

Data and Codes used in manuscript

“Robust Workforce Management with Crowdsourced Delivery”

August 2, 2024

Don’t hesitate to get in touch with **Chun Cheng** (chun.cheng@polymtl.ca) and **Yue Zhao** (yuezhao@u.nus.edu) if you have any questions when reproducing the results.

To reproduce the results, the Gurobi license (<https://www.gurobi.com/>) is needed, along with other libraries used in the Python files (e.g., numpy, math, matplotlib, etc.). Readers also need to install the RSOME package (<https://xiongpengnus.github.io/rsome/>) to reproduce the results of the moment-based distributionally robust (DRO) model in Appendix C.

The folder includes two sub-folders:

- **Code_Solomon**: the data and codes for evaluating the performance of the basic and generalized reduced information models, i.e., the results in Table 1, Figure 2, and Figure C1.
- **Code_Simulation**: the codes for comparing the robust satisficing, the empirical, and the moment-based DRO models, i.e., the results in Figures 3–5 and C2.

1 Details about the sub-folder “Code_Solomon”

***** data and results *****

→ **Folder “SolomonInstance”**: the data set. Specifically, file “0-Xtype.txt” includes each node’s coordinates of Solomon’s type X instances. The three sub-folders “typeC”, “typeR”, and “typeRC” include the detailed solutions of each Solomon instance. The sub-folder “driverData” contains ad-hoc couriers’ information.

→ **Folder “NewResult_Dist”**: the results of Table 1 and Figure 2.

→ **Folder “NewResult_RandBid”**: the results of Figure C1.

***** conduct experiments to evaluate the reduced information models *****

→ **main.py**: run this python file to perform all the experiments.

→ **Function.py**: functions used to read files, generate ad-hoc couriers’ data, select jobs for ad-hoc couriers based on their utilities, etc.

→ **CalculateCostFunc.py**: implement the true and the reduced information models to get the corresponding cost values.

***** draw figures *****

→ **figure.py**: draw Figures 2 and C1.

2 Details about the sub-folder “Code_Simulation”

→ **main.py**: run this python file to perform all the experiments.

***** conduct experiments to evaluate the robust satisficing model *****

→ **run_main.py**: conduct the experiments presented in Section “Comparison of out-of-sample performance”, i.e., the results in Figures 3 and 4.

→ **run_util_mode.py**: conduct the experiments presented in Section “Investigation on different schemes of ad-couriers’ expected payments”, i.e., the results in Figure 5.

→ **run_RSOME.py**: implement the moment-based DRO model in Appendix C, i.e., the results in Figure C2.

→ **script*.py**: run the corresponding experiments in parallel.

***** draw figures using jupyter notebook *****

→ **demo_main.ipynb**: draw Figures 3 and 4.

→ **demo_UtilityMode.ipynb**: draw Figure 5.

→ **demo_RSOME.ipynb**: draw Figure C2.

***** folders *****

→ **Folder “results”**: all the numerical results.

→ **Folder “src”**: a python file named “simu.py” is stored here. This file includes:

- a class ‘COURIER’ for simulating ad-hoc couriers’ dynamics and behaviors;
- a class ‘DEMAND’ for generating jobs at each period;
- a class ‘simulation’ for simulating couriers’ bidding process, implementing the assignment model, generating training samples and testing samples, etc.;
- a class ‘optimization’ for implementing all the optimization processes of our models, preparing breakpoints, solving the empirical model, implementing the binary search algorithm to solve the robust satisficing model, etc.