🌐 A Simple Exploration of Simulated Annealing: Protocol

Chasz Griego[1]

[1]Carnegie Mellon University

Ⓒ Chasz Griego

AUG 04, 2023

ABSTRACT

This protocol describes technical steps to understand the computational environment for A Simple Exploration of Simulated Annealing hosted on Code Ocean.

SAFETY WARNINGS

> ❗ Code can take a considerable amount of time. Be cautious when modifying.

Protocol status: Working
We use this protocol and it's working

Reproducing the Data Analysis

1   Access the capsule for this project at [Code Ocean](#). Select "**Edit Capsule"** to work with your own copy.

2   The analysis was completed with Python in a Jupyter Notebook. The following notebook is found in the **code** folder in the file manager:

```
analysis.ipynb
```

2.1   Details about the computational environment, including the versions of Python, Jupyter, and other libraries are found in the **environment** section of the file manager, which contains a Dockerfile.

2.2   Data for the analysis is found in separate folders in the **data** folder in the file manager:

- **grid_sample**: A .npy file of the sample grid used in simulated annealing
- **separated_grids**: Multiple .npy files for every grid resulting from a simulation
- **simulation_data**: Multiple .csv files with run details at every iteration of a simulation

3   To execute **analysis.ipynb**, select "**Reproducible Run**"

4 Once the reproducible run is complete, the resulting figures, an HTML copy of the executed **analysis.ipynb**, and a run output log are stored in the **results** section found in the file manager.

Running Additional Simulations and Producing New Data

5 Each simulation can be run again, with the same sample grid, by executing the following Jupyter Notebook found in the **code** folder:

```
production.ipynb
```

6 **WARNING:** Running **production.ipynb** will overwrite all of the data in each subdirectory of **data.**

6.1 Copy each subdirectory in a new folder with a title such as "**original**" or something that is more descriptive. Do this before each subsequent run of **production.ipynb**. Update file paths in **analysis.ipynb** if you wish to analyze data that was moved.

7 Modify line 10 of the **run** file inside the **code** section to select **production.ipynb** to run in a reproducible run:

```
--execute production.ipynb \
```

8 Select "**Reproducible Run**" to run additional simulations and produce new data.

9 To reanalyze the new data, modify line 10 of the **run** file to select **analysis.ipynb**.

> **Note**
>
> **WARNING:** Any files that are in **results** may be overwritten. Move these files into a new folder before proceeding.

10 Select "**Reproducible Run**" to execute a new analysis.

Running Simulations with Different Parameters

11 There are various ways to alter parameters to run additional simulations:

Step 11 includes a Step case.

New sample grid
New simulation

-- step case --

New sample grid

12 There are several parameters that can be changed in **production.ipynb** to produce a new sample grid:

- **rows:** number of rows in the grid
- **cols:** number of columns in the grid
- **percentage:** composition of 1's in the randomly generated grid of mixed 1's and 0's
- **seed:** defines the random number seed used to generate the random mixture of 1's and 0's

13 Change any of these parameters (individually or in combination), and after running **production.ipynb**, a new .npy file will be generated in **data/grid_sample.**

> **Note**
>
> **WARNING:** If the seed is not changed, the .npy file for the previous sample will likely be overwritten. Make sure to relocate or rename these files before running.

14 Repeat all steps in "**Running Additional Simulations and Producing New Data**" to produce new data from the new sample.