# Path Planning using Parallel Computing

Aaron John Sabu
*Dept. of Electrical Engg. (UG Student)*
*Indian Inst. of Technology Bombay*
Mumbai, India
aaronjs@iitb.ac.in

Athul CD
*Dept. of Mechanical Engg. (UG Student)*
*Indian Inst. of Technology Bombay*
Mumbai, India
athulcd@iitb.ac.in

Ayan Sharma
*Dept. of Electrical Engg. (UG Student)*
*Indian Inst. of Technology Bombay*
Mumbai, India
@iitb.ac.in

Vaibhav Malviya
*Dept. of Electrical Engg. (UG Student)*
*Indian Inst. of Technology Bombay*
Mumbai, India
vaibhav.m@iitb.ac.in

Shivasubramanian Gopalakrishnan
*Dept. of Mechanical Engg. (Asst. Professor)*
*Indian Inst. of Technology Bombay*
Mumbai, India
sgopalak@iitb.ac.in

***Abstract*—The demand for Door-to-Door (D2D) delivery services has seen a huge surge in the wake of the pandemic and lockdowns. Path planning algorithms will play a huge role in the future when delivery is performed autonomously using intelligent UGVs and electric cars. We investigate how path planning technique can be made faster using parallel computing techniques. We perform serial implementations on C++ of Dijkstra's algorithm, the Bellman-Ford algorithm, and the Floyd-Warshall algorithm following which we implement the same with the augmentation of parallel computing platforms such as OpenMP and CUDA. The algorithms are run on two types of databases: a very large graph based on the roadmap of New York state, and smaller randomly-generated graphs of prescribed size. We present the simulation results and provide conclusions based on our inferences from the process of parallelization.**

***Index Terms*—path planning, parallel computing, OpenMP, CUDA, Dijkstra's algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm**

## I. Introduction

## II. Implementation

### A. Dijkstra's Algorithm

### B. Bellman-Ford Algorithm

### C. Floyd-Warshall Algorithm

## III. Main Results

### A. Dijkstra's Algorithm

### B. Bellman-Ford Algorithm

### C. Floyd-Warshall Algorithm

## IV. Conclusion

## Acknowledgment

The support and the resources provided by PARAM Sanganak under the National Supercomputing Mission, Government of India at the Indian Institute of Technology Kanpur are gratefully acknowledged.

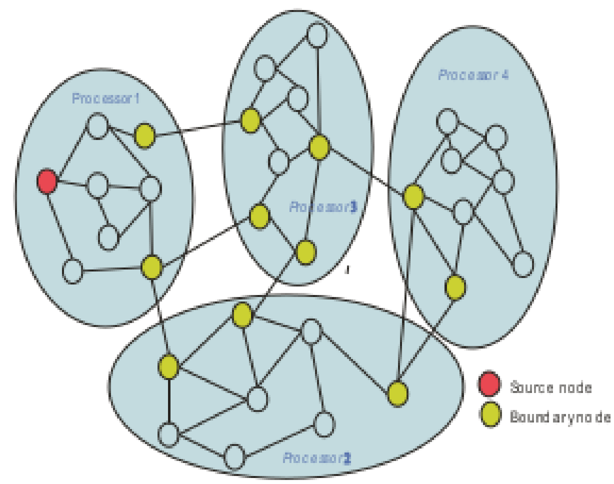


Fig. 1. Graph Partitioning (Dijkstra's algorithm) [1]

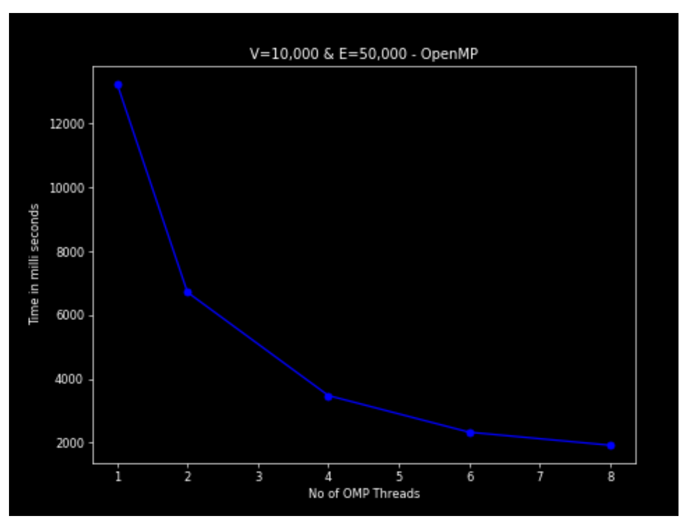


Fig. 2. OpenMP Timing Study for New York database (Dijkstra's algorithm)

