भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

Artificial Intelligence

Assignment 2

Topics:

Branch and Bound, IDA* on FrozenLake

Hill Climbing, Simulated Annealing on TSP

Team Members:

M Akash - CS22B037

Akshatha R H - CS22B003

Branch and Bound on FrozenLake

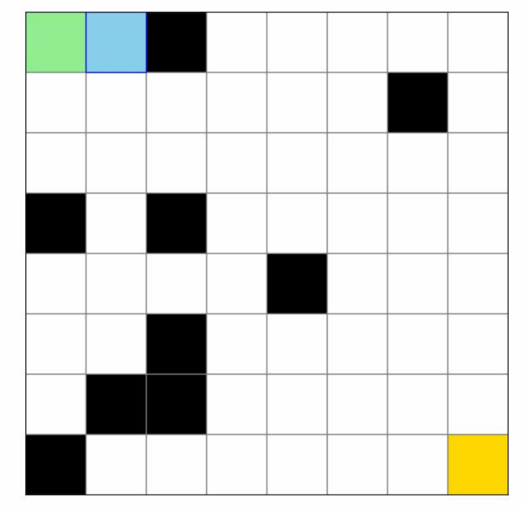
Environment: FrozenLake-v1 (Randomly generated 8x8 maps), Non-slippery(deterministic)

Heuristic: Euclidean Distance- h(n) = sqrt[(x2-x1)^2+(y2-y1)^2], expands based on min(cost+h(n))

Results: Goal was reached with exploring 44 nodes, implying minimal branching needed.

Reasons:

- Heuristic guided directly toward goal, minimal branching.
- BnB doesn't backtrack or explore alternatives after pruning higher-cost paths.
- Heuristic helps in sparse maps but fails in highly constrained ones.
- On open maps, this heuristic works very well - directs search straight toward the goal.



IDA* on FrozenLake

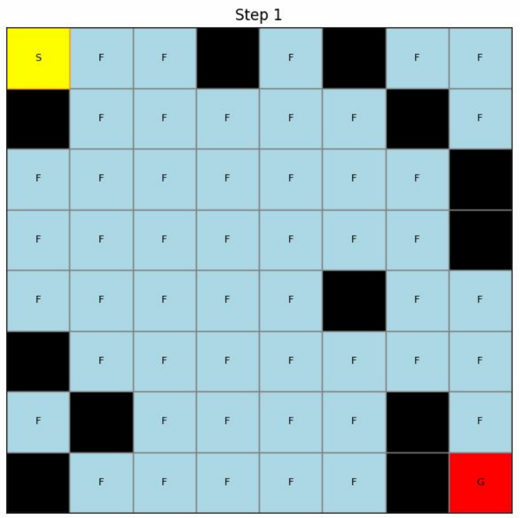
Environment: FrozenLake-v1 (Randomly generated 8x8 maps), Non-slippery(deterministic)

Heuristic: Manhattan Distance- h(n) = |x1-x2|+|y1-y2|, expands based on min(cost+h(n)), admissible

Results: Goal was reached with path length 15, and it took 2.9ms to reach the goal.

Reasons:

- The first viable path happened to also be close to optimal - so fewer thresholds were needed.
- Heuristic is Admissible & simple for grid movement (no diagonals).
- Guides search toward goal even if it’s inaccessible due to obstacles.
- Memory-efficient due to its depth-first iterative nature.
- With many being holes, branching factor reduces, but effective branching gets complex.



