Code for “Optimizing Self-Organized Volunteer Efforts in Response to the COVID-19 Pandemic”

#### Authors: Anping Zhang1†,Ke Zhang1†, Wanda Li1, Yue Wang2,Yang Li1∗, Lin Zhang1

**NCE computation**: Compute self-organizational intervals on Shenzhen’s data

Note:

1. O\_NCE.csv. T\_NCE.csv and P\_NCE.csv are pre-computed NCEs for Shenzhen and its district using data files “issuer\_task\_data.csv” and “issuer\_user\_data”.
2. The “task label” column in organizer\_task\_data.csv represents the task type extracted from task descriptions using LDA.
   1. Label 1: Transportational Topic tasks;
   2. Label 2: volunteering topic tasks;
   3. Label 3: Reopening Topic tasks;
   4. Label 4: Educational topic tasks;
   5. Label 5: environmental topic tasks;
   6. Label 6: Covid-19 topic tasks.
3. “neigborhood\_1.csv” and “neighborhood\_2.csv” are data for case studies.

To run:

1. Run NCE.ipynb to generate an NCE plot with color shaded self-organization intervals

**Causality Analysis**: Causality analysis on what dynamic factors have caused self-organization events.

Note:

1. causality\_data.csv contains three types NCE, internal and external variables (policies impulse and covid-19 daily new cases)
2. all\_diff\_data.csv is differencing from causality\_data.csv to make sure our time-series data is stationary for causality analysis

To run:

1. Install tigramite package from <https://github.com/jakobrunge/tigramite.git>
2. Install graphviz package from https://graphviz.org/download/
3. Run Causality\_analysis.ipynb to obtain full causal graphs for self-organization intervals

**Simulation part**: A simulation of users participating in a fixed number of tasks.

Simulation rules:

1. Simulation is initialized with a fixed number of agents and tasks

1. Each task is represented by a cell in a 2D grid. All tasks have a limit on

the number of agents it can recruit: `max\_agent\_per\_cell`;

2. At each step, each agent decides whether to participate in a task with

probability, `p\_participate.`

3. If the agent is participating, it will join the first available task

from its recent participation history within a time window, ordered by highest frequency;

(To simulate the user behavior of participating in the same task.)

If no space is available for all these tasks, it will join a random available task nearby the current task.

To run:

1. Install mesa package from https://mesa.readthedocs.io/

2. Run UserModel.py to run the simulation

3. paint\_subplot\_simulation.py is used to draw NCEs and gains under different parameters. People can change the parameters in UserModel.py to get NCE and gains.