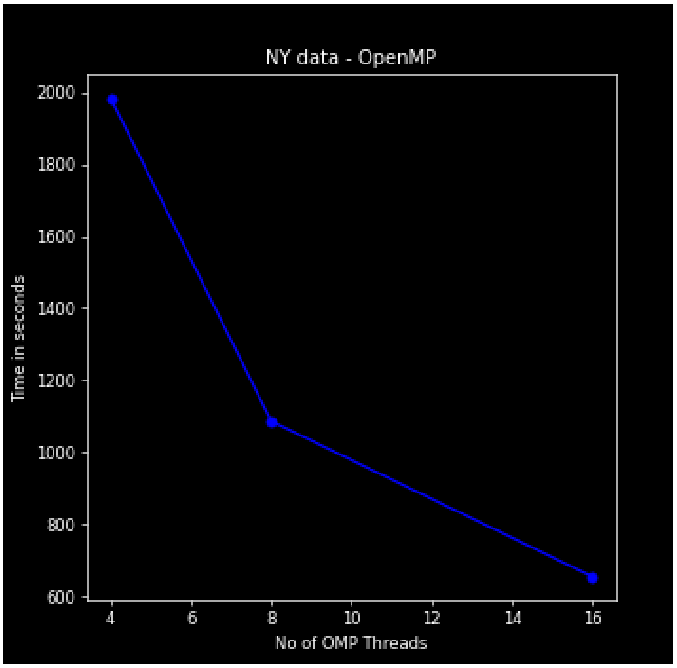


Fig. 3. OpenMP Timing Study for random graph (Dijkstra's algorithm)

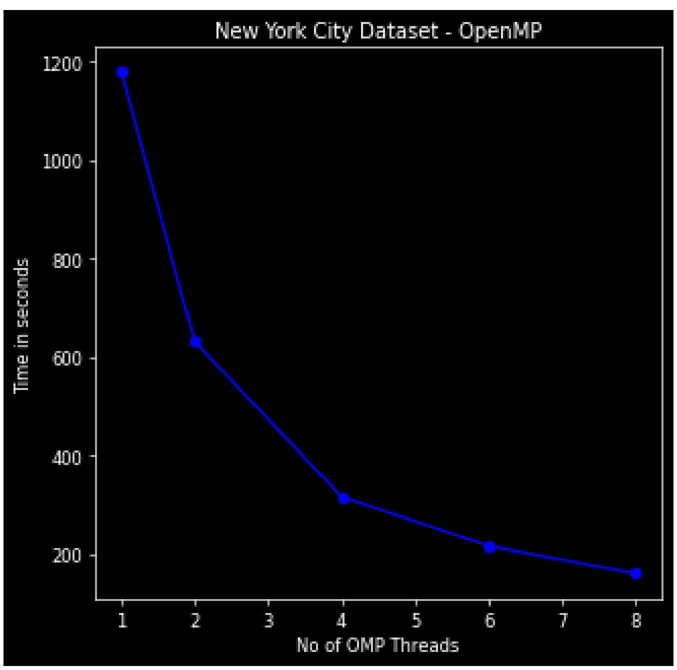


Fig. 4. OpenMP Timing Study (Bellman-Ford algorithm)

