Reproducibility Report for ACM SIGMOD 2023 Paper: "EAR-Oracle: On Efficient Indexing for Distance Queries between Arbitrary Points on Terrain Surface"

MADHULIKA MOHANTY, Inria & IPP, France
HUBERT MOHR-DAURAT, Imperial College London, UK
PRAJNA UPADHYAY, BITS Pilani Hyderabad Campus, India
BO HUANG, Southern University of Science and Technology, China

The provided code is reasonably easy to run but requires exactly specified hardware and software versions. The *UnFixedS* and *KAlgo* algorithms take undefined amount of time to finish. Thus, their results could not be reproduced in all the experiments. The results of only the small datasets could be fully reproduced due to the execution time and memory required for larger datasets. Overall, the reproduced results verify the core thesis of the paper and thus, the paper should be marked reproducible.

1 INTRODUCTION

In this paper [1], the authors propose a new indexing structure, *EAR-Oracle*, to efficiently calculate geodesic shortest distance, i.e., the length of the shortest path between two points on a 3D terrain data. In-depth experiments on various datasets from small to large size and with variations on diverse parameters show that this new indexing technique scales to larger data compared with other indexing techniques. In addition, *EAR-Oracle* outperforms the baselines that use online computations (i.e., no pre-calculated indexing) by orders of magnitude in terms of query time.

2 SUBMISSION

The authors provided as a public repository the code and datasets to reproduce the experiments at `github.com/ItakEjgo/weighted_distance_oracle`. This repository contains, in particular, the following content to guide the user:

- a `README.md` file containing detailed information about the file structure of the repository and instructions for installing dependencies, compiling the code and running the experiments
- a Python script file at `build/master_script.py` to automatize the execution of all experiments and the generation of the graphs with a single command to run
- a configuration file at `build/script_config` to tweak the parameters of the experiments

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 shows a comparison between the authors' original environment and the ones from the reviewers.

Table 1. Hardware & Software environment

	Paper	Repro Review #1	Repro Review #2	Repro Review #3
CPU	Intel Xeon Gold 5122	Intel(R) Xeon(R) Gold 5218	2x Intel Xeon Silver 4114	Intel(R) Xeon(R) Silver 4114
cores	4	32	10	20
GHz	3.60	2.30	2.20	2.20
RAM	256GB	187GB	196GB	256GB
OS	Linux	Linux	Linux	Linux

4 REPRODUCIBILITY EVALUATION

4.1 Process

The datasets fall into three categories: small, medium and large, which can be each turned on or off through the configuration file. We reproduced the experiments fully only for the small datasets, as the largest datasets require high execution time for some of the baselines (several days for the medium and several weeks for the large datasets). In addition, the large dataset requires up to 256GB of RAM for some experiments.

The authors were responsive in providing answers to our questions regarding the datasets and the experiments. They have also actively improved the code during the review period to fix various minor issues for executing the experiments (changes that they submitted to the public repository).

Following the instructions in the README file, we installed the dependencies directly on our local machine (i.e., without using the provided container), cloned the repository (containing both the code and datasets) and compiled the code without encountering any particular issues.

With the execution of a single Python script, `build/master_script.py`, all the experiments were executed and the graphs were generated as `*.eps` files in `build/out/`. In addition, raw results and log files were generated in `exp/` in separate subfolders for each experiment, allowing additional insights and re-generating the graphs, if required, without re-running the experiments.

4.2 Results

We reproduced the core experiments from Section 5 of the paper (Figures 7 to 13) shown in Figures 1 to 9. We omitted in this report redundant results from the different reviews and always selected the most complete results for each figure. The results match the authors' original results. In particular, the missing bars in Figure 4 (a) for the RM dataset (which is twice larger than the other small datasets) show that *SE-Oracle* fails to build indexing with the provided memory budget while *EAR-Oracle* succeeds. All the figures show almost identical execution times and memory consumption.

These reproduced results support the core thesis of the paper.

5 SUMMARY

Based on the reproducibility results, the reviewers unanimously agree to mark the paper as reproducible.

