Parallelizing CBS
High & Low Level Parallelization

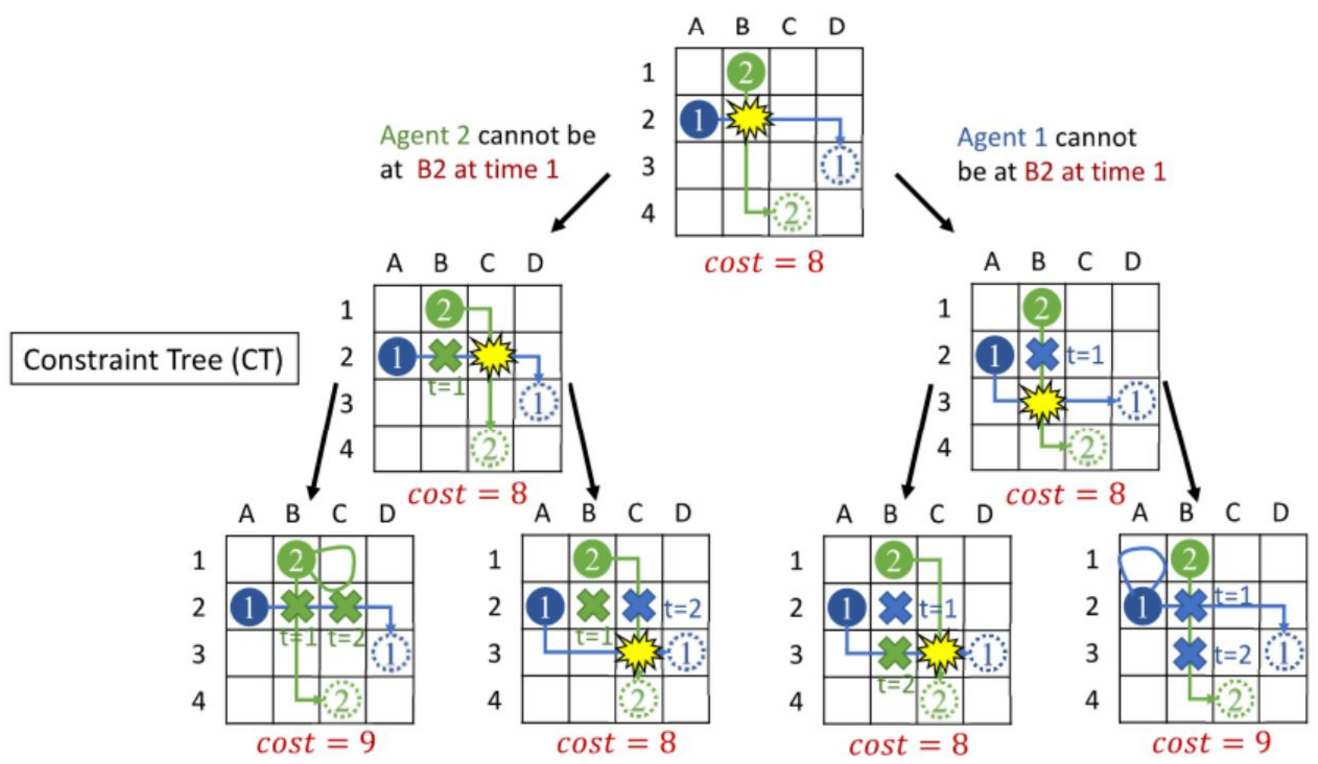
Matthew Booker, Jaekyung Song

15-618: Parallel Computer Architecture and Programming

Multi-Agent Path Finding (MAPF)

Given *N* agents, each with a start and goal position, the task is to find a path through the environment for each agent that is robot-robot and robot-obstacle collision free with minimum cost i.e. sum of lengths of all paths

Conflict Based Search [1,2]