Step 1

S	F	F	■	F	■	F	F
■	F	F	F	F	F	■	F
F	F	F	F	F	F	F	■
F	F	F	F	F	F	F	■
F	F	F	F	F	■	F	F
■	F	F	F	F	F	F	F
F	■	F	F	F	F	■	F
■	F	F	F	F	F	■	G

Hill Climbing on TSP

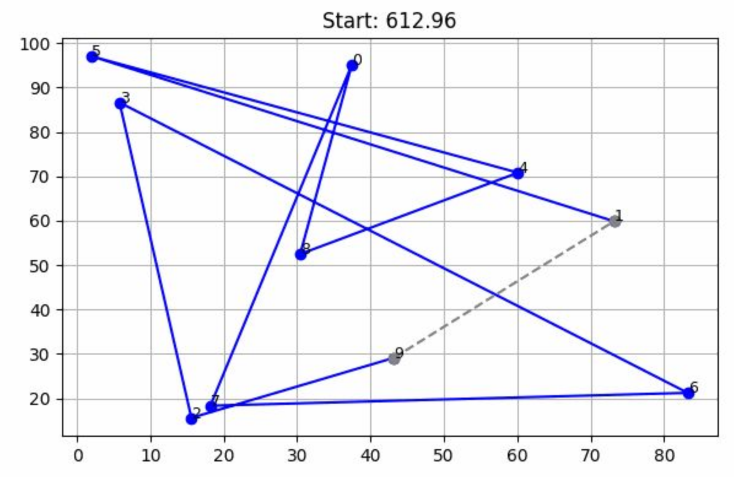
Environment: 10 randomly placed cities (2D Euclidean coordinates)

Heuristic: We use ‘Total Tour Length = Sum of pairwise Euclidean distances’, Reward= -cost

Results: Every run converged **in exactly 5 steps,** avg runtime is 2.6ms, had 42% improvement(613 to 356.68)

Reasons:

- Hill Climbing is a Greedy algorithm, always moves to best immediate neighbor.
- This leads to quick convergence but causes it to get stuck in local minima.
- Hill Climbing does not backtrack or explore worse paths.
- This is a direct and reliable heuristic for evaluating how "good" a solution is — but not helpful for guiding out of local minima.



Simulated Annealing on TSP

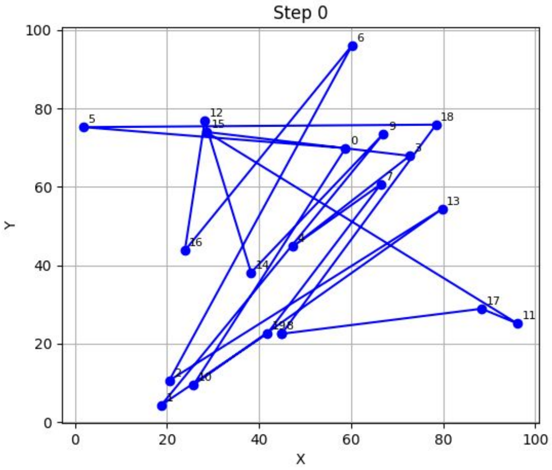
Environment: Cities=20, Initial Temp=1000, Cooling rate=0.995, Stopping Temp=1e-3

Heuristic: This cost function guides the search toward shorter paths. No explicit heuristic.

Results: Converged quickly (after 1 run), took <0.05sec, stabilized in under 2,300 iterations(max cap is 1,00,000), Best tour cost was ~470

Reasons:

- While Total tour distance always accurate, it doesn't help SA plan ahead.
- SA relies heavily on the cooling rate to balance exploration and exploitation.
- SA here only swaps two cities (1-move neighborhood).
- SA's power comes from accepting worse solutions early on (via probability). As temperature drops, this probability shrinks, and the algorithm behaves more like hill climbing.



Final Comparison & Insights

Algorithm	Env	Avg Convergence	Avg Time	Remarks
Branch & Bound	FrozenLake	(7,7) (point of conv)	140ms	Greedy, gets trapped
IDA*	FrozenLake	15 (steps)	1.2ms	Complete, slow on dense maps
Hill Climbing	TSP (10)	5 (steps)	2.6ms	Fast, stuck in local optima
Sim. Annealing	TSP (20)	1795 (iterations)	0.2s	Best cost, escapes local optima