IDSO-Managed Bid-Based Transactive Design for DER Participation in Wholesale Markets While Preserving T–D Interactions*

Swastik Sharma†, Swathi Battula‡ and Sri Niwas Singh§

1 Introduction

This repository contains all relevant code, data files, scripts, and documentation for the work carried out in [1]. As described in greater length in [1], the methodology and other related things are omitted from this document. In the next section, we will provide an overview of the repository, including all the folders and files present in the root directory.

2 Overview of the Repository

The root directory, `IDSOAggregationProblem`, and its containing folders are shown in the following diagram:
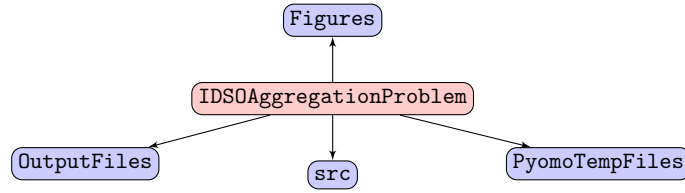
1. `src`: This is the brain of the whole application. It contains all the `.py` files and `.bat` files necessary for running the Python program.
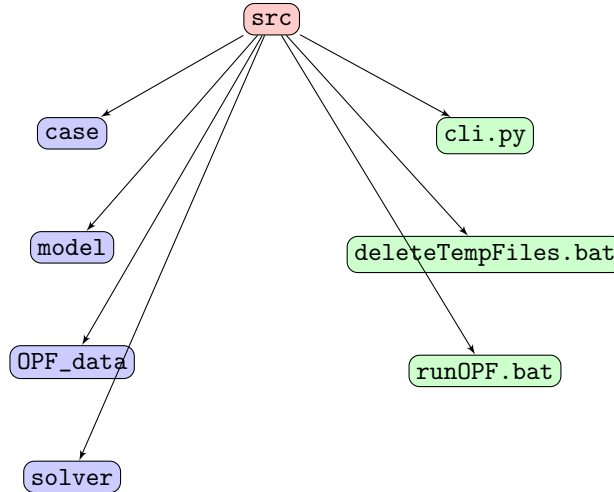
   (a) `case`: This folder contains readers to read a matpower based `.m` files. You can safely ignore this if you wish to use `.csv` files directly to input data. The use of `.m` files have not been extensively explored in this program, but interested individuals can utilise these helper files to integrate them.

(b) `model`: This folder contains all the `.py` files for running the pyomo-based optimisation framework.

(c) `OPF_data`: This folder contains all the `.csv` or `.xslx` files necessary to run the OPF. The folder contains all the files for an IEEE 123 bus system and a simple 4 Bus Test Case. Using the same format, one can incorporate any distribution system for analysis.

(d) `solver`: This folder has the `.py` programs to initialise the solver related libraries. If required, the `results.py` program is used to obtain output and dual variables from the Pyomo optimiser.

(e) `cli.py`: This Python file runs the entire program. It initialises the DATA and OPF model and stores the results in `.txt` or `.csv` format in the OutputFiles folder.

(f) `deleteTempFiles.bat`: This batch file deletes Pyomo's temporary files in the PyomoTemp-Files folder.

(g) `IDSO.py`: This Python file contains all the code to model IDSO and DERs. The DERs are randomly allocated across the feeder buses, and their bids and offers are randomised.

2. `OutputFiles`: This is used to store the results from OPF, such as the voltage magnitude at each node, line power flows, etc., in both `.txt` and `.csv` files for each run.

3. `PyomoTempFiles`: This contains the temporary files generated by Pyomo.

3 Running the Program:

The program parses arguments from the user to initialise data. For example, to run the file, the following command needs to be executed as shown in Fig. 1.:

```
python cli.py --LMP 13 --M 1000 --C0 2.5 --pf 0.9 --seed 85 --TestCase IEEE123TestCase
```

A description and the default values these arguments take are provided in Table 1. These default values are similar to the ones used in the paper for the IEEE 123 Bus Test Case.

Table 1: Description of the arguments a user needs to provide. For more details, please see [1].

Argument	Description	Default Value
`--LMP`	LMP value in cents per kWh	13
`--M`	Used to initialise the offer values for IDSO's T-DOPF Objective Function	1000
`--C0`	IDSO marginal cost in cents per kWh	2.5
`--pf`	DERs' operational power factor	0.9
`--seed`	DERs are randomly initialised this helps protect reproducibility	85
`--TestCase`	This selects the Test Case to use from the `OPF_data` folder	IEEE123TestCase

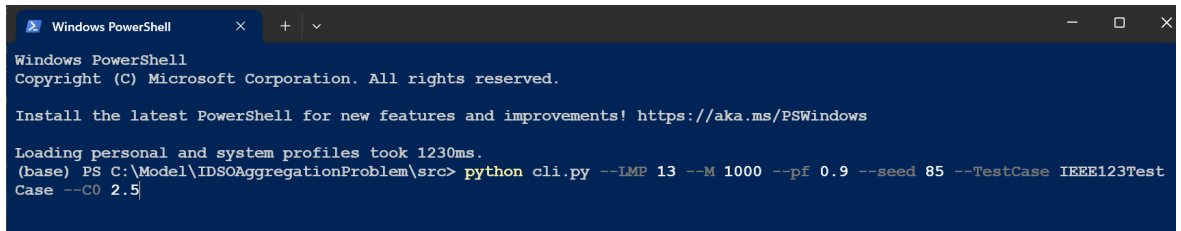Figure 1: Command executed on a command line interface.

References

[1] S. Sharma, S. Battula, and S. N. Singh, "IDSO-Managed Bid-Based Transactive Distribution Systems Design for DER Participation in Wholesale Markets While Preserving T-D Interactions," 2025. [Online]. Available: https://arxiv.org/abs/2508.08187