REFERENCES

[1] Bo Huang, Victor Junqiu Wei, Raymond Chi-Wing Wong, and Bo Tang. 2023. EAR-Oracle: On Efficient Indexing for Distance Queries between Arbitrary Points on Terrain Surface. *Proc. ACM Manag. Data* 1, 1, Article 14 (may 2023), 26 pages. https://doi.org/10.1145/3588694

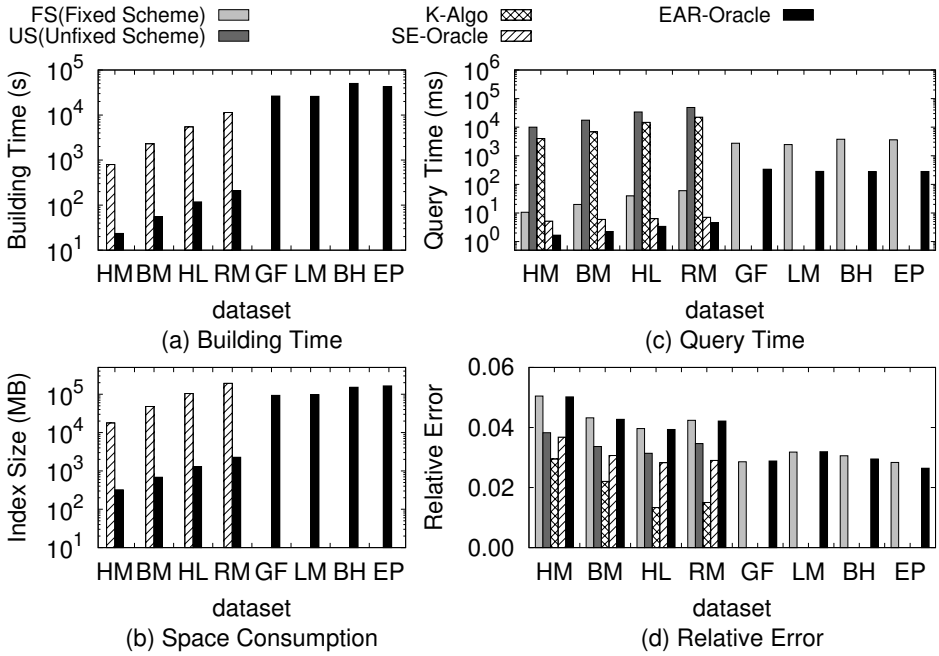

Fig. 1. Experimental Results on Unweighted Terrain Datasets (Reviewer #3)

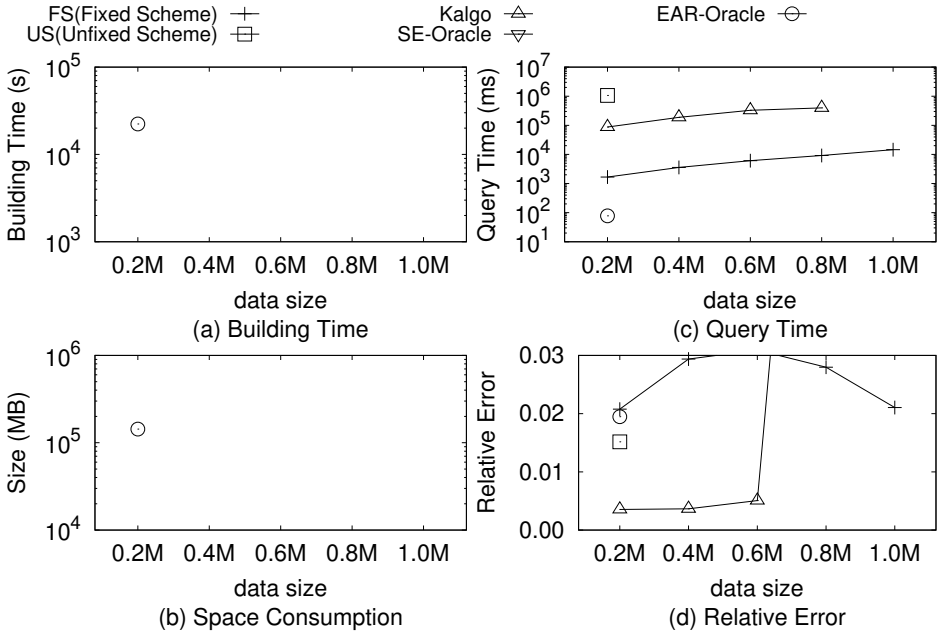

Fig. 2. Scalability Test on EP Dataset (Reviewer #1)

