
ZHENGXIN YOU, The Hong Kong Polytechnic University, Kowloon, Hong Kong
AAMOD KHATIWADA, Northeastern University, Boston, USA

We have assessed the artifacts and reproduced the key findings and performance results reported in the original paper. The authors have provided compiled binary files of the experimental code and a separate README file containing guidelines, scripts, and links necessary for replicating the results and plots. Using them, we have successfully replicated the results presented in Tables 1-3 and Figures 5-10 from the original paper.

1 INTRODUCTION

The original paper [1] investigates the heuristic search approach to optimize the join order. The main idea is converting join order optimization to the shortest path problem. Then further optimization can be employed according to the properties of the join order problem and heuristic search.

The experiment compares proposed approaches with state-of-the-art algorithms on (1) optimization time and (2) quality of the computed query plan by the normalized cost (for the potentially suboptimal algorithms). Through experimental evaluations, the paper shows that for star and clique queries, heuristic search gives an optimal plan faster by an order of magnitude than the baselines.

2 SUBMISSION

The proposed method and other approaches for comparison are implemented in mutable, a main-memory database system developed by the authors’ group. ¹ A short README file contains links to the source code of this system:

- GitHub repository with code, data, and scripts at: https://github.com/mutable-org/mutable. The same resources are also available at GitLab repository https://gitlab.cs.uni-saarland.de/bigdata/mutable/mutable.
- The README file describes a setup of the system, mutable, through pre-compiled binaries and the steps to run the experiments.
- Scripts locate at a submodule named evaluation https://gitlab.cs.uni-saarland.de/bigdata/mutable/evaluation.

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 details ours and authors’ environments. As mutable is a main-memory database, the authors do not provide storage information. In our environment, the operating system is Ubuntu 20.04.4 LTS and the kernel is Linux 5.15.0-86-generic.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Repro Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Core i9-10900 CPU @ 2.8GHz</td>
</tr>
<tr>
<td>Cores</td>
<td>10 (× 2 threads)</td>
</tr>
<tr>
<td>RAM</td>
<td>64 GiB</td>
</tr>
</tbody>
</table>

¹https://mutable.uni-saarland.de/
4 REPRODUCIBILITY EVALUATION

4.1 Process
In the present reproducibility report, we used the pre-compiled binary files (version 0.0.90) as introduced in README, and cloned the repository of the same version from the URL mentioned above. Because of an authorization issue, we clone the corresponding submodule evaluation separately with its subsubmodules. We followed the steps in README file to set up the system. During the process, pygraphviz and psycopg2 packages resulted in dependency issues that we resolved using their respective open documentation. Then, we used the shell commands provided in the folder evaluation to run the experiments.

The results were generated at the root of the project, which includes three CSV files. Then running the Jupyter file in the folder evaluation, the figures were generated in the folder evaluation/fig.

4.2 Results
The resources provided by the authors are sufficient to reproduce all results shown in Tables 1-3 and Figures 5-10 in the original paper [1]. These figures contain the results of the experiments on the optimization time, the cost of the query plan, and the detailed analysis of the heuristic search algorithm. Our reproduced results are shown in Figure 1–Figure 6, which are consistent with the original paper. It is worth noting that, although there are slight variations in runtime values compared to the original paper, the observed patterns and relative performance differences remain identical. As indicated by the authors, replicating Table 4 in the original paper, located outside the experimental section, would necessitate a modification in the source code and recompilation of mutable system. Consequently, the script for reproducing this result was not provided. Nevertheless, this is trivial and does not impact the paper’s conclusions.

Fig. 1. Reproduction of Figure 5 in the original paper: Information value of different heuristics.

5 SUMMARY
The authors provide detailed, complete scripts and documentation for the reproduction. All experimental results were successfully reproduced and visualized on our machine with fully automated scripts.

REFERENCES

2Submodule’s commit SHA: e0ee342f5516efb248ce663d10d0b4d5f4f11567
3https://pygraphviz.github.io/
4https://www.psycopg.org/docs/
Fig. 2. Reproduction of Figure 6(a-c) in the original paper: Comparison of heuristic search to state of the art.
Fig. 3. Reproduction of Figure 7(a, b) in the original paper: Landscape of possible query graphs.

Fig. 4. Reproduction of Figure 8(a, b) in the original paper: Speed-up of $A^*_1 + h_{zero}$ in QGraEL.

Fig. 5. Reproduction of Figure 9 in the original paper: Detailed running time analysis of heuristic search.
Fig. 6. Reproduction of Figure 10 in the original paper: Pareto frontier of optimization time vs. plan cost. The visualization Jupyter script for this figure needs a minor modification: change \texttt{r'DP\textsubscript{CCP}'} and \texttt{r'TD\textsubscript{MinCutAGaT}'} to \texttt{r'$\text{DP}_\text{CCP}$'} and \texttt{r'$\text{TD}_\text{MinCutAGaT}$'}. 