Fair Matching under Constraints: Theory and Applications Replication

Overview

The data in this replication package includes simulated matching results from our code. The code in this replication package simulates SOFM and serial dictatorship and generates data for the 4 figures and 4 tables in the paper.

Data Availability and Provenance Statements

☐ This paper does not involve analysis of external data (i.e., no data are used or the only data are generated by the authors via simulation in their code).

If box above is checked and if no simulated/synthetic data files are provided by the authors, please skip directly to the section on Computational Requirements. Otherwise, continue.

Statement about Rights

☑ I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.

License for Data

Data Citation

`Kamada, Yuichiro, and Kojima, Fuhito.(2021).Replication package for:Fair Matching under Constraints:Theory and Applications.Zenodo. https://doi.org/10.5281/zenodo.4516857`

Summary of Availability

☐ All data **are** publicly available.
☑ Some data **cannot be made** publicly available.
☐ **No data can be made** publicly available.

Details on each Data Source

The original data for this project are confidential, but may be obtained by contacting the city of Yamagata and Bunkyo.

Dataset list

Data file	Source	Notes	Provided
`data/bunkyo-data.xlsx`	Bunkyo City	Confidential	No
`data/yamagata-data.xlsx`	Yamagata City	Confidential	No

Computational requirements

Software Requirements

- R

R Packages Used

- readxl
- readr
- here
- tidyr
- dplyr
- futile.logger
- ggplot2
- stringr

Tip on how to install R packages: you can call install.packages() to install a package in R.

Memory and Runtime Requirements

Summary Approximate time needed to reproduce the analyses on a standard (CURRENT YEAR) desktop machine:

- ☐ <10 minutes
- ☐ 10-60 minutes
- ☐ 1-6 hours
- ☐ 6-12 hours
- ☑ 12-72 hours
- ☐ 3-14 days
- ☐ > 14 days
- ☐ Not feasible to run on a desktop machine, as described below.

Details Portions of the code were last run on a **4-core Intel-based desktop with Windows 10**.

Description of programs/code

- Programs in `src` will clean original data, generate simulated outcomes for SOFM and serial dictatorship, and do some analysis on the data.

License for Code

Instructions to Replicators

- Run `src/R/random_tie_break.R` to simulate matching with serial dictatorship.
- Run `src/SOFM_C/SOFM_C/sofm.cpp` to simulate matching with SOFM. Make sure the input data format is right. You can use `src/c_preprocess.R` to prepare the input data and `src/decode_matching_fromC.R` to decode outputs from sofm.cpp.
- Run programs under `src/analysis` and `src/random_break_ties` to compare matching assignments, number of matches, distribution of matches, and envy for each matching rule.

List of tables and programs

The provided code reproduces:

☐ All numbers provided in text in the paper
☐ All tables and figures in the paper
☑ Selected tables and figures in the paper, as explained and justified below.

For Table 1-4, their data is generated from corresponding programs listed below. Those numbers in the tables are collected by running corresponding programs under different scenarios respectively. The programs don't generate tables directly (they don't need to) since we draw tables in Latex.

Figure/Table #	Source
Table 1 & 3	src/random_break_ties/comparetwo_random.R
Table 2 & 4	src/random_break_ties/envy_random.R
Figure 1 &3	Generated from data results
Figure 2 &4	src/random_break_ties/errorbar_dist.R The output for figure 2 is output/compare_results/yamagata/T_W_AD_AE/All_cum_error0_sim250.png. The output for figure 4 is output/compare_results/bunkyo/V/All_cum_error0_sim250.png.