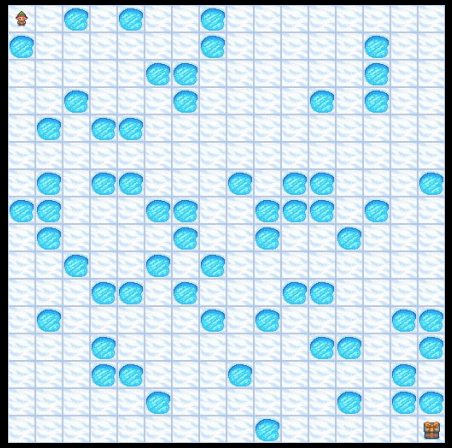
6 Apr 2025

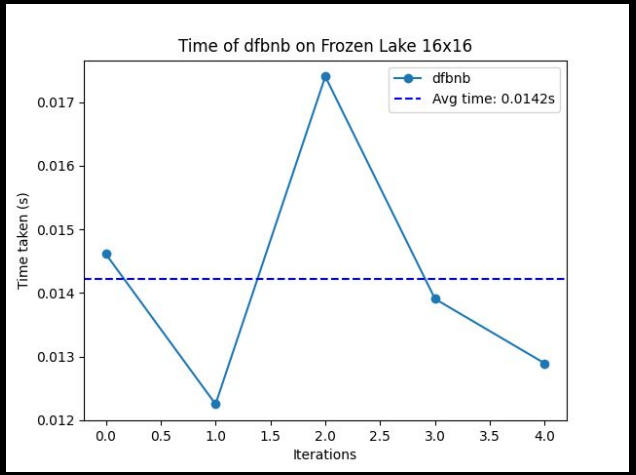
AI
Assignment 2

Shivadharshan : CS22B057
Akilesh : CS22B040

Github repo : https://github.com/Shiva9361/AI-2

01 . Branch & Bound On Frozen Lake





Heuristic Function Used : No heuristic used; BnB explores all possible paths with pruning based on path cost.

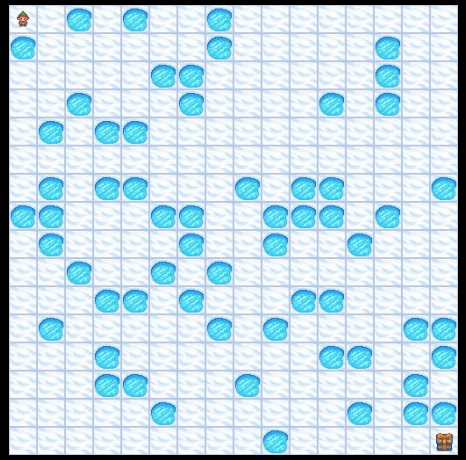
- Branch and Bound often explores a large part of the state space before converging.
- Some runs timed out when no optimal path was found within the time limit.

Observation:

- The algorithm is deterministic and performs exhaustive search with pruning.
- Sensitive to environment size; performance degrades on large maps like 16x16 , 32x32 & 64x64

Avg Execution Time : 0.0142s

02 . Iterative Deepening A* On Frozen Lake

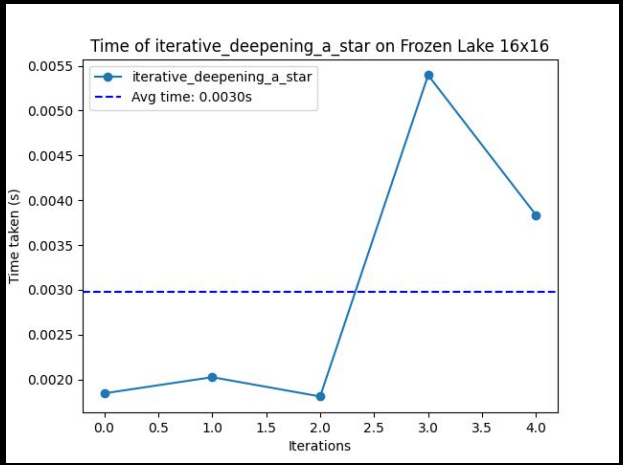


Heuristic Function Used : Manhattan distance from current cell to goal.

- IDA* converged faster on average due to effective heuristic pruning.
- Path quality was optimal or near-optimal.

