

Replication Package for

A Geometrically Convergent Solution to Spatial Hypercube Queueing Models

By Cheng Hua, Jun Luo, Arthur J. Swersey, Yixing Wen

In the main context of our paper, we evaluate the performance of our proposed Conditional Probability Update (CPU) algorithms as well as its parallel version. This ReadMe file provides detailed instructions for replicating the figures, experiments, and analyses presented in the paper. For any issues or questions, please contact jluo_ms@sjtu.edu.cn.

1. General Introduction

The replication package includes Python code files, structured to facilitate the reproduction of all figures and tables in the paper. The package is compatible with both Linux and Windows systems. To compile and run the codes successfully, please use Python with a version higher than 3.9. Also, to produce the results, users need to prepare their Python environment by installing the following file:

➤ `requirement.txt`

In what follows, we outline the experiment configurations and provide a comprehensive guide to the code files.

2. Experiment Settings and Code Files

The paper tests various scenarios by varying the number of units and system utilization. Numerical experiments are conducted to report the computation time, mean response time (MRT), unit utilization, and the maximum percentage relative error (MPRE) of different measures between methods. The test experiment hardware setup used by the author is as follows:

- ✧ **CPU:** AMD Ryzen 9 5950X 16-Core Processor with Nvidia GeForce RTX 3090 Graphics
- ✧ **Memory (RAM):** 64GB
- ✧ **Storage:** 1.1TB

Notice that the computation time of experiments are correlated to machines' execution status (e.g., temperature, available memory, version of operation system, etc.). So it's common that the computation time might be slightly different.

Except for the discrete event simulation procedure, experiments do not need to replicate since they take the same distribution of units' location and fraction of calls under the same scenarios. All experiments will record their computation time and MPRE of certain measures comparing to other approaches. Please refer to our paper

for further details. The experiments are listed in Table 1.

For each code file, the result is exported to a CSV file with the corresponding filename after execution. It is important to note that the tables presented in the paper have undergone formatting adjustments to enhance readability, such as unifying scientific notation scales and highlighting specific results in bold. These formatting adjustments require post-processing. The tables can be directly reproduced by the above code files representatively.

Table 1 List of Experiment Code Files and Corresponding Introductions

File Names of Experiment Codes	Introduction
CPU - Heterogeneous Rate Cases	
<ul style="list-style-type: none"> ➤ HeterogeneousRate-Greenville.py ➤ HeterogeneousRate-StPaul.py 	The experiments of two cities under heterogeneous rate cases. The differences between the first two experiments files lie solely in the city's structure and system's estimated parameters, with no changes in procedure design.
CPU - Homogeneous Rate Cases	
<ul style="list-style-type: none"> ➤ HomogeneousRate-Greenville.py ➤ HomogeneousRate-StPaul.py 	The experiments of two cities under homogeneous rate cases.
Parallel CPU	
<ul style="list-style-type: none"> ➤ Parallel.py 	The experiments of parallel experiment of St. Paul. under homogeneous rate cases.
Discrete Event Simulation	
<ul style="list-style-type: none"> ➤ SimulationComparison.py 	The experiments of discrete event simulation of St. Paul. under heterogeneous rate cases and different service time distribution.
Supplementary Files	
<ul style="list-style-type: none"> ➤ Figure 5: TimeSaving&MPRE.py ➤ Figure 6: TimeSaving&MPRE-homo.py 	The execution of code relays on the result files given by the CPU algorithm under different cases.