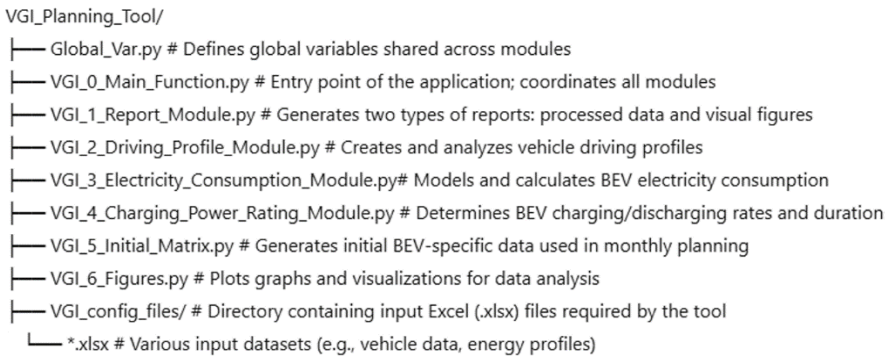

Figure 1: Tree Structure of VGI Planning Tool with descriptions

To utilize the VGI Planning Tool, the following steps are required:

1. Install Dependencies

Ensure that all required Python libraries are installed. Use the following command:

pip install -r requirements.txt

2. Prepare Input Data

Download the ZIP archive from the repository's Code section. After extraction, verify that the "VGI_config_files" folder contains 16 Excel (.xlsx) files. These files provide the necessary configuration and input datasets for the simulation.

3. Execute the Simulation

Run the main script "VGI_0_Main_Function.py". This script serves as the entry point and coordinates the execution of all other modules.

User Configuration

Within "VGI_0_Main_Function.py", users can define key simulation parameters, including:

- The number of battery electric vehicles (BEVs)
- The simulation time horizon (number of months)
- Flags controlling model behavior (described below)

Upon execution, the application creates:

- One output folder per selected month (e.g., "Output_VGI_Files_1" corresponding to January, please see Figure 2)
- A "Figures" folder containing visual outputs

Each "Output_VGI_Files_<Month>" file includes nine CSV files summarizing regional and zonal outputs such as BEV charging and discharging profiles, electricity consumption, and system-level analytics. The Figures folder contains six plots corresponding to Figures 7–12 in the associated publication.

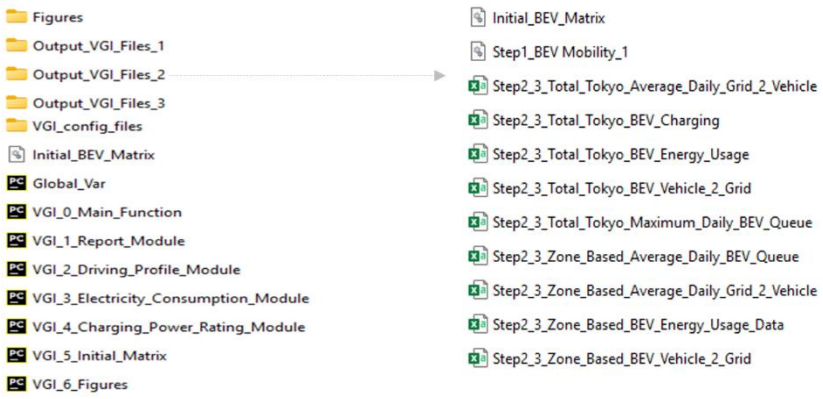

Figure 2: Left: Project folder after a 3-month simulation; Right: Generated output CSV files

Model Control Flags

The main script includes four key flags defined in "VGI_0_Main_Function.py", which control optional functionalities as described in table below.

Variable	Explanations
GV.Trip_Module_Run_Flag	If set to True, the model generates new BEV trip data.
GV.Model_V2G_Service_Flag	If set to True, the model includes V2G services, invoking the optimization module.
GV.Unmanaged_Charging_Flag	Enables simulation of unmanaged charging behavior when this flag is set to True.
GV.Initial_Matrix_Existance_Flag	If set to True, the model uses pre-existing BEV driving data. If set to False, it generates synthetic data on BEV home/work locations, commuter types, and more.

Performance Considerations

Simulations involving V2G functionality significantly increase computational time due to the embedded optimization model. For instance, a full-year simulation (12 months) with 58,900 BEVs and V2G enabled may require over 120 hours of processing time on standard computing resources.