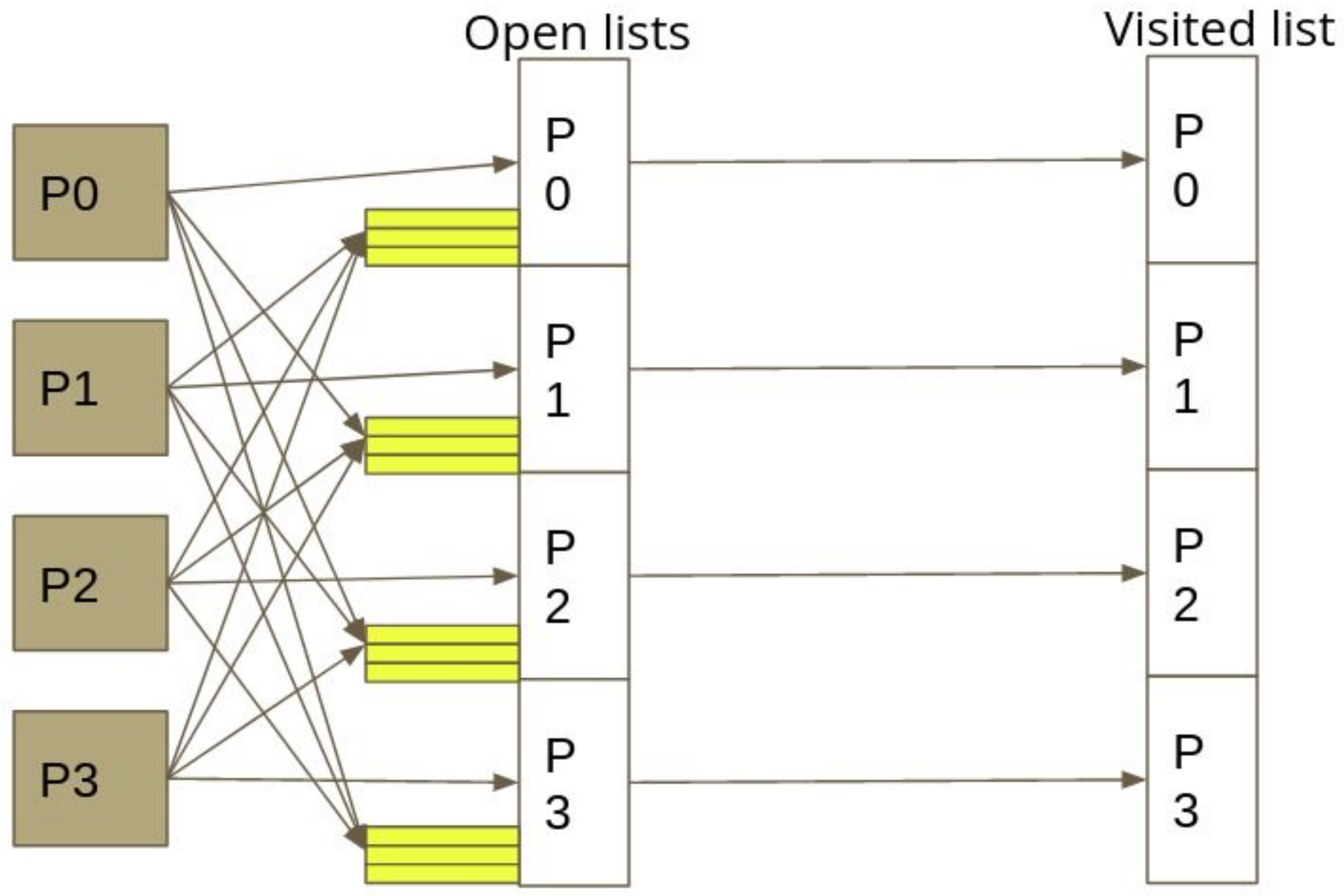


- Two level algorithm used to solve MAPF problems
- Low level uses A* for Single Agent Path Finding
- High level uses a binary search tree to select conflicts and solve for the optimal solution.