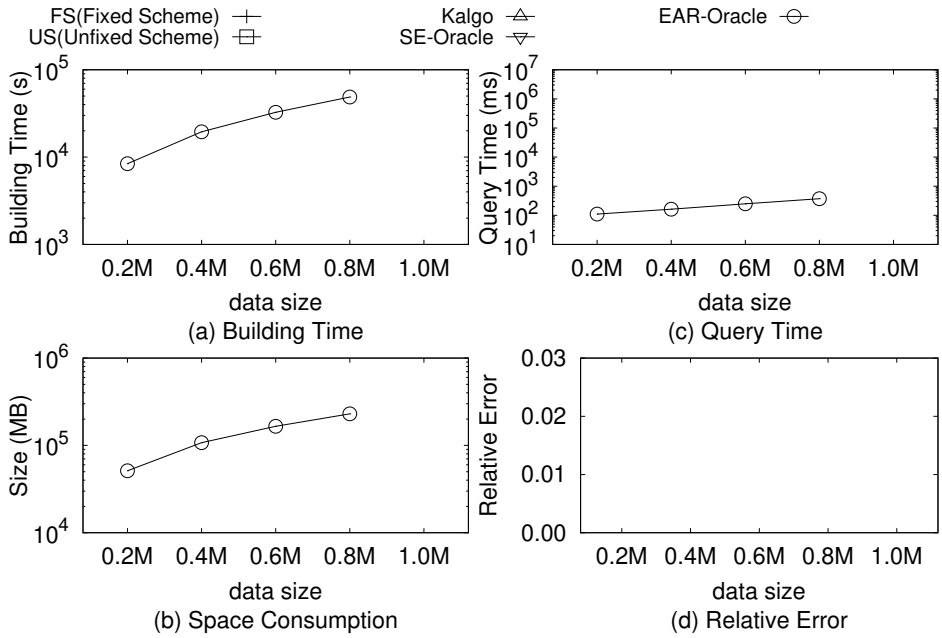

Fig. 3. Scalability Test on EP Dataset (Reviewer #2)

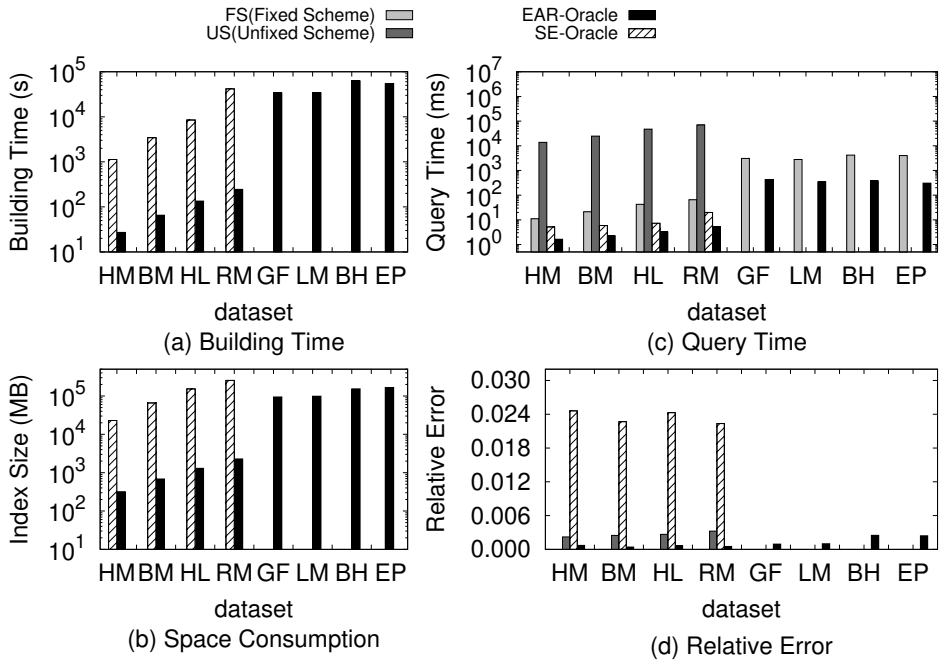

Fig. 4. Experimental Results on Weighted Terrain Datasets (Reviewer #3)

