Reproducibility Report for ACM SIGMOD 2022 Paper: "Cooperative Route Planning Framework for Multiple Distributed Assets in Maritime Applications"

TIANZHENG WANG, Simon Fraser University, Canada

The provided code and scripts are covered all the needed experiments and are complete and usable. Key results were successfully reproduced using the scripts, with additional help from the original authors.

1 INTRODUCTION

The paper being reproduced is titled "Cooperative Route Planning Framework for Multiple Distributed Assets in Maritime Applications" which was published in SIGMOD 2022, by Sepideh Nikookar, Paras Sakharkar, Sathya Somasunder, and Senjuti Basu Roy from New Jersey Institute of Technology, Adam Bienkowski, Matthew Macesker and Krishna Pattipati from University of Connecticut, and David Sidoti of Navy Research Lab [1].

The reproducibility test included experiments comparing seven models that correspond to the experiments done by the original paper. Overall, the reproduced results showed similar trends compared to what was reported by the original paper, although the reproducibility process is manual, and we used additional help from the authors.

2 SUBMISSION

The original submission itself is a package that provides instructions to reproduce the results. Brief readmes on how to train and run each model are given. The main omissions were the process instructions and scripts for repeating exact experiments as described in the paper. Later the reviewer and authors worked together to mitigate this issue (described later). It is suggested that the above process be automated as much as possible. Overall, the reproducibility effort needed manual intervention, but the submission gives all the needed ingredients for reproducing the results. There were some minor issues regarding getting certain models to be successfully trained and executed, but the authors provided quick and easy fixes that uncleared the roadblocks. Software dependencies were easy to satisfy, using default Python tools (e.g., `pip`) without issues.

3 HARDWARE AND SOFTWARE ENVIRONMENT

Experiments are conducted on a quad-socket server with configurations shown in Table 1 ("Repro Review"). The main differences are that the machine used (1) has a different CPU with more cores and sockets but runs at a lower frequency and (2) runs Linux instead of MacOS.

4 REPRODUCIBILITY EVALUATION

4.1 Process

Using the original submission, training and running the models were very straightforward, following the instructions in the submission and running scripts. Additional communications with the authors for clarifications on how to interpret the results and how to plot was needed, but the issues were minor and got resolved timely.

To produce the numbers for each figure/experiment, we needed the exact setup and parameters for each experiment, which was missing in the original submission, as mentioned earlier. Given the limited time, the reviewer and the authors worked together to restructure the submission format by organizing the scripts into individual directories, each of which is self-contained for generating a (sub)figure in the paper. The user then only needs to follow instructions to invoke model training

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	1 × Intel Core i5	2 ×Intel Xeon E5-4610
cores	4	10 × 4
GHz	2.4	1.8
RAM	16GB	512GB
Storage	Not applicable	Not applicable
OS	MacOS Big Sur	Arch Linux with kernel 5.19.5

and inference commands. This way, all the parameters are pre-set and it requires no manual change to generate raw numbers. The user should then parse the result outputs and organize them into proper table formats for plotting.

The plotting process was manual, and for the sake of time, the reviewer did not use the provided plotting source files and software (Veusz) but (1) directly computed the values from the code output and then (2) compared these results with those presented on the paper, by following additional instructions provided by the authors.

4.2 Results

The reviewer was able reproduce and verify the main claims made in the original paper using the aforementioned process. The results showed similar trends. The only caveat was that some of the experiments were not completely independent from each other. In particular, the *Baseline-1* and *RandomWalk* experiments use the output from `Approx-MaMoRL`. It would be ideal to have scripts that can simplify this process.

REFERENCES

[1] Sepideh Nikookar, Paras Sakharkar, Sathyanarayanan Somasunder, Senjuti Basu Roy, Adam Bienkowski, Matthew Macesker, Krishna R. Pattipati, and David Sidoti. 2022. Cooperative Route Planning Framework for Multiple Distributed Assets in Maritime Applications. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 1518–1527. https://doi.org/10.1145/3514221.3526131