Major Challenge

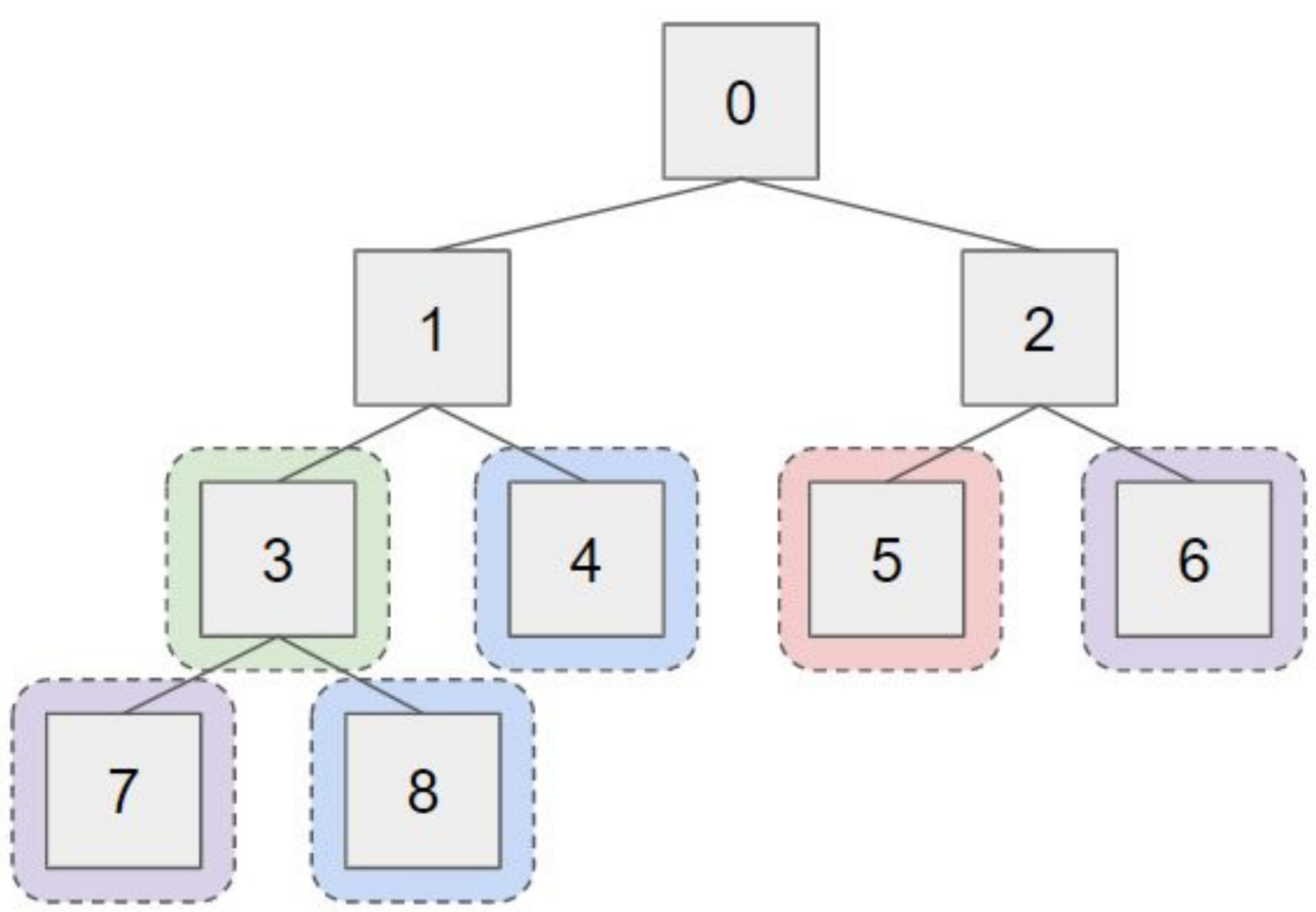
- CBS and A* use best first search and only consider a single node at a time
- Sequential nature gives very powerful properties: completeness and optimality
- Care must be taken to preserve these properties during parallelization
- Detecting termination is difficult as all cores must agree that termination should occur