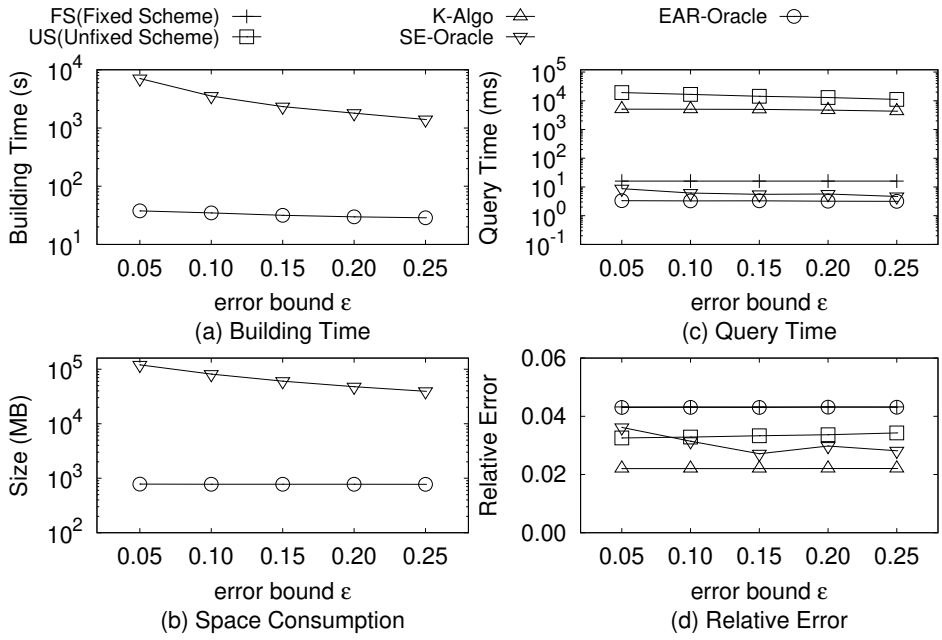

Fig. 5. Effect of ϵ on BigMountain Dataset (Reviewer #1)

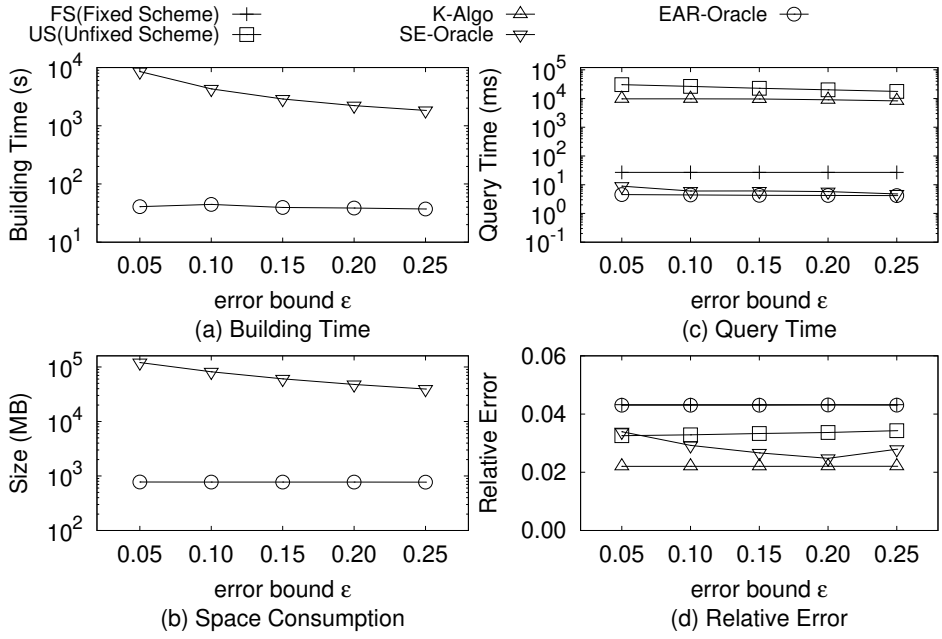

Fig. 6. Effect of ϵ on BigMountain Dataset (Reviewer #2)