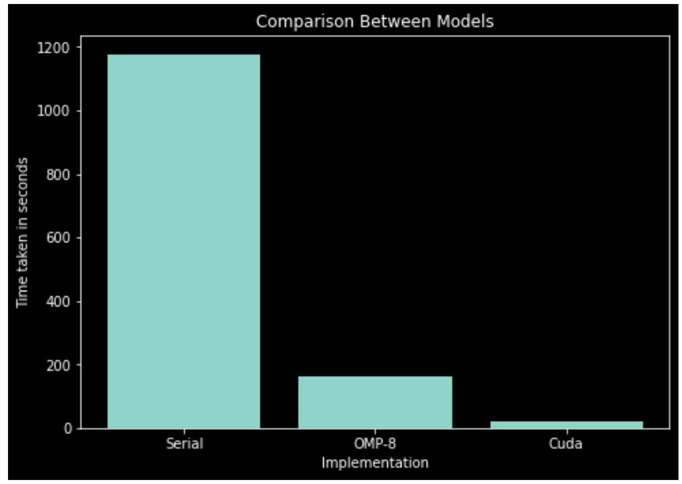


Fig. 5. Comparison between Models (Bellman-Ford algorithm)

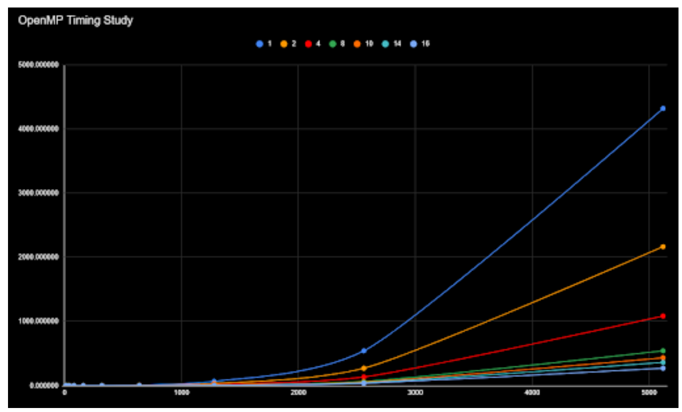


Fig. 6. OpenMP Timing Study (Floyd-Warshall algorithm)

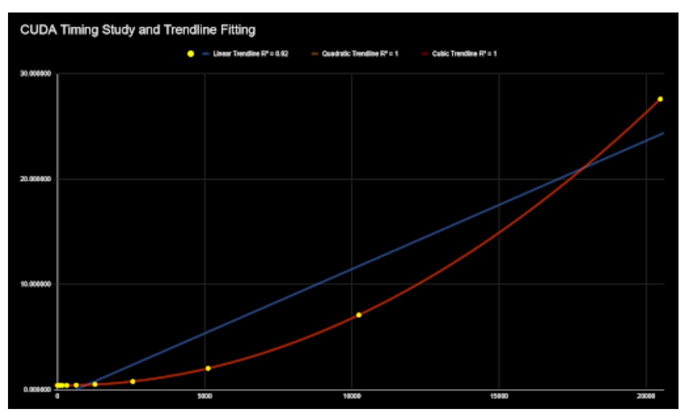


Fig. 7. CUDA Timing Study and Trendline Fitting (Floyd-Warshall algorithm)

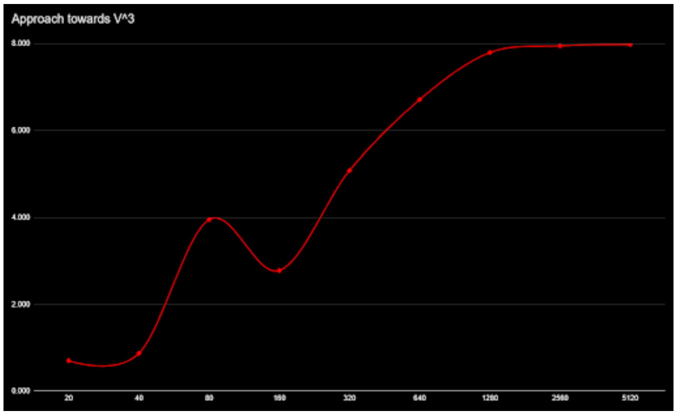


Fig. 8. Approach towards $V^3$ (Floyd-Warshall algorithm)

## REFERENCES


[1] Y. Tang, Y. Zhang and H. Chen, "A Parallel Shortest Path Algorithm Based on Graph-Partitioning and Iterative Correcting," 2008 10th IEEE International Conference on High Performance Computing and Communications, 2008, pp. 155-161, doi: 10.1109/HPCC.2008.113.

[2]