Approach 1 - Parallel Low Level Search



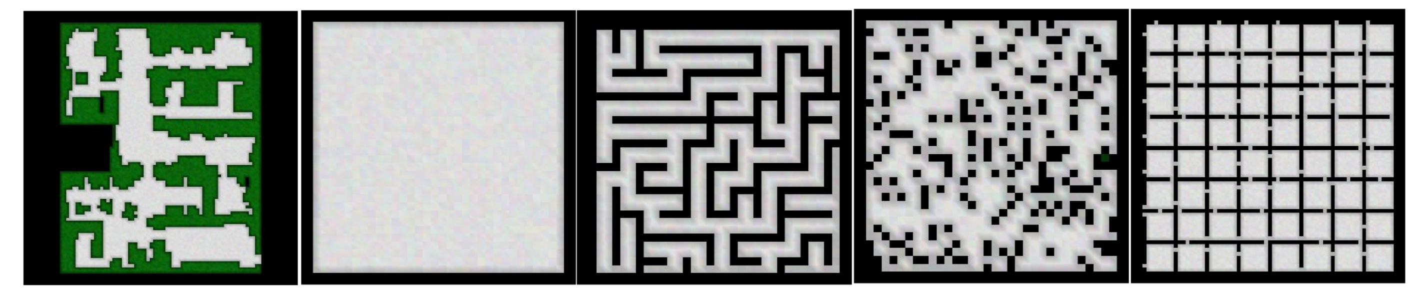
- Implementation of Hash Distributed A* [3]
- Use hashing function to distribute neighbors of expanded nodes to multiple cores
- Use buffers to reduce contention as processors push to each other's open lists
- Partition visited list to allow parallel access and modification
- Search terminates when all open lists and buffers are empty

Approach 2 - Parallel High Level Search



- Nodes in Constraint Tree are independent
- Allows distributing nodes across cores
- Process initial few nodes sequentially until enough exist to spread across cores evenly.