Observation:

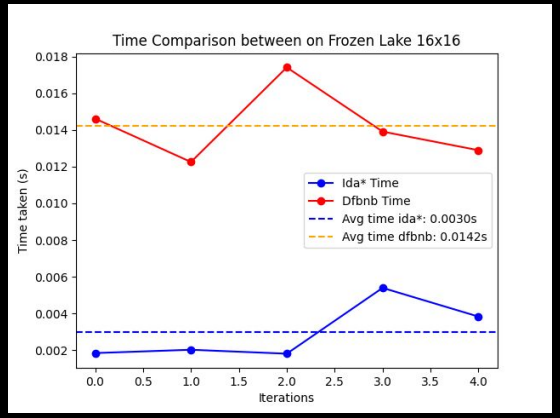
- IDA* efficiently balances between memory usage and search depth.
- More scalable than Branch and Bound on larger maps.



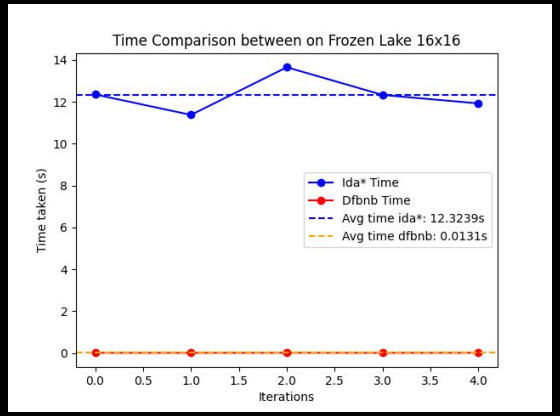
Avg Execution Time : 0.0030s

Comparison between IDA* and BNB

IDA* with d_array



IDA* without d_array



Key Observations :

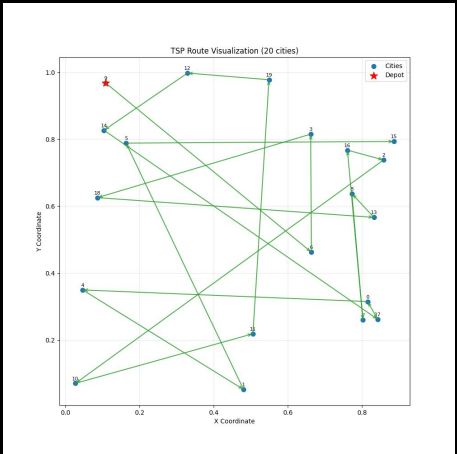
- IDA* consistently took less time than Branch and Bound in most runs.
- Average time for BnB was significantly higher due to its blind exploration.
- Heuristic in IDA* significantly improves convergence time.

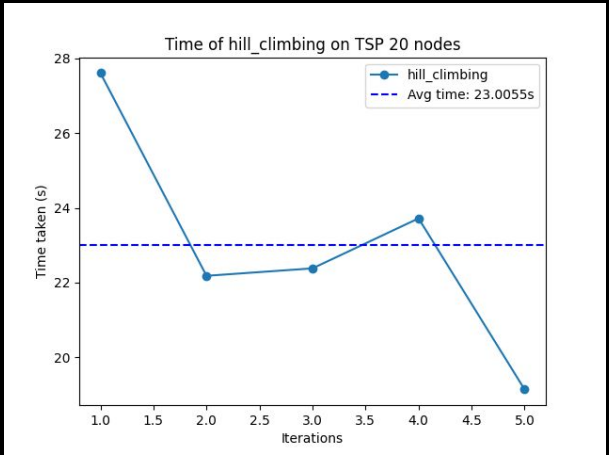
Initially, without maintaining a distance array (d_array), IDA* had much larger execution time than Branch and Bound.

This was due to lack of pruning, causing repeated exploration of the same states.

Introducing the distance array enabled effective pruning, drastically improving performance but with the cost of memory.

03 . Hill Climbing On Travelling Salesman Problem





Heuristic Function Used : Greedy local search: chooses a neighboring solution only if it improves the current cost (distance).

