RFLP_code

Overview

Electronic companion of the paper: *Data-Driven Reliable Facility Location Design* by

- Hao Shen: School of Business, Renmin University of China, Beijing, China, shenhao@rmbs.ruc.edu.cn
- Mengying Xue: International Institute of Finance, School of Management, University of Science and Technology of Chinaa, Hefei, Anhui, China, xmy2021@ustc.edu.cn
- Zuo-Jun Max Shen: Faculty of Engineering; Faculty of Business and Economics, The University of Hong Kong, Hong Kong, China maxshen@hku.hk

This code was tested on:

- Ubuntu server equipped with 20 processors and 40G RAM.
- Gurobi 9.5 with Python API

Folders and Related Scripts

- `Data/` : Network data from Synder and Daskin (2005) [source](Larry Snyder » Data Sets for “Reliability Models for Facility Location: The Expected Failure Cost Case”), processed weather data from NOAA from 1950 to 2021 in json file, [source](Storm Events Database | National Centers for Environmental InformationStorm Events Database | National Centers for Environmental Information; folders for synthetic data
  - RawData: store the generated synthetic data based on randomly generated distributions
  - RawDataFileName
  - RawDataDistribution
  - RawDataStorm
  - RawDataStormFileName
  - Storm:
    - `disruption_49.json` : synthetic data file for the network with 49 nodes with disruption states using the data from NOAA. We refer readers to Section 6 in the manuscript to see the data processing of the weather data. Each row represents one historical record of weather hazard, and each column represents the state (disruption or not) of the 49 locations on the network.
    - `StormEvents_details_Begin_1950_End_2021.csv` : the aggregated data of the historical records downloaded from NOAA. Used for generating `disruption_49.json` .
- `DataRelated/` : Python code for preprocessing raw data and generating synthetic data used in numerical studies.
  - `DataGenerate.py` : load the information from Synder and Daskin's data set, generate synthetic data for the numerical experiments
  - `DataGenerateStorm.py`

    - `DataProcess.py`
    - `StormDataProcess.py`
- `MomentBased/` : Python code for marginal-moment method in [Lu et al. (2015)](Reliable Facility Location Design Under Uncertain Correlated Disruptions | Manufacturing & Service Operations Management) and cross moment method in [Li et al. (2022)](A General Model and Efficient Algorithms for Reliable Facility Location Problem Under Uncertain Disruptions | INFORMS Journal on Computing)
    - 1
- `results/` : optimization results
- `Wasserstein/` : Python code for optimizing the RFLP using type-infinity wasserstein DRO in [Xie (2020)](Tractable reformulations of two-stage distributionally robust linear programs over the type-∞ Wasserstein ball - ScienceDirectTractable reformulations of two-stage distributionally robust linear programs over the type-∞ Wasserstein ball - ScienceDirect)
- `PUB/` : Python code for RFLP using PUB estimator proposed by Data-driven Reliable Faciltiy Location Problem.
- `SampleAverage/` : Python code for RFLP using sample average approximation method.
- `Reliability/` : Python codes for testing reliability and generating average performances of RFLP using PUB estimator, Wasserstein DRO, and moment based methods.
- `Utility/` : Python code for utility functions and constants used in the experiments
- `Plot/` : Python code for the plots in the paper Data-driven Reliable Faciltiy Location Design.

Scripts and Modules

Run each of the following scripts, and the result file is in the format of `.csv` and will be stored in the folder `/results/`

- `data_generate.py` : generate synthetic data and case study data
- `run_PUB.py` : generate performances for PUB estimators in both data with covariates and no covairates
- `run_moment.py` : generate performances for marginal moment and cross moment in both data with covariates and no covairate case
- `run_wass.py` : generate performances for RFL with type-infinity wasserstein DRO method.
- `run_PUB_Storm.py` : generate the performance of PUB estimator with covariates in the case study using the weather data from NOAA
- `run_moment_storm.py` : generate the performance of marginal and cross moment based methods with covariates in the case study using the weather data from NOAA
- `plot_fig.py` : generate plots comparing different methods in the paper

Syntheic Data Generation

To generate the data, run function `GenerateRawDataFileList` in `data_generate.py`. Generated files will be stored in /Data/RawData. The distribution information (including

marginal prob. for each covaraites, covariate based cov matrix, covaraite based means, and covariate based lower and upper bounds) on supplies and demands will be stored in /Data/RawDataDistribution; file names are stored in /Data/RawDataFileName.

Storm Data Generation

To generate the data in the case study, run function `RawDataGenerateStormFileList` in the `data_generate.py` (see the commented function in the file). The generated data files will be stored in `Data/RawDataStorm` with the file names lists stored in `Data/RawDataStormFileName`.