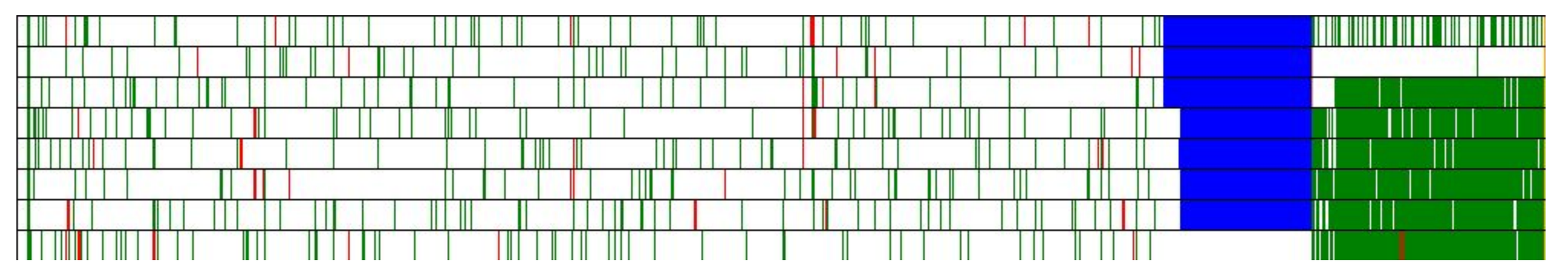
Results



Maps used for testing & evaluation

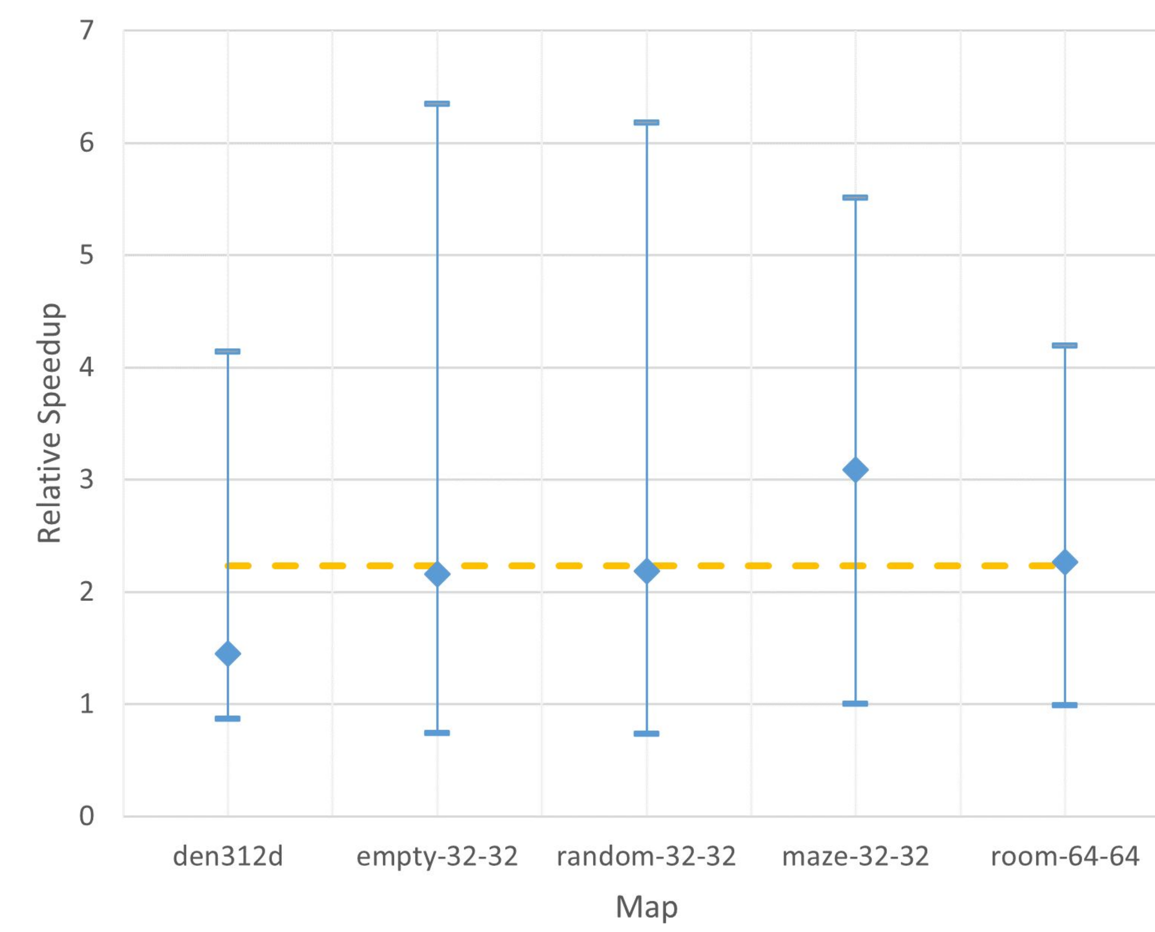
Parallel Low Level

Map	A* runtime (ms)	HDA* runtime (ms)	Speedup
den312d	237.1	212.2	1.12
empty	24.9	16.6	1.51
maze	580.8	502.1	1.16
random	19.5	16.5	1.18
room	592.3	598.0	0.99



- HDA* yielded 20% average speedup
- Overhead visualization for 8 cores shows high synchronization and contention inefficiencies

Parallel High Level



- Average relative speedup of 2.23x with a maximum of 6.18x
- Average speedup of runtimes over 1s is 4.83x
- Overhead of parallelization leads to poor speedup for simple MAPF instances

References

[1] G. Sharon, R. Stern, A. Felner, N. R. Sturtevant, ”Conflict-based search for optimal multi-agent pathfinding”, Artificial Intelligence, Volume 219, 2015, Pages 40-66, https://doi.org/10.1016/j.artint.2014.11.006.
[2] Jiaoyang Li., CMU 16-891: Multi-robot Planning and Coordination
[3] Kishimoto, A., Fukunaga, A., Botea, A. (2009). Scalable, Parallel Best-First Search for Optimal Sequential Planning. Proceedings of the International Conference on Automated Planning and Scheduling, 19(1), 201-208. https://doi.org/10.1609/icaps.v19i1.13350