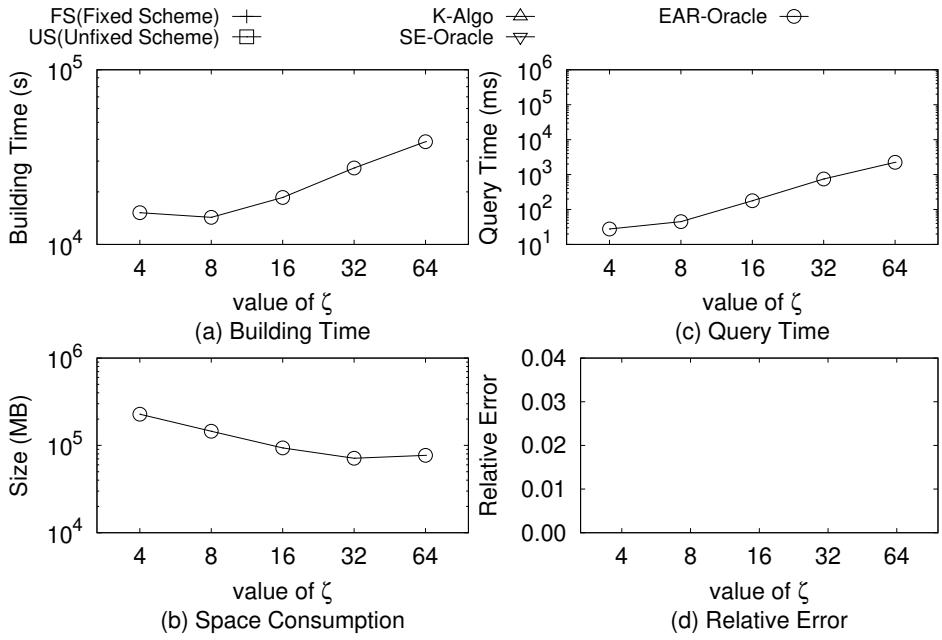

Fig. 7. . Effect of ζ on GunnisonForest (simplified) Dataset (Reviewer #1)

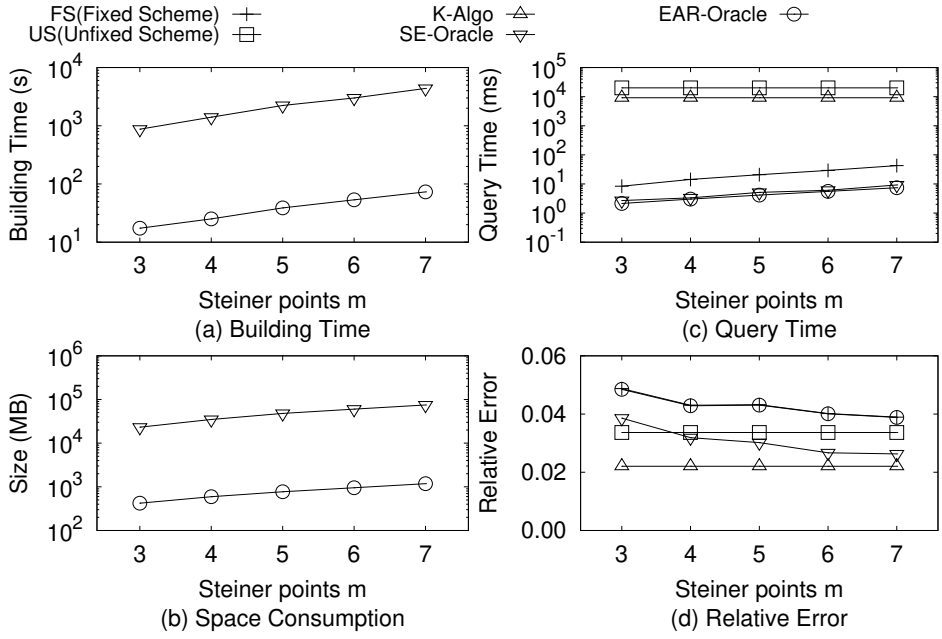

Fig. 8. Effect of m on BigMountain Dataset (Reviewer #2)

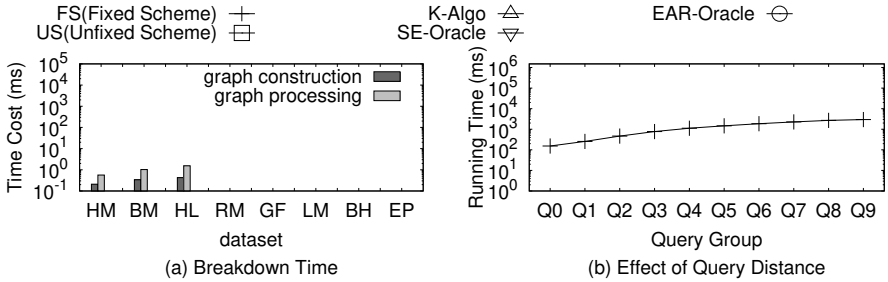

Fig. 9. Breakdown Time and Effect of Query Distance (Reviewer #2)