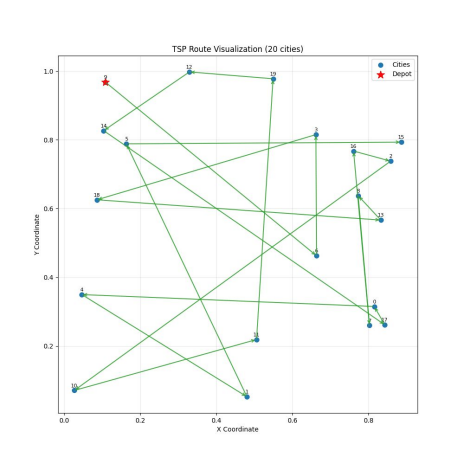
- Fast initial improvements.
- Gets trapped in local optima due to lack of exploration.
- Route quality is highly dependent on initial solution.
- Terminates when no improvement is observed for 50 iterations.

Observation:

- Converges quickly, but often before reaching global optimum.
- Stops at local minima due to no worse-move acceptance.

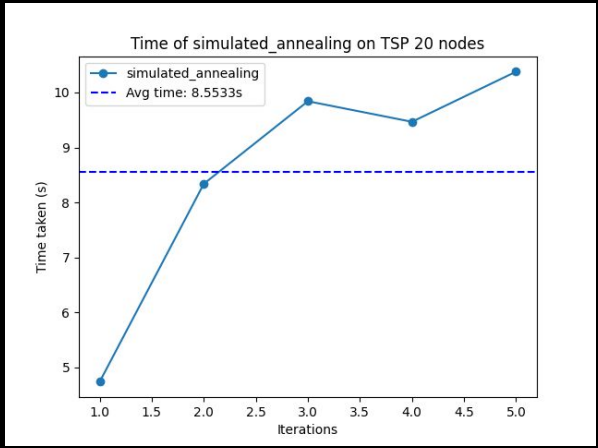
Avg Execution Time : 23.005s

04 . Simulated Annealing On Travelling Salesman Problem



Heuristic Function Used : Probabilistic acceptance of worse solutions using exp(-Δ/T), allowing exploration.
Gradual cooling helps transition from exploration to exploitation.

- More robust than HC.
- Higher-quality routes on average.
- Avoids local minima by exploring worse neighbors early.
- Slows down as temperature drops but avoids early stagnation.

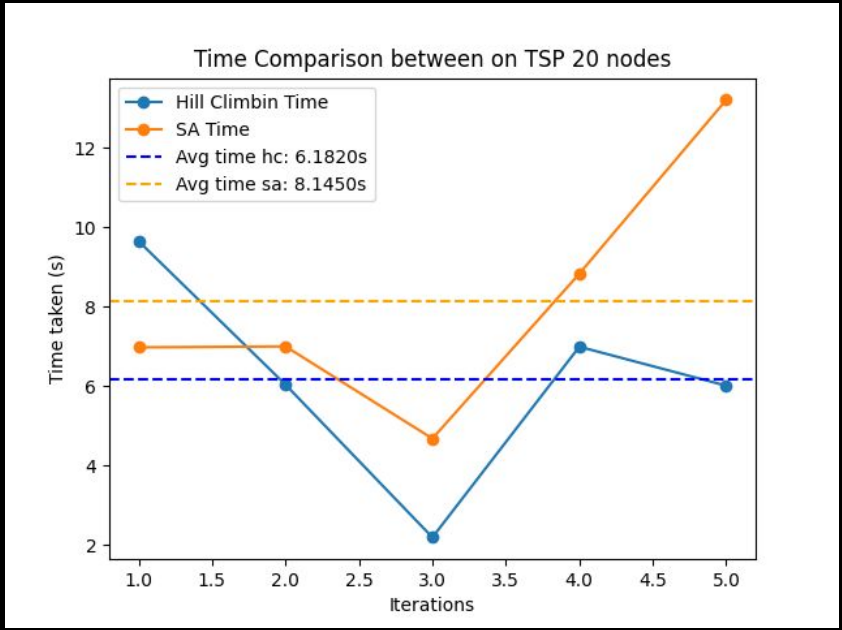


Observation:

- Slightly higher Execution Time than HC due to continued exploration.
- Converges later, but often closer to global optimum.

Avg Execution Time : 8.55s

Comparison between SA and HC



Key Observations :

- Simulated Annealing outperforms Hill Climbing in both solution quality and consistency, though at a slightly higher time cost.
- HC is faster but not as reliable for complex TSP instances.